Towards Probabilistic Generative Models for Socially Intelligent Robots

Xavier Alameda-Pineda

Habilitation à Diriger des Recherches

Towards Probabilistic Generative Models for Socially Intelligent Robots

Learning Multi-Person Robot Interactions from Audio-Visual Data

Xavier Alameda-Pineda, PhD

Inria Grenoble Rhône-Alpes
Laboratoire Jean-Kuntzmann
Multidisciplinary Institute of Artificial Intelligence
Université Grenoble Alpes

Defended before the following committee:

Prof. Ramesh Jain	University of California, Irvine	Rapporteur
Dr. Emmanuel Vincent	Inria Nancy Grand-Est	Rapporteur
Prof. Christian Wolf	Institut National de Sciences Appliquées, Lyon	Rapporteur
Prof. Rita Cucchiara	Università degli Studi di Modena e Reggio Emilia	Examinatrice
Prof. Kristen Grauman	University of Texas at Austin	Examinatrice
Dr. Edmond Boyer	Inria Grenoble Rhône-Alpes	Examinateur

Abstract

Socially intelligent robots represent a scientific and technological challenge that needs significant contributions at the cross-roads of machine learning, computer vision, audio processing and robotics to be addressed. Naturally, given that the communication modalities most commonly used in human interaction are audio and video, audio-visual fusion plays is a skill to be mastered for robots to exhibit social intelligence. The scenarios and tasks to be addressed in social robotics, require learning methods operating in unsupervised settings and handling the uncertainty in perception and in action. In addition, the developed tools cannot disregard the recent advances and breakthrough in representation learning brought by deep neural architectures. Consequently, combinations of probabilistic models – able to account for uncertainty – and deep neural networks – learning powerful data representations – seem the most appropriate methodological framework to develop social intelligence for robotic platforms. In this manuscript, we motivate the interest of developing machine learning methods exploiting audio-visual data to endow robots with social intelligence. We present the foundations of probabilistic generative models and discuss in detail seven recent contributions. Each of them addresses a different task, useful to acquire social intelligence, and are developed under the large framework of deep probabilistic models – combinations of deep neural networks and probabilistic models. Several future research opportunities as well as my personal scientific ambition for the next years conclude the manuscript.

Contents

Contents v

I. Introduction 1
- I.1 Socially Intelligent Robots . . . 1
- I.2 Probabilistic Generative Models . . . 2
- I.3 Models with Computationally Tractable $p(\mathbf{x})$ . . . 4
- I.4 Models with Computationally Intractable $p(\mathbf{x})$ . . . 5
- I.5 Learning with Audio-Visual Data . . . 8
- I.6 Document Structure . . . 10

A Computationally Tractable $p(\mathbf{x})$ 11

II. Robust Clustering for Audio-Visual Speaker Detection 13
- II.1 Introduction . . . 14
- II.2 Gaussian Mixtures with Weighted Data . . . 15
- II.3 EM with Fixed Weights . . . 16
- II.4 Modeling the Weights . . . 17
- II.5 EM with Random Weights . . . 17
- II.6 Estimating the Number of Components . . . 19
- II.7 Algorithm Initialization . . . 21
- II.8 Experiments . . . 22
- II.9 Audio-Visual Clustering . . . 25
- II.10 Conclusions . . . 28

III. Non-linear Regression for Acoustico-Articulatory Speaker Adaptation 29
- III.1 Introduction . . . 30
- III.2 Gaussian Mixture Regression . . . 31
- III.3 Integrated Cascaded GMR . . . 32
- III.4 EM algorithm for IC-GMR . . . 33
- III.5 Joint GMR . . . 36
- III.6 EM algorithm for J-GMR (with missing data) . . . 37
- III.7 Discussion . . . 39
- III.8 Experiments . . . 40
- III.9 Conclusions . . . 44
- III.10 Appendix: Derivation of Q for IC-GMR . . . 44
- III.11 Appendix: Maximization of Q for IC-GMR . . . 45
- III.12 Appendix: Calculation of Q for the Joint GMR model . . . 46

IV. Robust Deep Regression for Computer Vision 49
- IV.1 Introduction . . . 50
- IV.2 Related Work . . . 50
- IV.3 Deep Regression with a Robust Mixture Model . . . 51
- IV.4 Experiments . . . 53
- IV.5 Conclusions . . . 57

B Computationally Intractable $p(\mathbf{x})$ **59**

V. Variational Expectation-Maximisation for Audio-Visual Multi-Speaker Tracking **61**
V.1 Introduction . . . 62
V.2 Related Work . . . 63
V.3 Proposed Model . . . 64
V.4 Variational Approximation . . . 66
V.5 Variational Expectation Maximization . . . 67
V.6 Algorithm Implementation . . . 69
V.7 Experiments . . . 70
V.8 Conclusions . . . 78
V.9 Appendix: Learning the parameters of the linear transformations . . . 79

VI. Conditional Random Fields for Deep Pixel-Level Inference **81**
VI.1 Introduction . . . 82
VI.2 Related work . . . 82
VI.3 Attention-Gated CRFs for Deep Structured Multi-Scale Feature Learning . . . 83
VI.4 Exploiting AG-CRFs with a Multi-scale Hierarchical Network . . . 85
VI.5 Experiments . . . 86
VI.6 Conclusions . . . 89

VII. Variational Auto-Encoders for Audio-Visual Speech Enhancement **91**
VII.1 Introduction . . . 92
VII.2 Related Work . . . 93
VII.3 Audio VAE . . . 94
VII.4 Visual VAE . . . 95
VII.5 Audio-visual VAE . . . 95
VII.6 AV-VAE for Speech Enhancement . . . 97
VII.7 The Mixture of Inference Networks VAE . . . 99
VII.8 MIN-VAE for Speech enhancement . . . 102
VII.9 Experiments . . . 104
VII.10 Conclusions . . . 106

VIII. Conditional Adversarial Networks for Unsupervised Person Re-Identification **107**
VIII.1 Introduction . . . 108
VIII.2 Related work . . . 108
VIII.3 Clustering based Unsupervised Person Re-ID . . . 109
VIII.4 Beyond Unsupervised Person Re-ID with Adversarial Networks . . . 110
VIII.5 Experimental Validation . . . 111
VIII.6 Conclusions . . . 116

IX. Conclusion and Future Research Opportunities **117**
IX.1 Summary . . . 117
IX.2 Conclusions . . . 118
IX.3 Future Research Opportunities . . . 118

Bibliography **123**

Chapter *I*

Introduction

I.1 Socially Intelligent Robots

Robots sharing our daily lives is both a societal mirage and a scientific Ithaca. A societal mirage because of all the science-fiction stories about companions robot and the recent advances in artificial intelligence; a scientific Ithaca because the interaction skills of current robotic platforms are limited to a handful of tasks and environments. My long-term research goal is to contribute to the development of *socially intelligent robots*. Social intelligence, as defined by Edward Lee Thorndike in 1920, is the ability to understand and manage our relationship with other people and to engage in adaptive social interactions, see [1]. In other words, social intelligence is the competence to understand the environment optimally and react appropriately for socially successful conduct, see [2]. At the light of this definition, it becomes quite clear that we are very far from developing robots that exhibit social intelligence. The amount and variety of scientific skills required to implement social intelligence in robotic platforms are vast, and it would be overly ambitious and unrealistic to pretend to address it with the expertise of a single researcher or research group. I am particularly interested in developing low-level robotic social intelligence, meaning "close to the raw sensor signal:" in the lower abstraction layers. High-performance tools and methods able to process raw robotic data (perception and action signals) are necessary before connections with more abstract representations (semantics, knowledge) are possible. The rest of the manuscript should be understood in this scientific and technological context.

There are three important aspects to be taken into account to develop socially intelligent robots (or systems) at any level of abstraction. First, environments populated by humans are inherently multi-modal, and we give priority to auditory and visual sensors because they are (i) the primary perception modalities for social interaction, and (ii) technologically mature. Second, these environments are populated by persons, in plural, and one must develop tools to handle the complexity of multi-person interactions. Third, the robot must perceive and act, thus inducing changes in the environment with its presence and actions. These three aspects hide a common challenge: the phenomena linking the sources of information and the raw signals are very complex, and it is very difficult to establish rules on how the information is mixed and transformed from the sources to the sensors. Examples of such complex relationships are between the images of a speaker's lips and the corresponding sound waveform, between the raw audio and the dynamics of a multi-person conversation or between the robot motion commands needed to join an ongoing interaction and the reaction of its participants. Current advances in machine learning and deep learning demonstrated that these phenomena can be learned to a certain extent. This motivates the subtitle of this manuscript: "Learning Multi-Person Robot Interactions from Audio-Visual Data."

The information processing community at large (computer vision, audio processing, multimedia, natural language processing, to name a few) has been significantly impacted by the development of deep neural networks capable to digest large amounts of observations and to solve certain specific tasks close to or beyond human performance. This has changed the way we tackle information processing and how we address and conceive machine learning methods. Despite the undeniable advances made thanks to deep neural networks, it is unclear how to efficiently deal with the uncertainty proper to the scenarios of our interest. For instance, when combining observations from different modalities, or through time, or when retrieving the source that generated an observation, it is important to account for uncertainty while learning. Simply because most of these tasks are solved in an unsupervised way, i.e. without ground truth. Probabilistic generative models (PGM) are very well tailored for this kind of setup, since they are able to model the data generation process without requiring access to labels. Indeed, PGM learn the probability distribution of the observations, often depending on hidden or latent variables. Since PGMs deal with the full distribution of the latent/observed variables (or a good approximation of it), they can take into account uncertainty in processing information for both perception and action.

Naturally, one would like to combine the representation power of deep neural networks, with the flexibility and the interpretability of probabilistic generative models. There are different possible ways to do that, depending on the level

of fusion between these two big methodological families. Firstly, one can simply extract observations with deep neural networks, e.g., person detections and sound directions of arrival, and then fuse the observations together and through time with a PGM for instance to track multiple speakers. In this case, deep neural networks are trained before hand, and then used to provide information to the unsupervised PGM. Examples of PGMs able to integrate deeply extracted features are shown in Chapters II, III and V. A more integrated learning scheme would consist in using a deep neural network as a back-bone to extract features, and then implementing probabilistic inference within the network, as if it was part of the feed-forward pass. In this case, the parameters of the probabilistic model become parameters of the network, and the PGM is used to train also the back-bone. However, by design, the parameters of this back-bone are not seen as parameters of the PGM. Examples of this way of combining deep and probabilistic models can be found in Chapters IV and VI. Finally, a fully integrated option consists in including all the parameters of the network as parameters of the PGM, and then training the entire model within the probabilistic formulation. Examples of this family can be found in Chapters VII and VIII. The difference between the second and third familis is due to historical reasons rather than to a key methodological aspect. For instance, both a CNN backbone toped with a mixture model and a variational autoencoder are trained by maximising a lower bound of the observed log-likelihood. However, the tools used to optimise the respective lower bounds are very different. It is not clear if one of this methodologies is systematically better than the others. At a first glance, one may think that integrating all parameters of the model within the probabilistic formulation is the optimal choice, but this may lead to computationally very heavy training and inference algorithms. Splitting the work in two parts, first extracting key information from the raw data using a DNN and then fusing the extracted information by means of a PGM will simplify the probabilistic inference algorithms, but the deep representations will be learned to minimise a different objective than the likelihood of the PGM. This is why, even if I am firmly convinced that combining probabilistic graphical models and deep networks is the key to develop social robotic abilities, I also believe that we are only at the beginning of this gold mine. I consequently chose the title of this manuscript to be: "Towards Probabilistic Generative Models for Socially Intelligent Robots."

The remaining of the Chapter is structured as follows. First, in Section I.2 the foundations of PGMs, the linear-Gaussian model and the problem of modeling with latent variables are introduced. Then, two big families of PGMs, those with computationally tractable and intractable likelihood, are discussed, in Sections I.3 and I.4 respectively. Given the role auditory and visual sensing played in my research, some generalities on learning with audio-visual data are presented in Section I.5. Finally the structure of the document is depicted.

I.2 Probabilistic Generative Models

Traditionally, generative models have been considered in contrast to discriminative models, for classification. Indeed, generative models learn $p(x|c)$, where x is the observation and c is the class, and then compute $p(c|x)$ using the Bayes theorem, while discriminative models learn $p(c|x)$ directly. The advantage of generative models is that a new class can be simply added by learning $p(x|c')$, while discriminative models have to relearn from scratch when a new class is added. The drawback of generative models is the underlying assumption that the conditional probability $p(x|c)$ fits the chosen distribution, which is not always the case. Moreover, generative models possess a prominent advantage: one can sample from them and generate data. However, generating data by random sampling of probabilistic models does not always provide satisfactory results.

With the development of deep architectures the generation of new sample points became not only a feasible option, but it actually open new lines of research since the generated samples can be too real – the so-called deepfakes. Additionally, the data generated from the probabilistic model, specially if some of their properties can be controlled, can be used to train robust recognition models. This may be useful when the collection, annotation and curation of large datasets is costly or very hard (e.g. privacy or ethical issues).

In this section we will describe the foundations of probabilistic generative models. More precisely, we will discuss two basic concepts in probabilistic generative models. First, the linear-Gaussian model, which is the building block of many widely popular PGMs, including probabilistic principal component analysis and variational auto-encoders. Second, we will discuss the problem of learning with latent variables, which is the most common formulation for unsupervised learning with probabilistic generative models.

I.2.1 The Linear Gaussian Model

The linear-Gaussian model is an extremly popular model in the literature. This is due to a key intrinsic property of the multivariate Gaussian distribution, which is defined on a generic Euclidean (real) space of finite dimension X, $\mathbb{R}^X$. The probability density function (PDF) of a multivariate Gaussian distribution writes:

$$p(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{I.1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^X$ is the *mean vector* and $\boldsymbol{\Sigma} \in \mathbb{R}^{X \times X}$ is the covariance matrix, and must be symmetric and positive definite. The term in the exponential if often named the *Mahalanobis distance* [3], and is denoted as:

$$\mathcal{M}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \|\mathbf{x} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}} = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}). \tag{I.2}$$

The level-curves of the probability density function are the same of the Mahalanobis distance: the shorter the distance the higher the value of the PDF. For any unit vector $\mathbf{v}$, the quantity $\mathbf{v}^\top \boldsymbol{\Sigma} \mathbf{v}$ denotes the variance of the distribution in the direction $\mathbf{v}$, see [4]. Moreover, since the covariance matrix is symmetric and positive definite, all its eigenvalues are real and strictly positive. We can therefore write the eigendecomposition as $\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with the (strictly positive) eigenvalues and $\mathbf{U}$ is a matrix with the eigenvectors as columns. Therefore, the inverse of the covariance matrix – the precision matrix – writes $\boldsymbol{\Sigma}^{-1} = \mathbf{U} \boldsymbol{\Lambda}^{-1} \mathbf{U}^\top$. Combining these two properties, one can see that the level-curves of $\mathcal{M}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ are further away from $\boldsymbol{\mu}$ in the direction(s) or larger variance (*i.e.* in the direction of the eigenvector corresponding to the largest eigenvalue(s)). From the fact that $\boldsymbol{\Sigma}$ (and $\boldsymbol{\Sigma}^{-1}$) are symmetric and positive define (and therefore have strictly positive eigenvalues), one can understand that the level curves of (I.1) and (I.2) are ellipsoids.

The Gaussian distribution has many other interesting properties, but we will focus in the relationship of this distribution with affine transformations. Indeed, a very well known result is the so-called *linear-Gaussian* model, heavily exploited in the literature, see for instance [5]. The main result states that, if we apply an affine transformation to a Gaussian random variable $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$, with $\mathbf{A} \in \mathbb{R}^{Y \times X}$ and $\mathbf{b} \in \mathbb{R}^Y$, of a certain finite dimension Y, the resulting random variable is also Gaussian:

$$\mathbf{X} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b} \quad \Rightarrow \quad \mathbf{Y} \sim \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top). \tag{I.3}$$

If there is additional Gaussian randomness on $\mathbf{Y}$ independent of the one in $\mathbf{X}$, then we have the construct necessary to interpret a bivariate (joint) Gaussian distribution on $(\mathbf{X}, \mathbf{Y})$ as a prior distribution on $\mathbf{X}$ plus a conditional distribution on $\mathbf{Y}|\mathbf{X}$ which consists of an affine transformation of $\mathbf{X}$ with some Gaussian random additive noise independent of $\mathbf{X}$. See further explanations in [4]. This property is one of the main reasons of the popularity of the Gaussian distribution when applied to pattern recognition, since it is the main element for probabilistic linear regression, probabilistic PCA (see next Section) and linear dynamical systems.

I.2.2 Learning with latent variables

Generally speaking, we are interested in unsupervised problems,[I.1] where we have access to a dataset of samples $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$. This dataset is used to estimate the parameters of the model, $\boldsymbol{\theta}$, through a maximum log-likelihood formulation:

$$\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(\mathbf{x}_n; \boldsymbol{\theta}). \tag{I.4}$$

This principle, when applied to a multivariate Gaussian distribution, provides the very classical estimates for the mean and covariance matrix (obtained taking the vector and matrix derivatives and equating them to zero):

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \quad \text{and} \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})^\top. \tag{I.5}$$

The principle of maximum likelihood in (I.4) builds on two very common assumptions. First, the samples in $\mathcal{D}$ are statistically independent of each other. Second, the samples in $\mathcal{D}$ are identically distributed. While the first holds for the reamining of the manuscript, the second holds everywhere except for Chapter II, where the samples are not identically distributed.

One drawback of the standard definition of the Gaussian distribution is that it does not allow to consider hidden or latent variables. An extremely large body of literature deals with the use of the Gaussian distribution in the scenario where there are hidden or latent variables. Typically, one assumes that for each sample in $\mathcal{D}$, $\mathbf{x}_n$, there is an associated hidden random variable $\mathbf{Z}_n$.[I.2] This variable can be continuous or discrete and may have or not have more complex structure. Different models arise for different choices, but all of them share the fact that the realisations of $\mathbf{z}_n$ that allowed to generate $\mathbf{x}_n$ are unknown during learning.

[I.1] Some of the contributions detailed later on cannot be strictly considered within the unsupervised learning framework, but we keep this line of presentation for the sake of simplicity.

[I.2] In this section, the hidden variable will always be denoted by $\mathbf{Z}$ and the observed variable by $\mathbf{X}$. Unfortunately, the complexity of the models described later on make very hard to uniform the notations through the manuscript, and each chapter will present its own particular notations.

The sentence above is confusing since we have not explained yet what do we mean by *generate*. In probabilistic generative models, we define the model in the order in which the generation – or random sampling – is supposed to happen. Typically, the hidden variable $\mathbf{Z}_n$ is generated (sampled) first using the parameters $\boldsymbol{\theta}_\mathbf{Z}$, thus obtaining the (unknown) realisation $\mathbf{z}_n$, and the observed variable $\mathbf{X}_n|\mathbf{z}_n$ is sampled using the conditional distribution using the parameters $\boldsymbol{\theta}_\mathbf{X}$ and obtaining $\mathbf{x}_n$. In one way or another, the generation of $\mathbf{X}_n$ depends on the value $\mathbf{z}_n$.[I.3]

The fact that $\mathbf{z}_n$ is hidden or latent is problematic in practice, because for many models the log-likelihood can only be obtained by marginalizing the hidden variable:

$$\mathcal{L}(\boldsymbol{\theta};\mathcal{D}) = \sum_{n=1}^{N} \log \int_{\mathcal{Z}} p(\mathbf{x}_n, \mathbf{z}_n; \boldsymbol{\theta}) \mathrm{d}\mathbf{z}_n, \tag{I.6}$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathbf{X} \cup \boldsymbol{\theta}_\mathbf{Z}$.

Since the direct optimisation of the above expression is often computationally or numerically intractable, it is a widely accepted common practice to consider a computationally tractable lower bound of the log-likelihood to be maximized instead. This strategy leads to the well known expectation-maximization algorithm [6], its variational version [7] as well as the variational auto-encoders [8]. These are the kind of models explored in this manuscript and their associated algorithms. In the following we will give a short overview of various models and provide interesting insights of each one, without providing all the details necessary to implement the learning algorithms. For these reasons we will drop the sample index n when presenting the models.

I.3 Models with Computationally Tractable $p(\mathbf{x})$

In this section we discuss three models for which the marginal distribution $p(\mathbf{x})$ is computationally tractable, even if its optimisation w.r.t. the parameters, *i.e.* (I.6), is not. The advantage of these models, as we will see after presenting them, is that we can learn their parameters with an exact expectation-maximization algorithm [6]. Their drawback is that they are somewhat simple. In other words, if we want to model more complex phenomena, we will certainly step out of the exact EM reduce circle, as we will see in the next Section.

I.3.1 Probabilistic PCA

This is the simplest model one can build, as it considers a multivariate Gaussian vector which is partly hidden. As discussed above, this can easily be mapped into two multivariate Gaussian vectors with an affine relationship between them. More precisely, we assume that the hidden variable is a continuous random vector of dimension Z, $\mathbf{z} \in \mathbb{R}^Z$:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \qquad p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}). \tag{I.7}$$

where $\mathbf{I}$ is the identity matrix of the appropriate dimension, $\mathbf{W} \in \mathbb{R}^{X\times Z}$, $\boldsymbol{\mu} \in \mathbb{R}^X$ and $\sigma^2 > 0$ is a positive scalar.

I.3.2 Linear Dynamical Systems

This model can be understood as the extension of Probabilistic PCA to sequences. Indeed, we will now assume that our observations consist on T vectors $\mathbf{x} = \mathbf{x}_{1:T}$, where $1:T$ collects the entire sequence, and therefore of T hidden vectors $\mathbf{z} = \mathbf{z}_{1:T}$. The dimensions of the hidden and observed spaces remain the same as in Probabilistic PCA. Linear dynamical systems assume a "dynamic" model linking the hidden variable through time in a sequential manner with first order Markovian dependecies, and an "observation" model linking the observation at each time step $\mathbf{x}_t$ with its corresponding hidden variable $\mathbf{z}_t$. More formally:

$$p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1; \boldsymbol{\mu}, \mathbf{V}), \qquad p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \mathbf{A}\mathbf{z}_{t-1}, \boldsymbol{\Gamma}), \qquad p(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{z}_t, \boldsymbol{\Sigma}), \tag{I.8}$$

where $\boldsymbol{\mu} \in \mathbb{R}^Z$, $\mathbf{A} \in \mathbb{R}^{Z\times Z}$, $\mathbf{C} \in \mathbb{R}^{X\times Z}$ and $\mathbf{V}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ are covariance matrices of dimension $Z \times Z$, $Z \times Z$ and $X \times X$ respectively. Besides the temporal aspect, there are a few differences between the definition of PPCA and the linear dynamical systems, regarding how the linear-Gaussian model is used. First, the affine transformations in PPCA, became linear transformations in LDS. Second, the spherical covariance matrices in PPCA (*i.e.* $\sigma^2\mathbf{I}$) became full covariance matrices in LDS. Of course, both PPCA and LDS can be defined in the most general case, but we chose to follow the seminal presentation in [4].

[I.3] Until now, we made a clear distinction between the random variables in upper capital letters, and their realisations in lower capital letters. Unless this distinction is required for the understanding of certain parts of the manuscript, it will not be used anymore.

I.3.3 Gaussian Mixture Models

This is the simplest model that considers discrete hidden variables. GMM assume that the hidden variable $\mathbf{z}$ represents a finite choice $\mathbf{z} \in \{1, \dots, K\}$. The sampling of $\mathbf{z}$ is done with probabilities $\pi_1, \dots, \pi_K$ that are normalised to 1 and that need to be estimated. Once the choice of $\mathbf{z}$, $\mathbf{x}$ is sampled from a Gaussian with parameters indexed by $\mathbf{z}$. We write:

$$p(\mathbf{z} = k) = \pi_k, \qquad p(\mathbf{x}|\mathbf{z} = k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{I.9}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are respectively the mean vector and covariance matrix of the k-th *component* of the GMM.

In the same way that PPCA can be extended to sequences leading to LDS, GMM can be extended to sequences as well. This means that the discrete hidden variable $\mathbf{z}$ is now a sequence of random variables with a temporal link. This model is very well known and usually referred to as hidden Markov model (HMM). It has been widely exploited in many applications requiring the analysis of sequential data. For the sake of concision, we will not discuss HMMs.

I.3.4 The expectation-maximization algorithm

All four models (PPCA, LDS, GMM and HMM) have very different underlying assumptions on the hidden variable, but share one very important property: the marginal $p(\mathbf{x})$ is computationally tractable. Indeed, for all the models described in this section, one can compute:

$$p(\mathbf{x}) = \int_{\mathcal{Z}} p(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z} \qquad \text{and} \qquad p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}. \tag{I.10}$$

The so-called *posterior distribution of the latent variable* $p(\mathbf{z}|\mathbf{x})$ is very important, as we will explain in the following. First, one can notice that, for any distribution q over the hidden variable, the following holds (see [4]):

$$\log p(\mathbf{x}; \boldsymbol{\theta}) = \underbrace{\int_{\mathcal{Z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})}{q(\mathbf{z})} \mathrm{d}\mathbf{z}}_{\mathcal{F}(q, \boldsymbol{\theta})} + D_{KL}(q(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})). \tag{I.11}$$

Since the Kullback-Leibler divergence is non-negative, we can immediately see that $\mathcal{F}$ is a lower bound of the log-likelihood we would like to maximise. Moreover, the KL divergence is zero, if and only if the two distributions are the same. So, if $p(\mathbf{z}|\mathbf{x})$ can be computed, then we can set $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x})$, and now we obtain a tight bound, since the second term of the previous equation is zero. In that case, if we now optimise the first term w.r.t. $\boldsymbol{\theta}$, and because $\mathcal{F}$ is tight to the log-likelihood, we are directly optimising the desired quantity. Notice that we are only interested in the part of the first term that depends on $\boldsymbol{\theta}$:

$$\mathcal{F}(q, \boldsymbol{\theta}) \overset{\boldsymbol{\theta}}{=} \int_{\mathcal{Z}} p(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \mathrm{d}\mathbf{z} := \mathcal{Q}(\boldsymbol{\theta}). \tag{I.12}$$

where $\overset{\boldsymbol{\theta}}{=}$ indicates equality up to an additive constant that does not depend on $\boldsymbol{\theta}$.

The function $\mathcal{Q}$ defined in (I.12) is referred to as the *expected complete-data log-likelihood* and it a very important mathematical object in EM-based learning models. Indeed, the function Q can be understood as an expectation of the complete log-likelihood over the a posteriori distribution:

$$\mathcal{Q}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} \Big[\log p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \Big]. \tag{I.13}$$

EM algorithms [4], [6] alternate between computing the Q function through the expectation above – the E step – and maximising the function Q w.r.t. the parameters $\boldsymbol{\theta}$ – the M step. Importantly, in addition to be computationally tractable, $p(\mathbf{z}|\mathbf{x})$ needs to allow for an exact computation of the Q function. If, on top of these two properties, the function Q can be optimised in closed-form, we refer to the associated EM to as an *exact EM*: the procedure will converge to a local maxima of the log-likelihood. Otherwise, some further approximations need to be done.

We are interested in the particular case when $p(\mathbf{x})$ is not computationally tractable. At its turn, this implies that $p(\mathbf{z}|\mathbf{x})$ is not computationally tractable either, and that Q cannot be computed in closed form, let alone optimised. Next section describes a few common models that *suffer* from this problem, and their associated learning algorithms.

I.4 Models with Computationally Intractable $p(\mathbf{x})$

In this section we discuss three models for which the marginal, $p(\mathbf{x})$, is not tractable, and their associated algorithms. This must not be seen as an exhaustive list, but rather as a set of simple examples that quickly step out of the exact EM learning framework described in the previous section.

I.4.1 Markov Random Fields

(MRF), also known as conditional random fields, are a class of probabilistic generative models that assume a set of observations $\mathbf{x} = \mathbf{x}_{1:V}$, indexed by the V vertices of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertices and edges of $\mathcal{G}$ respectively. A set of hidden random variables associated to $\mathbf{x}$, is also indexed by the vertices $\mathbf{z} = \mathbf{z}_{1:V}$.

The definition of a MRF is done through *potentials* instead of probability distributions. Typically, these potentials describe the relation between the observations and the hidden variables, as well as between hidden variables. While the former are referred to as *unary potentials* the later are referred to as *binary* potentials. Generally speaking one writes the following *energy potential*:

$$E(\mathbf{x}, \mathbf{z}) = \sum_{v \in \mathcal{V}} \psi_u(\mathbf{x}_v, \mathbf{z}_v) + \sum_{v, v' \in \mathcal{V}} e_{vv'} \psi_b(\mathbf{z}_v, \mathbf{z}_{v'}), \tag{I.14}$$

where $e_{vv'}$ is the weight associated to the edge from v to v', and ψ_u and ψ_b are the unary and binary potentials respectively. Once these potentials are defined, to fix ideas one can think of the Euclidean distance if both $\mathbf{x}_v$ and $\mathbf{z}_v$ are real-valued vectors of the same dimension, the energy is cast into a probability formulation by means of the exponential function. Indeed, we are interested in the a posteriori distribution of the latent variables:

$$p(\mathbf{z}|\mathbf{x}) \overset{\mathbf{z}}{\propto} \exp\Big(- E(\mathbf{x}, \mathbf{z})\Big), \tag{I.15}$$

where $\overset{\mathbf{z}}{\propto}$ denotes equality up to a multiplicative constant that does not depend on $\mathbf{z}$.

This multiplicative constant is precisely what poses a computational challenge. Even in the simplest case, *i.e.* using the Euclidean distance as potentials, the exact computation of the posterior is challenging. Indeed, if one writes down the posterior probability defined above, one quickly realises that it boils down to a Gaussian distribution of dimension VZ, where Z is the dimension of $\mathbf{z}$. Roughly speaking, the part of the precision matrix associating $\mathbf{z}_v$ and $\mathbf{z}_{v'}$ will be a diagonal matrix filled with $e_{vv'}$. In order to obtain the mean of the posterior distribution, one must inverse the precision matrix, which is an extremely costly operation. Overall, the posterior distribution is very well defined, but computationally intractable.

One possible solution is to exploit the mean-field approximation, which proposes to use an alternative to the true posterior distribution. In general, the mean-field approximation assumes that the set of hidden variables $\mathbf{z}$ splits into H non-intersecting subsets: $\mathbf{h}_h$, $h = 1, \dots, H$. Formally we write:

$$p(\mathbf{z}|\mathbf{x}) \approx \prod_{h=1}^{H} q_h(\mathbf{h}_h). \tag{I.16}$$

In this case, when runing the E-step, and minimising the KL divergence in (I.11), one obtains:

$$q_h(\mathbf{h}_h) \propto \exp\Big(\mathbb{E}_{\prod_{j \neq h} q_j(\mathbf{h}_j)}\Big[\log p(\mathbf{z}, \mathbf{x})\Big]\Big), \tag{I.17}$$

meaning that one must know use all other q_j's to obtain q_h. While this can lead to a computationally tractable solution, it has the main drawback that all q_h's except for one need to be initialised (in addition to the model parameters).

In the case of MRF, the initialisation problem could be easily solved by initialising all latent variables to the associated observation (or to the most probable value given that observation). Moreover, the most common assumption is to enforce that all hidden variables are independent of each other:

$$p(\mathbf{z}|\mathbf{x}) \approx \prod_{v} q_v(\mathbf{z}_v), \qquad q_v(\mathbf{z}_v) \propto \exp\Big(\mathbb{E}_{\prod_{v' \neq v} q_{v'}(\mathbf{z}_{v'})}\Big[\log p(\mathbf{z}, \mathbf{x})\Big]\Big). \tag{I.18}$$

Generally speaking, there is not reason whatsoever for the KL divergence to reduce to zero, when the function q is imposed to belong to a class of distributions (e.g. to factorize). As a consequence, $\mathcal{F}(q, \boldsymbol{\theta})$ is not a tight lower bound of $\log p(\mathbf{x}; \boldsymbol{\theta})$ in (I.11) anymore: variational techniques do not necessarily converge to a local optimum. The reader is referred to [7] for a more complete discussion on the topic.

I.4.2 Variational Auto-Encoders

In this section, we rapidly present the variational auto-encoder [9] and associated variational methodology for model training and inference. In a nutshell, VAEs can be seen as the non-linear extension of PPCA. The generative model of VAE follows:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}), \tag{I.19}$$

with

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \text{diag}\{\boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{z})\}\big), \qquad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}_L), \tag{I.20}$$

where $\text{diag}\{\cdot\}$ is the operator that forms a diagonal matrix from a vector by putting the vector entries on the diagonal, and $\boldsymbol{\mu}_{\boldsymbol{\theta}} : \mathbb{R}^Z \mapsto \mathbb{R}^X$ and $\boldsymbol{\sigma}_{\boldsymbol{\theta}} : \mathbb{R}^Z \mapsto \mathbb{R}^X_+$ are non-linear functions of $\mathbf{z}$ modeled by a DNN. This DNN is called the *decoder network* or the *generation network*, and is parametrized by a set of weights and biases denoted $\boldsymbol{\theta}$. Note that the marginal distribution of $\mathbf{x}$, $p_{\boldsymbol{\theta}}(\mathbf{x})$, is given by:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}. \tag{I.21}$$

Since any conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ can provide a mode, $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be highly multi-modal (in addition to being potentially highly dimensional). With this in mind it makes sense to set a diagonal covariance matrix in (I.20) since marginal distributions of arbitrary complexity can be obtained by designing and tuning the decoder network. Setting diagonal covariance matrices often makes the mathematical derivations easier.

However, (I.21) is not computationally tractable due to the non-linearity induced by the DNN. As a consequence, the posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is not tractable either and an exact EM cannot be derived. Instead, standard practice approximates the posterior distribution by means of another DNN often referred to as the *encoder*. We write:

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \approx q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \text{diag}\{\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})\}\big), \tag{I.22}$$

where $\boldsymbol{\mu}_{\boldsymbol{\phi}} : \mathbb{R}^F \mapsto \mathbb{R}^L$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}} : \mathbb{R}^F \mapsto \mathbb{R}^L_+$ are non-linear functions of $\mathbf{x}$ modeled by the encoder network. Due to this approximation, as it was the case in the MRF model, the KL divergence in (I.11) is not null, and the bound is not tight. In order to train VAE, a further step is taken, and the KL term in (I.11) is droped (ignored):

$$\log p(\mathbf{x}; \boldsymbol{\theta}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\big[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\big) =: \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}). \tag{I.23}$$

The aim now is to find both the best parameters of the generative model $\boldsymbol{\theta}$ (the ones we are interested in originally) and the best parameters of the variational posterior distribution $\boldsymbol{\phi}$. The optimisation problem to solve is:

$$\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}). \tag{I.24}$$

However, due to the non-linearities of the generative model, one cannot take the expectation in (I.23). Instead, one must approximate it by sampling from the posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) \approx \frac{1}{R} \sum_{r=1}^{R} \big[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(r)})\big] - D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\big), \qquad \mathbf{z}^{(r)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}). \tag{I.25}$$

This would be enough if we only needed to train the generative (decoder) parameters $\boldsymbol{\theta}$. However, we must also train the inference (encoder) parameters $\boldsymbol{\phi}$. This poses a problem since the sampling operation $\mathbf{z}^{(r)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is not differentiable w.r.t. $\boldsymbol{\phi}$. To overcome this issue, the reparametrization trick was proposed in [9]. The idea is quite simple, instead of sampling from a Gaussian distribution with parameters $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\text{diag}\{\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})\}$, we sample from a standard multivariate Gaussian distribution $\bar{\mathbf{z}}^{(r)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then compute the posterior sample as: $\mathbf{z}^{(r)} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) + \text{diag}\{\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})\}\bar{\mathbf{z}}^{(r)}$ in a way that now the sampling operation becomes differentiable w.r.t. $\boldsymbol{\phi}$.

I.4.3 Generative Adversarial Networks

A third, and extremely popular, model where approximate inference is required are generative adversarial networks (GAN) [10]. The reason – and the interest – of their approximate inference is the same as in VAE. As in VAE, GANs build upon deep neural networks, and therefore their construction in based on non-linear mappings. However, they differ in principle and in practice from VAE. Indeed, GANs are conceived to approximate complex marginal distributions on the observed variable $p(\mathbf{x})$, without requiring the a posterior distribution over a latent variable $p(\mathbf{z}|\mathbf{x})$.

However, GANs work with the latent variable $\mathbf{z}$ on top of the observed variable $\mathbf{x}$. While $\mathbf{z}$ is assumed to follow a standard multivariate Gaussian distribution, the conditional relationship between $\mathbf{x}$ and $\mathbf{z}$ is not stochastic, but deterministic. Indeed:

$$\mathbf{x} = f_{\boldsymbol{\theta}}(\mathbf{z}), \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{I.26}$$

Therefore the probability on $\mathbf{x}$ is defined as follows:

$$p(\mathbf{x} \in \mathcal{X}) = \int_{\mathcal{Z}(\mathcal{X})} |\nabla_{\mathbf{z}} f_{\boldsymbol{\theta}}(\mathbf{z})|^{-1} \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) \text{d}\mathbf{z}, \qquad \mathcal{Z}(\mathcal{X}) = \{\mathbf{z} | f_{\boldsymbol{\theta}}(\mathbf{z}) \in \mathcal{X}\}, \tag{I.27}$$

where we measure the set of $\mathbf{z}$'s that fall into $\mathcal{X}$ when passed through $f_{\boldsymbol{\theta}}$. In other words, $f_{\boldsymbol{\theta}}$ is transforming the standard multivariate Gaussian distribution of $\mathbf{z}$ into a more complex distribution, parametrized by $\boldsymbol{\theta}$ and the definition of $f_{\boldsymbol{\theta}}$. The question now is how to find the optimal parameters $\boldsymbol{\theta}$.

As previously announced, the aim of GANs is to approximate complex distributions of $\mathbf{x}$. To do so, GANs use a "discriminator" who's aim is to learn to make the difference between generated samples and real samples. Indeed, this discriminator – usually another neural network – is trained so as to recognize which samples come from the generator $f_{\boldsymbol{\theta}}$ and therefore are *fake*, and which ones are *real*. Therefore, the discriminator is trained to solve a binary classification task; and the generator is trained to make things difficult for the discriminator.

More formally, we assume that our discriminator is a function $d_{\boldsymbol{\lambda}}$ with parameters $\boldsymbol{\lambda}$, the learning of a GAN is cast into a min-max problem:

$$\max_{\boldsymbol{\theta}} \min_{\boldsymbol{\lambda}} \mathbb{E}_{p_T(\mathbf{x})}[1 - d_{\boldsymbol{\lambda}}(\mathbf{x})] + \mathbb{E}_{p(\mathbf{z})}[d_{\boldsymbol{\lambda}}(f_{\boldsymbol{\theta}}(\mathbf{z}))], \tag{I.28}$$

where $p_T(\mathbf{x})$ is the true posterior distribution of the data $\mathbf{x}$. Needless to say, these expectations cannot be computed analytically, and they need to be sampled from the training set and from a standard Gaussian respectively.

In order to interpret the previous equation, let us discuss what happens when one of the two networks, the generator or the discriminator, is frozen. Let us start by freezing the generator, $f_{\boldsymbol{\theta}}$. In that case, the parameters $\boldsymbol{\lambda}$ are trained to minimise the cross-entropy loss with label "1" for real data (i.e. sampled from the true distribution $p_T(\mathbf{x})$) and with label "0" for those samples generated by (the frozen) $f_{\boldsymbol{\theta}}$. If we now freeze the discriminator, the generator is clearly trained to maximize the output of concatenating discriminator and generator. In other words, the generator is trained to make $\mathbb{E}_{p(\mathbf{z})}[d_{\boldsymbol{\lambda}}(f_{\boldsymbol{\theta}}(\mathbf{z}))$ close to 1, and not to 0. The generator and the discriminator play an *adversarial game*. Finally, another possible interpretation is that the discriminator acts as a parametric loss for the generator, and this loss is modified depending on the generated samples.

GANs quickly became a machine learning revolution, and since their *debut* in 2014, there have been many applications and surprising results in terms of data (image) quality. However, training GANs requires a lot of know-how due to various problems. The most common problem is the fact that the discriminator can easily become too strong, making the task for the generator too hard. In practice, the gradient back-propagated to the generator is too small and the generator does not learn. In our practical application of Chapter VIII, due to the small capacity of the discriminator, we observed the opposite effect: the discriminator was often lost. Another recurrent difficulty is the so-called *mode collapse*. This happens when the generator produces very low-variance samples, resulting in a generated data set that lack of diversity. Various strategies have been proposed to overcome different problems, but this discussion is not in the scope of this manuscript.

I.5 Learning with Audio-Visual Data

Auditory and visual data are of very different physical, semantic and statistical nature. Indeed, while auditory data measures the air pressure of a membrane, and translate its variations into an electronic signal, visual data measures the amount of light a photosensor receives. For common objects, this has a major implication when in comes to source overlap: in audio, sources overlap by addition, in video sources overlap by occlusion. As an example, we will have trouble seeing a speaker hid behind another person, but we will clearly hear the spoken words.

The sensors used to record auditory and visual data (microphones and cameras) have also interesting consequences for computational approaches. While common microphones are (quasi) omnidirectional, i.e. they record sound from almost all directions, cameras have a limited field-of-view.[I.4] If we think of a robotic platform, this means that the robot can hear what happens all around, but can only see the part of the scene towards where its camera is pointing.

Another interesting problem is the source activity and presence in the raw data. As long as there is light, a person will always be reflecting this light, and therefore be perceivable with a camera if within its field of view. However, silent persons are unperceivable from and auditory point of view. Furthermore, when perceiving multiple sources, one must be careful to extract the information associated to each source from the raw data. In the visual modality this presence is binary: either the source is visible or it is not (occluded or out of the field of view). However, in the auditory modality, the sources are present only when active, but independently of the orientation of the device. In addition the perceived sound is usually modeled as a linear combination of the source sounds, meaning that the sounds are mixed in the environment before being perceived.

This raises the question of which representation should be used for each of the two modalities. While raw images appear to be an appropriate representation because (i) they are already spatialised and (ii) by nature cannot contain

[I.4] This claim refers to the a vast amount of cameras and microphones. Of course there exist specific devices such directional microphones and omnidirectional cameras, but they are not used everywhere. In other words, the vast majority of auditory and visual sensors used in consumer electronics are omnidirectional microphones and standard cameras with limited field-of-view.

one than more sources per pixel, the raw audio data (digital waveform) does not possess these interesting characteristics. This is why, very often, the raw audio signal is transformed into a different representation for which these properties partially hold. One of the most common representations used to this aim is the short-time Fourier transform (STFT). This representation operates on a sliding-window basis, and transforms chunks of the original signal into the Fourier domain, creating a time-frequency representation. The advantage of such a time-frequency is that one can make reasonable assumptions, such that there is only one predominant source per time-frequency point, thus getting closer to the visual spirit. This is called W-disjoint orthogonality, see [11].

Auditory and visual data differ also in the way they are contaminated with unwanted effects. Auditory data is usually contaminated with a variety of auditory events such as other people speaking or a door closing, with background noise and with reverberations. Visual data is usually contaminated by variations in the lighting conditions, background clutter and blur due to fast motion. All these pollute the raw data and make it challenging to extract information in some situations. Despite the difference between auditory and visual data, there is a large body of literature providing experimental evidence that, when correctly fused, the joint processing of auditory and visual data can provide huge benefits, see for instance [12]–[14]. This is certainly due to the complementarity of auditory and visual data. For instance, in the event of a door closing, this could be easily localised if seen, and denoised appropriately. For all this to be possible, there must be a way to relate the information extracted from the auditory and visual modalities.

The correspondence should be achieved at three different levels: temporal, geometric and semantic. The simplest form of temporal correspondence is synchrony. For many applications synchronising the video frames with the corresponding chunck of audio signal suffices to success in jointly processing auditory and visual data. However, there are tasks for which the events of interest may not be completely synchronised. For instance, in speech production, some phonemes exhibit desynchrony between the auditory and visual data. The geometric correspondence – often referred to as calibration – is usually addressed by establishing a mapping between the positions in the image and certain characteristics in the sound space representation. While this is possible for one microphone (if there are reverberations), it is far easier for microphone arrays. Roughly speaking, this is due to the fact that the sound does not arrive at the same time at all microphones, and this delay depends on the position of the sound source. Depending on the complexity of the sound representation this map can be purely geometric or learned. Finally, we desire to achieve a certain level of semantic correspondence. Differently from the geometric and temporal correspondences, semantic correspondence has a wide range of interpretations. At a very low level, we may want to guess which auditory and visual observations correspond to the same source. At a much higher level, we could think of inferring emotional states, interpersonal relationships or understanding humour. To do so, we must learn semantic correspondences not only between auditory and visual data, but also up to the recognition level, aiming to extract semantic cues.

The fact that the auditory and visual modalities are complementary, together with the challenges associated to properly fusing them, lead to a large body of literature on fusing multi-modal (audio-visual) data, see [15]. From early studies discussing the cocktail party problem, up to very recent works aiming to detect, localise and track audio-visual events, or to enhance classically mono-modal tasks with the aid of the other modality (e.g. audio-visual speech enhancement). One of the standard questions when addressing audio-visual fusion was *where* to fuse the data: meaning at which point of the processing pipeline. Before the raise of deep architectures, i.e. when recognition and representation where to separate processes, there were two main fusion strategies. *Early fusion* referred to the strategy of mixing the auditory and visual representations before the recognition step. By contrast, *late fusion* referred to as fusing the information after the recognition step took place independently for each modality. Both strategies have advantages and drawbacks highly depending on the addressed task. With the raise of deep learning, where there is no clear separation anymore between representation and recognition, the fusion strategies can happen at various stages of the pipeline at the same time. More importantly, for the processing pipelines learned end-to-end, the fusion helps shaping both the representation and the recognition directly for the tackled task.[I.5] Therefore, the key question is then at which level of the representation is appropriate to fuse the auditory and visual features. This is a very generic question, which answer depends on the type of network used to represent auditory and visual data, and their architectures, as well as on the task. Luckily for us (researchers on learning for audio-visual processing), up to know there is no generic recipe allowing to answer this very important question.

Correctly fusing audio-visual data is not decoupled from properly representing each of the data streams. Even if audio-visual tasks can be very different from mono-modal (auditory and visual) tasks, it is important to inspire from existing works in computer vision and audio-processing to successfully propose new contributions on audio-visual fusion. In addition, given the overall motivation tied to robotic platforms, the tools and methods proposed to endow robots with social intelligence, have to be adapted to the data streamed from the robotic sensors and to the computational power. Therefore, the contributions presented in this manuscript must be understood at the cross-roads of: computer vision, audio processing, machine learning and robotics. Consequently, some of the contributions will effectively deal with audio-visual fusion, some others with mono-modal processing, and some others with challenges that are associated to robotic platforms. For each of the contributed chapters, there is a *chapter pitch* right below the abstract, providing a quick idea of the contribution. I hope you will enjoy reading the rest of the manuscript.

[I.5] This is not particular to audio-visual fusion, but a general property of all end-to-end architectures.

I.6 Document Structure

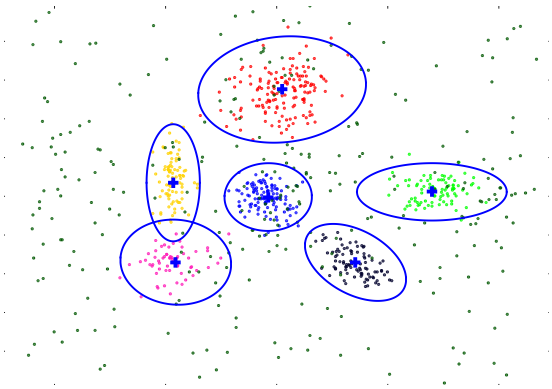
The rest of the manuscript is divided into two parts: Exact Inference and Approximate Inference. While the three chapters of the first part will follow the methodology described in Section I.3, the second part will present four contributions based on the methodologies discussed in Section I.4. This seven contributions are presented in a coherent format: the first page of each chapter provides an abstract, often an illustration, and a “chapter pitch” with the key information of the chapter. Then, the application and/or methodology are motivated in the introduction, positioned with respect to the state-of-the-art, described in detail from a methodological and algorithmic perspective, and finally evaluated with a series of experiments. After describing these contributions, one can find the overall Conclusions, and the References.

Part A

Computationally Tractable $p(\mathbf{x})$

Chapter *II*

Robust Clustering for Audio-Visual Speaker Detection



Abstract *In the clustering literature, parametric finite-mixture models play a central role due to their interesting mathematical properties and to the existence of maximum-likelihood estimators based on expectation-maximization (EM). In this chapter we propose a new mixture model that associates a weight with each observed point. We introduce the weighted-data Gaussian mixture and we derive two EM algorithms. The first one considers a fixed weight for each observation. The second one treats each weight as a random variable following a gamma distribution. We propose a model selection method based on a minimum message length criterion, provide a weight initialization strategy, and validate the proposed algorithms by comparing them with several state of the art parametric and non-parametric clustering techniques. We also demonstrate the robustness of the proposed method in the presence of data captured from different modalities, namely audio-visual speaker detection.*

Chapter Pitch

Methodological contribution A new mixture model robust to outliers, where each observation $\mathbf{x}$ has an associated hidden Gamma-distributed weight w.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \int_{\mathbb{R}^+} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_k, \frac{1}{w}\boldsymbol{\Sigma}_k\right) \mathcal{G}(w; \alpha, \beta) \mathrm{d}w.$$

Applicative task Robust clustering in general, and audio-visual speaker detection in particular.

Interesting insight In this model each sample *sees* a mixture model with the same proportions and mean vectors, but with the covariance matrices scaled with a sample-dependent weight. Therefore the sampling is independent but not identically distributed.

Dissemination This work was first published IEEE Workshop on Machine Learning for Signal Processing, and then at IEEE Transactions on Pattern Analysis and Machine Learning in 2016, see [16], [17]

II.1 Introduction

Finding significant groups in a set of data points is a central problem in many fields. Consequently, clustering has received a lot of attention, and many methods, algorithms and software packages are available today. Among these techniques, parametric finite mixture models play a paramount role, due to their interesting mathematical properties as well as to the existence of maximum likelihood estimators based on expectation-maximization (EM) algorithms. While the finite Gaussian mixture (GMM) [18] is the model of choice, it is extremely sensitive to the presence of outliers. Alternative robust models have been proposed in the statistical literature, such as mixtures of t-distributions [19] and their numerous variants, *e.g.* [20]–[25]. In contrast to the Gaussian case, no closed-form solution exists for the t-distribution and tractability is maintained via the use of EM and a Gaussian scale mixture representation, $\mathcal{T}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\alpha) = \int_0^\infty \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}/w)\mathcal{G}(w,\alpha/2,\alpha/2)dw$, where $\boldsymbol{x}$ is an observed vector, $\mathcal{N}$ is the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}/w$, and $\mathcal{G}$ is the gamma distribution of a univariate positive variable w parameterized by α. In the case of mixtures of t-distributions, with mixing coefficients π_k, $\sum_{k=1}^K \pi_k \mathcal{T}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\alpha_k)$, a latent variable w can also be introduced. Its distribution is a mixture of K gamma distributions that accounts for the component-dependent α_k [19]. Clustering is then usually performed associating a positive variable w_i, distributed as w, with **each** observed point $\boldsymbol{x}_i$. The distributions of both w_i and $\boldsymbol{x}_i$ do not depend on i. The observed data are drawn from i.i.d. variables, distributed according to the t-mixture, or one of its variants [19]–[25].

In this chapter we propose a finite mixture model in which variable w_i is used as a weight to account for the reliability of the observed $\boldsymbol{x}_i$ and this independently on its assigned cluster. The distribution of w_i is not a gamma mixture anymore but has to depend on i to allow each data point to be potentially treated differently. In contrast to mixtures of t-distributions, it follows that the observed data are independent *but not* identically distributed.

We introduce the weighted-data Gaussian mixture model (WD-GMM). We distinguish two cases, namely (i) the weights are known a priori and hence they are fixed, and (ii) the weights are modeled as variables and hence they are iteratively updated, given initial estimates. We show that in the case of fixed weights, the GMM parameters can be estimated via an extension of the standard EM which will be referred to as the *fixed weighted-data* EM algorithm (FWD-EM). Then we consider the more general case of weights that are treated as random variables. We model these variables with gamma distributions (one distribution for each variable) and we formally derive a closed-form EM algorithm which will be referred to as the *weighted-data* EM algorithm (WD-EM). While the M-step of the latter is similar to the M-step of FWD-EM, the E-step is considerably different as both the posterior probabilities (responsibilities) and the parameters of the posterior gamma distributions (the weights) are updated (E-Z-step and E-W-step).

The responsibilities are computed using the Pearson type VII distribution (the reader is referred to [22] for a discussion regarding this distribution), also called the Arellano-Valle and Bolfarine generalized t-distribution [26], and the parameters of the posterior gamma distributions are computed from the prior gamma parameters and from the Mahalanobis distance between the data and the mixture means. Note that the weights play a different role than the responsibilities. Unlike the responsibilities, which are probabilities, the weights are random variables that can take arbitrary positive values. Their posterior means can be used as an absolute measure of the relevance of the data. Typically, an outlying data point which is far from any cluster center will have a small weight while it may still be assigned with a significant responsibility value to the closest cluster. Responsibilities indicate which cluster center is the closest but not if any of them is close at all.

The idea of weighted-data clustering has already been proposed in the framework of non-parametric clustering methods such as K-means and spectral clustering, *e.g.* [27]–[30]. These methods generally propose to incorporate prior information in the clustering process in order to prohibit atypical data (outliers) to contaminate the clusters. The idea of modeling data weights as random variables and to estimate them via EM was proposed in [31] in the particular framework of Markovian brain image segmentation. In [31] it is shown that specific expert knowledge is not needed and that the data-weight distribution guide the model towards a satisfactory segmentation. A variational EM is proposed in [31] as their formulation has no closed form. In this chapter we build on the idea that, instead of relying on prior information about atypical data, *e.g.* [27]–[30], we devise a novel EM algorithm that updates the weight distributions. The proposed method belongs to the *robust clustering* category of mixture models because observed data that are far away from the cluster centers have little influence on the estimation of the means and covariances.

An important feature of mixture based clustering methods is to perform model selection on the premise that the number of components K in the mixture corresponds to the number of clusters in the data. Traditionally, model selection is performed by obtaining a set of candidate models for a range of values of K (assuming that the true value is in this range). The number of components is selected by minimizing a model selection criteria, such as the Bayesian inference criterion (BIC), minimum message length (MML), Akaike's information criteria (AIC) to cite just a few [18], [32]. The disadvantage of these methods is twofold. Firstly, a whole set of candidates has to be obtained and problems associated with running EM many times may emerge. Secondly, they provide a number of components that optimally approximate the density and not the true number of clusters present in the data. More recently, there seems to be a consensus among mixture model practitioners that a well-founded and computationally efficient model selection

strategy is to start with a large number of components and to merge them [33]. [32] proposes a practical algorithm that starts with a very large number of components (thus making the algorithm robust to initialization), iteratively annihilates components, redistributes the observations to the other components, and terminates based on the MML criterion. [34] starts with an overestimated number of components using BIC, and then merges them hierarchically according to an entropy criterion. More recently [35] proposes a similar method that merges components based on measuring their pair-wise overlap.

Another trend in handling the issue of finding the proper number of components is to consider Bayesian non-parametric mixture models. This allows the implementation of mixture models with an infinite number of components via the use of Dirichlet process mixture models. In [36], [37] an infinite Gaussian mixture (IGMM) is presented with a computationally intensive Markov Chain Monte Carlo implementation. At first glance, IGMM may appear similar to FWD-EM. However, these two algorithms are quite different. While IGMM is fully Bayesian the proposed FWD-EM is not, in the sense that no priors are assumed on the parameters, typically the means and covariance matrices. IGMM implies Student predictive distributions while FWD-EM involves only Gaussian distributions. More flexibility in the cluster shapes has been allowed by considering infinite mixture of infinite Gaussian mixtures (I^2GMM) [38]. The flexibility is however limited to a cluster composed of sub-clusters of identical shapes and orientations, which may alter the performance of this approach. Altogether, IGMM and I^2GMM are not designed to handle outliers, as illustrated in Section II.8, Figs. II.2-f and II.2-g. Infinite Student mixture models have also been considered [39], but inference requires a variational Bayes approximation which generates additional computational complexity.

Bayesian non-parametrics, although promising techniques, require a fully Bayesian setting. The latter, however, induces additional complexity for handling priors and hyper-priors, especially in a multi-variate context. In contrast, our latent variable approach allows exact inference. With respect to model selection, we therefore propose to extend the method of [32] to weighted-data mixtures. We formally derive an MML criterion for the weighted-data mixture model and we plug this criterion into an efficient algorithm which, starting with a large number of components, simultaneously estimates the model parameters, the posterior probabilities of the weights and the optimal number of components.

We also propose to apply the proposed weighted-data robust clustering method to the problem of fusing auditory and visual information. This problem arises when the task is, *e.g.* to detect a person that is both seen and heard, such as an active speaker. Single-modality signals – vision-only or audio-only – are often either weak or ambiguous, and it may be useful to combine information from different sensors, *e.g.* cameras and microphones. There are several difficulties associated with audio-visual fusion from a data clustering perspective: the two sensorial modalities (i) live in different spaces, (ii) are contaminated by different types of noise with different distributions, (iii) have different spatiotemporal distributions, and (iv) are perturbed by different physical phenomena, *e.g.* acoustic reverberations, lighting conditions, etc. For example, a speaker may face the camera while he/she is silent and may emit speech while he/she turns his/her face away from the camera. Speech signals have sparse spectro-temporal structure and they are mixed with other sound sources, such as music or background noise. Speaker faces may be totally or partially occluded, in which case face detection and localization is extremely unreliable. We show that the proposed method is well suited to find audio-visual clusters and to discriminate between speaking and silent people.

The remainder of this chapter is organized as follows. Section II.2 outlines the weighted-data mixture model; Section II.3 sketches the FWD-EM algorithm. Weights modeled with random variables are introduced in Section II.4 and the WD-EM is described in detail in Section II.5. Section II.6 details how to deal with an unknown number of clusters and Section II.7 addresses the issue of algorithm initialization. In Section II.8 the proposed algorithms are tested and compared with several other parametric and non-parametric clustering methods. Section II.9 addresses clustering of audio-visual data. Section II.10 concludes the chapter. Additional results and videos are available online.[II.1]

II.2 Gaussian Mixtures with Weighted Data

In this Section, we present the intuition and the formal definition of the proposed weighted-data model. Let $\boldsymbol{x} \in \mathbb{R}^d$ be a random vector following a multivariate Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$, namely $p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, with the notation $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. Let $w > 0$ be a weight indicating the relevance of the observation $\boldsymbol{x}$. Intuitively, higher the weight w, stronger the impact of $\boldsymbol{x}$. The weight can therefore be incorporated into the model by "*observing* $\boldsymbol{x}$ w *times*". In terms of the likelihood function, this is equivalent to raise $p(\boldsymbol{x}; \boldsymbol{\theta})$ to the power w, *i.e.* $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})^w$. However, the latter is not a probability distribution since it does not integrate to one. It is straightforward to notice that $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})^w \propto \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}/w)$. Therefore, w plays the role of the precision and is different for each datum $\boldsymbol{x}$. Subsequently, we write:

$$\hat{p}(\boldsymbol{x}; \boldsymbol{\theta}, w) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \frac{1}{w}\boldsymbol{\Sigma}\right), \tag{II.1}$$

[II.1] https://team.inria.fr/perception/research/wdgmm/

from which we derive a mixture model with K components:

$$\tilde{p}(\boldsymbol{x};\boldsymbol{\Theta},w)=\sum_{k=1}^{K}\pi_k\mathcal{N}\left(\boldsymbol{x};\boldsymbol{\mu}_k,\frac{1}{w}\boldsymbol{\Sigma}_k\right), \tag{II.2}$$

where $\boldsymbol{\Theta}=\{\pi_1,\ldots,\pi_K,\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K\}$ are the mixture parameters, $\pi_1,\ldots,\pi_K$ are the mixture coefficients satisfying $\pi_k\geq 0$ and $\sum_{k=1}^{K}\pi_k=1$, $\boldsymbol{\theta}_k=\{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}$ are the parameters of the k-th component and K is the number of components. We will refer to the model in (II.2) as the *weighted-data Gaussian mixture model* (WD-GMM). Let $\boldsymbol{X}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ be the observed data and $\boldsymbol{W}=\{w_1,\ldots,w_n\}$ be the weights associated with $\boldsymbol{X}$. We assume each $\boldsymbol{x}_i$ is independently drawn from (II.2) with $w=w_i$. The observed-data log-likelihood is:

$$\log\tilde{p}(\boldsymbol{X};\boldsymbol{\Theta},\boldsymbol{W})=\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\pi_k\mathcal{N}\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\frac{1}{w_i}\boldsymbol{\Sigma}_k\right)\right). \tag{II.3}$$

It is well known that direct maximization of the log-likelihood function is problematic in case of mixtures and that the expected complete-data log-likelihood must be considered instead. Hence, we introduce a set of n hidden (assignment) variables $\boldsymbol{Z}=\{z_1,\ldots,z_n\}$ associated with the observed variables $\boldsymbol{X}$ and such that $z_i=k, k\in\{1,\ldots,K\}$ if and only if $\boldsymbol{x}_i$ is generated by the k-th component of the mixture. In the following we first consider a fixed (given) number of mixture components K, we then extend the model to an unknown K, thus estimating the number of components from the data.

II.3 EM with Fixed Weights

The simplest case is when the weight values are provided at algorithm initialization, either using some prior knowledge or estimated from the observations (*e.g.* Section II.7), and are then kept fixed while alternating between the expectation and maximization steps. In this case, the expected complete-data log-likelihood is:

$$\mathcal{Q}_c\big(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(r)}\big)=\mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{W},\boldsymbol{\Theta}^{(r)})}\left[\log p\left(\boldsymbol{X},\boldsymbol{Z};\boldsymbol{W},\boldsymbol{\Theta}\right)\right], \tag{II.4}$$

where $\mathbb{E}_p[\cdot]$ denotes the expectation with respect to the distribution p. The $(r+1)$-th EM iteration consists of two steps namely, the evaluation of the posterior distribution given the current model parameters $\boldsymbol{\Theta}^{(r)}$ and the weights $\boldsymbol{W}$ (E-step), and the maximization of (II.4) with respect to $\boldsymbol{\Theta}$ (M-step):

$$\boldsymbol{\Theta}^{(r+1)}=\arg\max_{\boldsymbol{\Theta}}\mathcal{Q}_c\big(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(r)}\big). \tag{II.5}$$

It is straightforward to show that this yields the following FWD-EM algorithm:

II.3.1 The E-Step of FWD-GMM

The posteriors $\eta_{ik}^{(r+1)}=p(z_i=k|\boldsymbol{x}_i;w_i,\boldsymbol{\Theta}^{(r)})$ are updated with:

$$\eta_{ik}^{(r+1)}=\frac{\pi_k^{(r)}\hat{p}(\boldsymbol{x}_i;\boldsymbol{\theta}_k^{(r)},w_i)}{\tilde{p}(\boldsymbol{x}_i;\boldsymbol{\Theta}^{(r)},w_i)}, \tag{II.6}$$

where $\hat{p}$ and $\tilde{p}$ are defined in (II.1) and (II.2).

II.3.2 The M-Step of FWD-GMM

Expanding (II.4) we get:

$$\begin{aligned}\mathcal{Q}_c\big(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(r)}\big)&=\sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(r+1)}\log\pi_k\mathcal{N}\big(\boldsymbol{x}_i;\boldsymbol{\mu}_k;\frac{1}{w_i}\boldsymbol{\Sigma}_k\big)\\&\overset{\boldsymbol{\Theta}}{=}\sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(r+1)}\left(\log\pi_k-\log|\boldsymbol{\Sigma}_k|^{1/2}-\frac{w_i}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)\right),\end{aligned} \tag{II.7}$$

where $\overset{\Theta}{=}$ denotes equality up to a constant that does not depend on $\boldsymbol{\Theta}$. By canceling out the derivatives with respect to the model parameters, we obtain the following update formulae for the mixture proportions, means, and covariances matrices:

$$\pi_k^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n}\eta_{ik}^{(r+1)}, \tag{II.8}$$

$$\boldsymbol{\mu}_k^{(r+1)} = \frac{\sum_{i=1}^{n} w_i\eta_{ik}^{(r+1)}\boldsymbol{x}_i}{\sum_{i=1}^{n} w_i\eta_{ik}^{(r+1)}}, \tag{II.9}$$

$$\boldsymbol{\Sigma}_k^{(r+1)} = \frac{\sum_{i=1}^{n} w_i\eta_{ik}^{(r+1)}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{(r+1)}\right)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{(r+1)}\right)^\top}{\sum_{i=1}^{n}\eta_{ik}^{(r+1)}}. \tag{II.10}$$

II.4 Modeling the Weights

As we already remarked, the weights play the role of precisions. The notable difference between standard finite mixture models and the proposed model is that there is a different weight w_i, hence a different precision, associated with *each* observation $\boldsymbol{x}_i$. Within a Bayesian formalism, the weights $\boldsymbol{W}$ may be treated as random variables, rather than being fixed in advance, as in the previous case. Since (II.1) is a Gaussian, a convenient choice for the prior on w, $p(w)$ is the conjugate prior of the precision with known mean, *i.e.* a gamma distribution. This ensures that the weight posteriors are gamma distributions as well. Summarizing we have:

$$p(w;\boldsymbol{\phi}) = \mathcal{G}(w;\alpha,\beta) = \Gamma(\alpha)^{-1}\beta^{\alpha}w^{\alpha-1}e^{-\beta w}, \tag{II.11}$$

where $\mathcal{G}(w;\alpha,\beta)$ is the gamma distribution, $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}\mathrm{d}t$ is the gamma function, and $\boldsymbol{\phi} = \{\alpha,\beta\}$ are the parameters of the prior distribution of w. The mean and variance of the random variable w are given by:

$$\mathbb{E}[w] = \frac{\alpha}{\beta}, \qquad \text{and} \qquad \mathbb{E}[(w-\mathbb{E}(w))^2] = \frac{\alpha}{\beta^2}. \tag{II.12}$$

II.5 EM with Random Weights

In this section we derive the WD-EM algorithm associated to a model in which the weights are treated as random variables following (II.11). The gamma distribution of each w_i is assumed to be parameterized by $\phi_i = \{\alpha_i,\beta_i\}$. Within this framework, the expectation of the complete-data log-likelihood is computed over both the assignment and weight variables:

$$\mathcal{Q}_{\mathsf{R}}\big(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(r)}\big) = \mathbb{E}_{p(\boldsymbol{Z},\boldsymbol{W}|\boldsymbol{X};\boldsymbol{\Theta}^{(r)},\boldsymbol{\Phi})}[\log p(\boldsymbol{Z},\boldsymbol{W},\boldsymbol{X};\boldsymbol{\Theta},\boldsymbol{\Phi})], \tag{II.13}$$

where we used the notation $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_n\}$. We notice that the posterior distribution factorizes on i:

$$p\big(\boldsymbol{Z},\boldsymbol{W}|\boldsymbol{X};\boldsymbol{\Theta}^{(r)},\boldsymbol{\Phi}\big) = \prod_{i=1}^{n} p\big(z_i,w_i|\boldsymbol{x}_i;\boldsymbol{\Theta}^{(r)},\boldsymbol{\phi}_i\big)$$

and each one of these factors can be be decomposed as:

$$p\big(z_i,w_i|\boldsymbol{x}_i;\boldsymbol{\Theta}^{(r)},\boldsymbol{\phi}_i\big) = p(w_i|z_i,\boldsymbol{x}_i;\boldsymbol{\Theta}^{(r)},\boldsymbol{\phi}_i)p(z_i|\boldsymbol{x}_i;\boldsymbol{\Theta}^{(r)},\boldsymbol{\phi}_i), \tag{II.14}$$

where the two quantities on the right-hand side of this equation have closed-form expressions. The computation of each one of these two expressions leads to two sequential steps, the E-W-step and the E-Z-step, of the expectation step of the proposed algorithm.

II.5.1 The E-Z Step of WD-GMM

The marginal posterior distribution of z_i is obtained by integrating (II.14) over w_i. As previously, we denote the responsibilities with $\eta_{ik}^{(r+1)} = p(z_i = k|\boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i)$. The integration computes:

$$\begin{aligned}\eta_{ik}^{(r+1)} &= \int p\left(z_i = k, w_i|\boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i\right) dw_i \overset{k}{\propto} \int \pi_k^{(r)} p\left(\boldsymbol{x}_i|z_i = k, w_i; \boldsymbol{\Theta}^{(r)}\right) p(w_i; \boldsymbol{\phi}_i) dw_i \\ &= \int \pi_k^{(r)} \hat{p}\left(\boldsymbol{x}_i; \boldsymbol{\theta}_k^{(r)}, w_i\right) \mathcal{G}(w_i; \alpha_i, \beta_i) dw_i \overset{k}{\propto} \pi_k^{(r)} \mathcal{P}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(r)}, \boldsymbol{\Sigma}_k^{(r)}, \alpha_i, \beta_i),\end{aligned} \tag{II.15}$$

where $\mathcal{P}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \alpha_i, \beta_i)$ denotes the Pearson type VII probability distribution function, which can be seen as a generalization of the t-distribution:

$$\mathcal{P}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \beta) = \frac{\Gamma(\alpha + d/2)}{|\boldsymbol{\Sigma}|^{1/2}\Gamma(\alpha)(2\pi\beta)^{d/2}} \left(1 + \frac{||\boldsymbol{x} - \boldsymbol{\mu}||_{\boldsymbol{\Sigma}}^2}{2\beta}\right) \tag{II.16}$$

II.5.2 The E-W Step of WD-GMM

The posterior distribution of w_i, namely $p(w_i|z_i = k, \boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i)$ is a gamma distribution, because it is the conjugate prior of the precision of the Gaussian distribution. Therefore, we only need to compute the parameters of the posterior gamma distribution:

$$\begin{aligned}&p(w_i|z_i = k, \boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i) \overset{w_i}{\propto} p(\boldsymbol{x}_i|z_i = k, w_i; \boldsymbol{\Theta}^{(r)}) p(w_i; \boldsymbol{\phi}_i) \\ &= \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(r)}, \boldsymbol{\Sigma}_k^{(r)}/w_i) \mathcal{G}(w_i; \alpha_i, \beta_i) = \mathcal{G}(w_i; a_i^{(r+1)}, b_{ik}^{(r+1)}),\end{aligned} \tag{II.17}$$

where the parameters of the posterior gamma distribution are evaluated with:

$$a_i^{(r+1)} = \alpha_i + \frac{d}{2}, \qquad \text{and} \qquad b_{ik}^{(r+1)} = \beta_i + \frac{1}{2}||\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(r)}||_{\boldsymbol{\Sigma}_k^{(r)}}^2 \tag{II.18}$$

The conditional mean of w_i, namely $\overline{w}_{ik}^{(r+1)}$, can then be evaluated with:

$$\overline{w}_{ik}^{(r+1)} = \mathbb{E}_{p(w_i|z_i=k, \boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i)}[w_i] = \frac{a_i^{(r+1)}}{b_{ik}^{(r+1)}}. \tag{II.19}$$

While estimating the weights themselves is not needed by the algorithm, it is useful to evaluate them in order to fully characterize the observations and to discriminate between inliers and outliers. First notice that the marginal posterior distribution of w_i is a mixture of gamma distributions:

$$\begin{aligned}&p(w_i|\boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i) \sum_{k=1}^{K} p(w_i|z_i = k, \boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i) p(z_i = k|\boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i) \\ &= \sum_{k=1}^{K} \mathcal{G}(w_i; a_i^{(r+1)}, b_{ik}^{(r+1)}) \eta_{ik}^{(r+1)},\end{aligned} \tag{II.20}$$

and therefore the posterior mean of w_i is evaluated with:

$$\overline{w}_i^{(r+1)} = \mathbb{E}_{p(w_i|\boldsymbol{x}_i; \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i)}[w_i] = \sum_{k=i}^{K} \eta_{ik}^{(r+1)} \overline{w}_{ik}^{(r+1)}. \tag{II.21}$$

By inspection of (II.18) and (II.19) it is easily seen that the value of $\overline{w}_i$ decreases as the distance between the cluster centers and observation $\boldsymbol{x}_i$ increases. Importantly, the evaluation of $\overline{w}_i$ enables outlier detection. Indeed, an outlier is expected to be far from all the clusters, and therefore all $\overline{w}_{ik}$ will be small, leading to a small value of $\overline{w}_i$. It is worth noticing that this is not possible using only the responsibilities η_{ik}, since they are normalized by definition, and therefore their value is not an absolute measure of the datum's relevance, but only a relative measure of it.

II.5.3 The M-Step of WD-GMM

This step maximizes the expected complete-data log-likelihood over the mixture parameters. By expanding (II.13), we have:

$$\begin{aligned}\mathcal{Q}_{\mathsf{R}}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(r)}) &\overset{\boldsymbol{\Theta}}{=} \sum_{i=1}^{n}\sum_{k=1}^{K}\int_{w_i} \eta_{ik}^{(r+1)} \log \pi_k \mathcal{N}\Big(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \frac{1}{w_i}\boldsymbol{\Sigma}_k\Big) p(w_i|\boldsymbol{x}_i, z_i = k, \boldsymbol{\Theta}^{(r)}, \boldsymbol{\phi}_i)\mathrm{d}w_i \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \eta_{ik}^{(r+1)} \Big(\log \pi_k - \log |\boldsymbol{\Sigma}_k|^{1/2} - \frac{\overline{w}_{ik}^{(r+1)}}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\Big). \end{aligned} \tag{II.22}$$

The parameter updates are obtained from canceling out the derivatives of the expected complete-data log-likelihood (II.22). As with standard Gaussian mixtures, all the updates are closed-form expressions:

$$\pi_k^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n} \eta_{ik}^{(r+1)}, \tag{II.23}$$

$$\boldsymbol{\mu}_k^{(r+1)} = \frac{\sum_{i=1}^{n} \overline{w}_{ik}^{(r+1)} \eta_{ik}^{(r+1)} \boldsymbol{x}_i}{\sum_{i=1}^{n} \overline{w}_{ik}^{(r+1)} \eta_{ik}^{(r+1)}}, \tag{II.24}$$

$$\boldsymbol{\Sigma}_k^{(r+1)} = \frac{\sum_{i=1}^{n} \eta_{ik}^{(r+1)} \overline{w}_{ik}^{(r+1)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(r+1)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(r+1)}\right)^\top}{\sum_{i=1}^{n} \eta_{ik}^{(r+1)}}. \tag{II.25}$$

It is worth noticing that the M-step of the WD-EM algorithm is very similar to the M-step of the FWD-EM algorithm (section II.3). Indeed, the above iterative formulas, (II.23), (II.24), (II.25) are identical to the formulas (II.8), (II.9), (II.10), except that the fixed weights w_i are here replaced with the posterior means of the random weights, $\overline{w}_{ik}^{(r+1)}$.

II.6 Estimating the Number of Components

So far it has been assumed that the number of mixture components K is provided in advance. This assumption is unrealistic for most real-world applications. In this Section we propose to extend the method and algorithm proposed in [32] to the weighted-data clustering model. An interesting feature of this model selection method is that it does not require parameter estimation for many different values of K, as it would be the case with the Bayesian information criterion (BIC) [40]. Instead, the algorithm starts with a large number of components and iteratively deletes components as they become irrelevant. Starting with a large number of components has the additional advantage of making the algorithm robust to initialization. Formally, the parameter estimation problem is cast into a transmission encoding problem and the criterion is to minimize the expected length of the message to be transmitted:

$$\mathrm{length}(\boldsymbol{X}, \boldsymbol{\Theta}) = \mathrm{length}(\boldsymbol{\Theta}) + \mathrm{length}(\boldsymbol{X}|\boldsymbol{\Theta}). \tag{II.26}$$

In this context, the observations and the parameters have to be quantized to finite precision before the transmission. This quantization sets a trade off between the two terms of the previous equation. Indeed, when truncating to high precision, $\mathrm{length}(\boldsymbol{\Theta})$ may be long, but $\mathrm{length}(\boldsymbol{X}|\boldsymbol{\Theta})$ will be short, since the parameters fit well the data. Conversely, if the quantization is coarse, $\mathrm{length}(\boldsymbol{\Theta})$ may be short, but $\mathrm{length}(\boldsymbol{X}|\boldsymbol{\Theta})$ will be long. The optimal quantization step can be found by means of the Taylor approximation [32]. In that case, the optimization problem corresponding to the *minimum message length* (MML) criterion, is:

$$\boldsymbol{\Theta}_{\mathrm{MML}} = \underset{\boldsymbol{\Theta}}{\mathrm{argmin}} \Big\{ -\log p(\boldsymbol{\Theta}) - \log p(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{\Phi}) + \frac{1}{2}\log|\mathbf{I}(\boldsymbol{\Theta})| + \frac{\mathcal{D}(\boldsymbol{\Theta})}{2}\Big(1 + \log\frac{1}{12}\Big)\Big\}, \tag{II.27}$$

where $\mathbf{I}(\boldsymbol{\Theta}) = -\mathbb{E}\{D^2_{\boldsymbol{\Theta}} \log p(\boldsymbol{X}|\boldsymbol{\Theta})\}$ is the *expected* Fisher information matrix (FIM) and $\mathcal{D}(\boldsymbol{\Theta})$ denotes the dimensionality of the model, namely the dimension of the parameter vector $\boldsymbol{\Theta}$. Since the minimization (II.27) does not depend on the weight parameters, $\boldsymbol{\Phi}$ will be omitted for simplicity.

In our particular case, as in the general case of mixtures, the Fisher information matrix cannot be obtained analytically. Indeed, the direct optimization of the log-likelihood does not lead to closed-form solutions. Nevertheless, it was

noticed that the *complete* FIM upper bounds the FIM [32], and that the expected complete-data log-likelihood lower bounds the log-likelihood. This allows us to write the following equivalent optimization problem:

$$\boldsymbol{\Theta}_{\text{MML}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \Big\{ -\log p(\boldsymbol{\Theta}) - \log \mathcal{Q}_{\mathsf{R}}\big(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(r)}\big) + \frac{1}{2}\log|\mathbf{I}_c(\boldsymbol{\Theta})| + \frac{\mathcal{D}(\boldsymbol{\Theta})}{2}\Big(1 + \log\frac{1}{12}\Big)\Big\}, \tag{II.28}$$

where $\mathbf{I}_c$ denotes the expected complete-FIM and $\mathcal{Q}_{\mathsf{R}}$ is evaluated with (II.22).

As already mentioned, because there is a different weight w_i for each observation i, the observed data are not identically distributed and our model cannot be considered a classical mixture model. For this reason, the algorithm proposed in [32] cannot be applied directly to our model. Indeed, in the proposed WD-GMM setting, the complete-FIM is:

$$\mathbf{I}_c(\boldsymbol{\Theta}) = \operatorname{diag}\Big(\pi_1\sum_{i=1}^{n}\mathbf{I}_i(\boldsymbol{\theta}_1), \ldots, \pi_K\sum_{i=1}^{n}\mathbf{I}_i(\boldsymbol{\theta}_K), n\mathbf{M}\Big) \tag{II.29}$$

where $\mathbf{I}_i(\boldsymbol{\theta}_k) = -\mathbb{E}\{D^2_{\boldsymbol{\theta}_k}\log\mathcal{P}(\boldsymbol{x}_i|\boldsymbol{\theta}_k, \alpha_i, \beta_i)\}$ is the Fisher information matrix for the i-th observation with respect to the parameter vector $\boldsymbol{\theta}_k$ (mean and the covariance) of the k-th component, $\mathcal{P}$ is defined in (II.16), and $\mathbf{M}$ is the Fisher information matrix of the multinomial distribution, namely the diagonal matrix $\operatorname{diag}(\pi_1^{-1}, \ldots, \pi_K^{-1})$. We can evaluate $|\mathbf{I}_c(\boldsymbol{\Theta})|$ from (II.29):

$$|\mathbf{I}_c(\boldsymbol{\Theta})| = n^{K(M+1)}|\mathbf{M}|\prod_{k=1}^{K}\pi_k^M\left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_i(\boldsymbol{\theta}_k)\right|, \tag{II.30}$$

where M denotes the number of free parameters of each component. For example, $M = 2d$ when using diagonal covariance matrices or $M = d(d+3)/2$ when using full covariance matrices.

Importantly, one of the main advantages of the methodology proposed in [32] is that one has complete freedom to choose a prior distribution on the parameters, $p(\boldsymbol{\Theta})$. In our case, inspired by (II.30), we select the following prior distributions for the parameters:

$$p(\boldsymbol{\theta}_k) \propto \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_i(\boldsymbol{\theta}_k)\right|^{-\frac{1}{2}}, \tag{II.31}$$

$$p(\pi_1, \ldots, \pi_K) \propto |\mathbf{M}|^{-\frac{1}{2}}. \tag{II.32}$$

By substitution of (II.30)–(II.32) into (II.28) we obtain the following optimization problem:

$$\boldsymbol{\Theta}_{\text{MML}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \Big\{ \frac{M}{2}\sum_{k=1}^{K}\log\pi_k - \log\mathcal{Q}_{\mathsf{R}}\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(r)}\right) + \frac{K(M+1)}{2}\left(1 + \log\frac{n}{12}\right)\Big\}, \tag{II.33}$$

where we used $\mathcal{D}(\boldsymbol{\Theta}) = K(M+1)$.

One may notice that (II.33) does not make sense (diverges) if any of the π_k's is allowed to be null. However, in the current length coding framework, there is no point in transmitting the parameters of an empty component. Therefore, we only focus on the non-empty components, namely those components for which $\pi_k > 0$. Let $\mathcal{K}^+$ denote the index set of non-empty components and let $K^+ = |\mathcal{K}^+|$ be its cardinality. We can rewrite (II.33) as:

$$\boldsymbol{\Theta}_{\text{MML}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\Big\{ \frac{M}{2}\sum_{k\in\mathcal{K}^+}\log\pi_k - \log\mathcal{Q}_{\mathsf{R}}\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(r)}\right) + \frac{K^+(M+1)}{2}\left(1 + \log\frac{n}{12}\right)\Big\}. \tag{II.34}$$

The above minimization problem can be solved by modifying the EM algorithm described in Section II.5 (notice that there is an equivalent derivation for the fixed-weigth EM algorithm described in Section II.3). Indeed, we remark that the minimization (II.34) is equivalent to using a symmetric improper Dirichlet prior for the proportions with exponent $-M/2$. Moreover, since the optimization function for the parameters of the Gaussian components is the same (equivalently, we used a flat prior for the mean vector and covariance matrix), their estimation formulas (II.24) and (II.25) still hold. Therefore, we only need to modify the estimation of the mixture proportions, namely:

$$\pi_k = \frac{\max\left\{0, \sum_{i=1}^{n}\eta_{ik} - \frac{M}{2}\right\}}{\sum_{k'=1}^{K}\max\left\{0, \sum_{i=1}^{n}\eta_{ik'} - \frac{M}{2}\right\}}. \tag{II.35}$$

The $\max$ operator in (II.35) verifies whether the k-th component is supported by the data. When one of the components becomes too weak, *i.e.* the required minimum support $M/2$ cannot be obtained from the data, this component is annihilated. In other words, its parameters will not be estimated, since there is no need in transmitting them. One has to be careful in this context, since starting with a large value of K may lead to several empty components. In order

Algorithm 1: WD-EM with model selection based on the MML criterion.

input : $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n, K_{\boldsymbol{y}}, K_{\text{high}}, \boldsymbol{\Theta}^{(0)} = \{\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}\}_{K=1}^{K_{\text{high}}}, \boldsymbol{\Phi}^{(0)} = \{\alpha_i^{(0)}, \beta_i^{(0)}\}_{i=1}^n$
output: The minimum length mixture model: $\boldsymbol{\Theta}_{\min}$ and the final data weights: $\boldsymbol{W}_{\min}$

Set: $r = 0, \mathcal{K}^+ = \{k\}_{k=1}^{K_{\text{high}}}, \mathbf{LEN}_{\min} = +\infty$ **while** $|\mathcal{K}^+| \geq K_{\boldsymbol{y}}$ **do**
  repeat
    for $k = 1$ **to** K_{high} **do**
      E-Z step using (II.15): $\eta_{ik}^{(r+1)} = \frac{\pi_k^{(r)} \mathcal{P}\left(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(r)}, \boldsymbol{\Sigma}_k^{(r)}, \alpha_i^{(r)}, \beta_{ik}^{(r)}\right)}{\sum_{l=1}^{K_{\text{high}}} \pi_l^{(r)} \mathcal{P}\left(\boldsymbol{x}_i; \boldsymbol{\mu}_l^{(r)}, \boldsymbol{\Sigma}_l^{(r)}, \alpha_i^{(r)}, \beta_{il}^{(r)}\right)}$
      E-W step using (II.18): $\alpha_i^{(r+1)} = \alpha_i^{(0)} + \frac{d}{2}$ $\beta_{ik}^{(r+1)} = \beta_i^{(0)} + \frac{1}{2}||\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(r)}||^2_{\boldsymbol{\Sigma}_k^{(r)}}$ $\overline{w}_{ik} = \frac{\alpha_i^{(r+1)}}{\beta_{ik}^{(r+1)}}$
      M-step $\pi_k^{(r+1)} = \frac{\max\left\{0, \sum_{i=1}^n \eta_{ik}^{(r+1)} - \frac{M}{2}\right\}}{\sum_{l=1}^{K_{\text{high}}} \max\left\{0, \sum_{i=1}^n \eta_{il}^{(r+1)} - \frac{M}{2}\right\}}$ **if** $\pi_k^{(r+1)} > 0$ **then**
        Evaluate $\boldsymbol{\theta}_k^{(r+1)}$: mean $\boldsymbol{\mu}_k^{(r+1)}$ using (II.24) and covariance $\boldsymbol{\Sigma}_k^{(r+1)}$ using (II.25).
      else
        $K^+ = K^+ - 1$
      end
    end
    $\boldsymbol{\Theta}^{(r+1)} = \left\{\pi_k^{(r+1)}, \boldsymbol{\theta}_k^{(r+1)}\right\}_{k=1}^{K_{\text{high}}}$ Compute optimal length $\mathbf{LEN}_{\text{MML}}^{(r+1)}$ with (II.34). $r \leftarrow r + 1$
  until $|\Delta \mathbf{LEN}_{MML}^{(r)}| < \varepsilon$
  if $\mathbf{LEN}_{MML}^{(r)} < \mathbf{LEN}_{\min}$ **then**
    $\mathbf{LEN}_{\min} = \mathbf{LEN}_{\text{MML}}^{(r)}$ $\boldsymbol{\Theta}_{\min} = \boldsymbol{\Theta}^{(r)}$ $\boldsymbol{W}_{\min} = \{\overline{w}_i\}_{i=1}^n$ with $\overline{w}_i = \sum_{k=1}^{K_{\text{high}}} \eta_{ik} \overline{w}_{ik}$
  end
  $k^* = \text{argmin}_{k' \in \mathcal{K}^+}\left(\pi_{k'}^{(r)}\right)$, $\mathcal{K}^+ = \mathcal{K}^+ / k^*$
end

to avoid this singular situation, we adopt the component-wise EM procedure (CEM) [41], as proposed in [32] as well. Intuitively, we run both E and M steps for one component, before moving to the next component. More precisely, after running the E-Z and E-W steps for the component k, its parameters are updated if $k \in \mathcal{K}^+$, otherwise the component is annihilated if $k \notin \mathcal{K}^+$. The rationale behind this procedure is that, when a component is annihilated its probability mass is immediately redistributed among the remaining components. Summarizing, CEM updates the components one by one, whereas the classical EM simultaneously updates all the components.

The proposed algorithm is outlined in Algorithm 1. In practice, an upper and a lower number of components, K_{high} and $K_{\boldsymbol{y}}$, are provided. Each iteration r of the algorithm consists of component-wise E and M steps. If needed, some of the components are annihilated, and the parameters are updated accordingly, until the relative length difference is below a threshold, $|\Delta \mathbf{LEN}_{\text{MML}}^{(r)}| < \varepsilon$. In that case, if the message length, *i.e.* (II.34) is lower than the current optimum, the parameters, weights, and length are saved in $\boldsymbol{\Theta}_{\min}$, $W_{\min}$ and $\mathbf{LEN}_{\min}$ respectively. In order to explore the full range of K, the less populated component is artificially annihilated, and CEM is run again. The complexity of Algorithm 1 is similar to the complexity of the algorithm in [32], with the exception of the E-W step. However, the most computationally intensive part of this step (matrix inversion and matrix-vector multiplications in (II.18)) is already achieved in the E-Z step.

II.7 Algorithm Initialization

The EM algorithms proposed in Section II.3, Section II.5, and Section II.6 require proper initialization of both the weights (one for each observation and either a fixed value w_i or parameters α_i, β_i) and of the model parameters. The K-means algorithm is used for an initial clustering, from which values for the model parameters are computed. In this section we concentrate onto the issue of weight initialization. An interesting feature of our method is that the only constraint on the weights is that they must be positive. Initial w_i values may depend on expert or prior knowledge and may be experiment- or goal-dependent. This model flexibility allows the incorporation of such prior knowledge. In the absence of any prior information/knowledge, we propose a data-driven initialization scheme and make the assumption that densely sampled regions are more important that sparsely sampled ones. We note that a similar strategy could be

used if one wants to reduce the importance of dense data and to give more importance to small groups of data or to sparse data.

We adopt a well known data similarity measure based on the Gaussian kernel, and it follows that the weight w_i associated with the data point i is evaluated with:

$$w_i = \sum_{j \in \mathcal{S}_i^q} \exp\left(-\frac{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma}\right), \tag{II.36}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance, $\mathcal{S}_i^q$ denotes the set containing the q nearest neighbors of $\boldsymbol{x}_i$, and σ is a positive scalar. In all the experiments we used $q = 20$ for the simulated datasets and $q = 50$ for the real datasets. In both cases, we used $\sigma = 100$. In the case of the FWD-EM algorithm, the weights w_i thus initialized remain unchanged. However, in the case of the WD-EM algorithm, the weights are modeled as latent random variables drawn from a gamma distribution, hence one needs to set initial values for the parameters of this distribution, namely α_i and β_i in (II.11). Using (II.12) one can choose to initialize these parameters such as $\alpha_i = w_i^2$ and $\beta_i = w_i$, such that the mean and variance of the prior distribution are w_i and 1 respectively.

II.8 Experiments

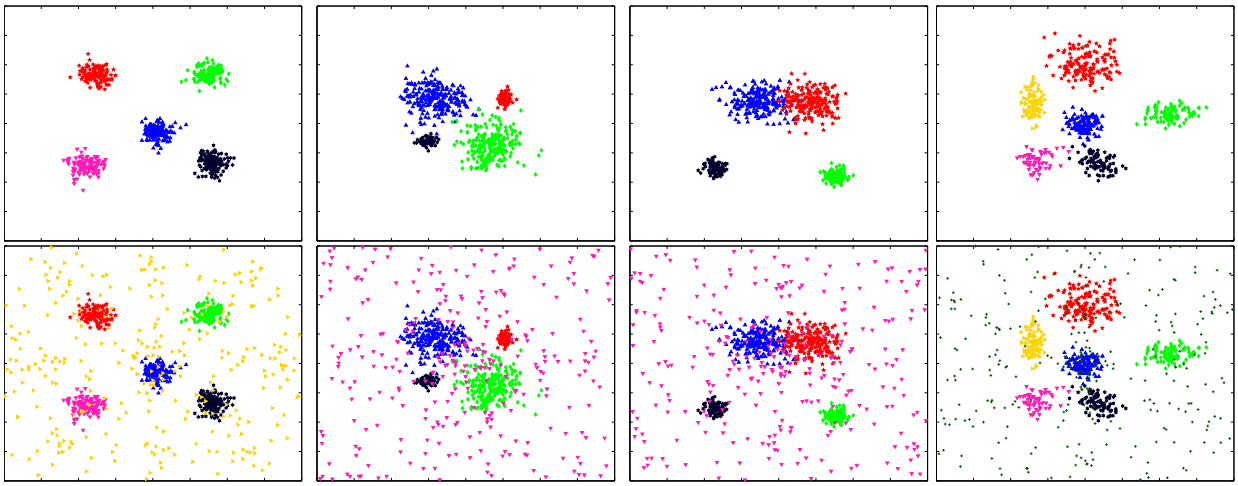


Figure II.1: *Samples of the SIM dataset with no outliers (top row) and contaminated with 50% outliers (bottom row). The 600 inliers are generated from Gaussian mixtures while the 300 outliers are generated from a uniform distribution.*

The proposed algorithms were tested and evaluated using eight datasets: four simulated datasets and four publicly available datasets that are widely used for benchmarking clustering methods. The main characteristics of these datasets are summarized in Table II.1. The simulated datasets (SIM) are designed to evaluate the robustness of the proposed method with respect to outliers. The simulated inliers are drawn from Gaussian mixtures while the simulated outliers are drawn from a uniform distribution, *e.g.* Figure II.1. The SIM datasets have different cluster configurations in terms of separability, shape and compactness. The eight datasets that we used are the following:

- **SIM-Easy:** Five clusters that are well separated and compact.

Table II.1: *wgmm/images/Datasets used for benchmarking and their characteristics: n is the number of data points, d is the dimension of the data space, and K is number of clusters.*

Data Set	n	d	K
SIM-Easy	600	2	5
SIM-Unbalanced	600	2	4
SIM-Overlapped	600	2	4
SIM-Mixed	600	2	6
MNIST [42]	10,000	141	10
Wav [43]	5,000	21	3
BCW [44]	569	30	2
Letter Recognition [45]	20,000	16	26

- **SIM-Unbalanced:** Four clusters of different size and density.
- **SIM-Overlapped:** Four clusters, two of them overlap.
- **SIM-Mixed:** Six clusters of different size, compactness and shape.
- **MNIST** contains instances of handwritten digit images normalized to the same size [42]. We preprocessed these data with PCA to reduce the dimension from 784 to 141, by keeping 95% of the variance.
- **Wav** is the Waveform Database Generator [43].
- **BCW** refers to the Breast Cancer Wisconsin data set [44], in which each instance represents a digitized image of a fine needle aspirate (FNA) of breast mass.
- **Letter Recognition** contains $20,000$ single-letter images that were generated by randomly distorting the images of the 26 uppercase letters from 20 different commercial fonts [45]. Each letter/image is described by 16 features. This dataset is available through the UCI machine learning repository.

Table II.2: *Results obtained with the MNIST, WAV, BCW, and Letter Recognition datasets. The clustering scores correspond to the Davies-Bouldin (DB) index. The best results are shown in underlined **bold**, and the second best results are shown in **bold**. The proposed method yields the best results for the WAV and BCW datasets, while I²GMM yields the best results for the MNIST dataset. Interestingly, the non-parametric methods (K-means, HAC and Ncut) yield excellent results for Letter Recognition.*

Dataset	WD-EM	FWD-EM	GMM	GMM+U	FM-uMST	IGMM	I²GMM	K-Means	KK-Means	Ncut	HAC
MNIST	2.965(0.15)	3.104(0.21)	3.291(0.14)	3.245(0.09)	**2.443(0.00)**	3.555(0.06)	**2.430(0.14)**	2.986(0.01)	2.980(0.02)	4.760(0.08)	3.178(0.00)
WAV	**0.975(0.00)**	1.019(0.00)	1.448(0.03)	1.026(0.04)	1.094(0.10)	1.028(0.02)	2.537(0.35)	1.020(0.00)	**0.975(0.05)**	2.781(0.06)	1.089(0.00)
BCW	**0.622(0.00)**	0.687(0.00)	0.714(0.00)	0.689(0.00)	0.727(0.00)	0.719(0.00)	0.736(0.09)	0.659(0.00)	**0.655(0.00)**	0.838(0.00)	0.685(0.00)
Letter Recognition	1.690(0.00)	1.767(0.01)	2.064(0.06)	2.064(0.06)	1.837(0.00)	2.341(0.11)	1.724(0.03)	**1.450(0.02)**	1.504(0.03)	**1.626(0.00)**	**1.626(0.00)**

Table II.3: *Micro F_1 scores obtained on the real data sets (MNIST, WAV, BCW and Letter Recognition). The number in parenthesis indicates the standard deviation of 20 repetitions. Based on this classification score, I²GMM yields the best result.*

Data set	WD-EM	FWD-EM	GMM	GMM+U	FM-uMST	IGMM	I²GMM	K-Means	KK-Means	Ncut	HAC
MNIST	0.524(0.01)	0.455(0.01)	**0.573(0.00)**	0.549(0.01)	0.519(0.00)	**0.689(0.02)**	0.545(0.06)	0.497(0.02)	0.507(0.02)	0.402(0.00)	0.532(0.00)
WAV	**0.774(0.00)**	0.534(0.00)	0.535(0.00)	0.552(0.00)	**0.632(0.08)**	0.543(0.01)	0.493(0.00)	0.521(0.00)	0.522(0.00)	0.387(0.00)	0.597(0.00)
BCW	**0.965(0.00)**	0.907(0.00)	0.885(0.00)	0.915(0.00)	**0.927(0.00)**	0.914(0.00)	0.682(0.00)	0.907(0.00)	0.910(0.00)	0.859(0.00)	0.879(0.00)
Letter Recognition	0.315(0.01)	0.323(0.00)	**0.423(0.00)**	**0.423(0.00)**	0.379(0.00)	0.306(0.02)	**0.466(0.01)**	0.340(0.00)	0.343(0.01)	0.347(0.00)	0.347(0.00)

In addition to the two proposed methods (FWD-EM and WD-EM) we tested the following algorithms:

- **GMM** uses EM with the standard Gaussian mixture model, implemented as described in [46];
- **GMM+U** uses EM with a GMM and with an additional uniform component, [47];
- **FM-uMST** stands for the *finite mixture of unrestricted multivariate skew t-distribution* algorithm of [25];
- **IGMM** stands for the *infinite Gaussian mixture model* [36];
- **I²GMM** stands for the *infinite mixture of infinite Gaussian mixtures* [38];
- **K-Means** is the standard K-means algorithm;
- **KK-Means** is the kernel K-means algorithm of [48];
- **NCUT** is the spectral clustering algorithm of [49].
- **HAC** is the hierarchical agglomerative clustering algorithm of [50].

All the above algorithms need proper initialization. All the mixture-based algorithms, WD-EM, FWD-EM, GMM, GMM+U, FM-uMST, IGMM and I²GMM start from the same proportions, means, and covariances which are estimated from the set of clusters provided by K-means. The latter is randomly initialized several times to find a good initialization. Furthermore, algorithms WD-EM, FWD-EM, GMM, GMM+U and FM-uMST are iterated until convergence, i.e, the log-likelihood difference between two consecutive iterations is less than 1%, or are stopped after 400 iterations.

Table II.4: *DB scores obtained on the SIM-X dataset (**best** and **second best**).*

	Outliers	WD-EM	FWD-EM	GMM	GMM+U	FM-uMST	IGMM	I^{2}GMM	K-Means	KK-Means	Ncut	HAC
SIM-Easy	10%	**0.229(0.01)**	0.295(0.01)	0.295(0.01)	**0.222(0.02)**	0.307(0.02)	1.974(0.12)	0.500(0.16)	0.291(0.01)	0.330(0.07)	0.283(0.01)	0.266(0.00)
	20%	**0.266(0.02)**	0.338(0.01)	0.342(0.01)	**0.233(0.01)**	0.349(0.02)	1.564(0.43)	0.626(0.28)	0.344(0.01)	0.420(0.10)	0.335(0.01)	0.330(0.01)
	30%	**0.330(0.01)**	0.385(0.01)	0.384(0.02)	**0.227(0.02)**	0.501(0.04)	1.296(0.12)	0.570(0.27)	0.372(0.01)	0.381(0.03)	0.366(0.02)	0.376(0.01)
	40%	**0.358(0.01)**	0.445(0.04)	0.453(0.05)	**0.211(0.02)**	0.585(0.06)	1.259(0.16)	0.534(0.21)	0.417(0.01)	0.411(0.01)	0.409(0.01)	0.401(0.01)
	50%	**0.380(0.01)**	0.455(0.02)	0.459(0.02)	**0.195(0.01)**	0.568(0.05)	1.107(0.06)	0.626(0.21)	0.422(0.01)	0.439(0.03)	0.422(0.01)	0.438(0.01)
SIM-Unbalanced	10%	**0.270(0.01)**	0.954(0.72)	1.354(1.02)	**0.277(0.01)**	1.104(0.76)	1.844(0.29)	0.491(0.17)	0.405(0.02)	0.433(0.05)	0.402(0.02)	0.427(0.02)
	20%	**0.329(0.03)**	4.503(4.33)	3.003(1.85)	**0.269(0.01)**	1.181(0.44)	1.278(0.45)	0.591(0.13)	0.512(0.02)	0.515(0.03)	0.477(0.03)	0.529(0.02)
	30%	**0.399(0.03)**	3.502(3.09)	2.034(1.22)	**0.252(0.03)**	1.414(0.88)	1.272(0.35)	0.601(0.10)	0.548(0.03)	0.540(0.03)	0.531(0.02)	0.570(0.03)
	40%	**0.534(0.13)**	2.756(2.33)	2.097(1.15)	**0.251(0.02)**	1.650(0.94)	1.239(0.36)	0.615(0.05)	0.557(0.03)	0.567(0.02)	0.563(0.02)	0.597(0.02)
	50%	**0.557(0.10)**	2.400(1.44)	1.520(0.38)	**0.268(0.01)**	1.612(0.69)	1.144(0.36)	0.665(0.10)	0.580(0.03)	0.585(0.03)	0.583(0.03)	0.636(0.02)
SIM-Overlapped	10%	**0.305(0.02)**	0.693(0.31)	1.510(0.97)	**0.307(0.02)**	1.373(0.63)	2.168(0.20)	0.554(0.14)	0.395(0.03)	0.428(0.06)	0.385(0.01)	0.427(0.01)
	20%	**0.368(0.03)**	1.562(0.45)	1.881(0.50)	**0.293(0.01)**	2.702(1.28)	1.837(0.37)	0.608(0.08)	0.467(0.02)	0.532(0.07)	0.440(0.02)	0.502(0.01)
	30%	**0.472(0.04)**	1.825(0.55)	2.209(0.64)	**0.294(0.03)**	5.101(1.99)	1.568(0.61)	0.586(0.15)	0.532(0.02)	0.521(0.03)	0.508(0.01)	0.557(0.01)
	40%	0.549(0.04)	2.372(0.54)	2.597(0.73)	**0.322(0.01)**	4.569(1.72)	1.320(0.40)	0.687(0.11)	0.546(0.02)	0.556(0.03)	**0.541(0.03)**	0.593(0.02)
	50%	0.641(0.06)	2.269(0.44)	2.247(0.60)	**0.298(0.02)**	5.762(3.34)	1.174(0.25)	0.815(0.12)	0.563(0.03)	0.576(0.02)	**0.560(0.03)**	0.618(0.02)
SIM-Mixed	10%	**0.282(0.01)**	0.443(0.11)	0.448(0.11)	0.290(0.01)	0.951(0.35)	2.032(0.46)	0.414(0.12)	0.358(0.01)	0.418(0.06)	0.359(0.01)	0.355(0.01)
	20%	**0.351(0.02)**	0.857(0.52)	1.325(0.79)	**0.286(0.01)**	1.062(0.38)	1.782(0.44)	0.462(0.08)	0.413(0.02)	0.476(0.06)	0.409(0.01)	0.428(0.01)
	30%	**0.396(0.02)**	1.368(0.74)	1.524(0.64)	**0.278(0.01)**	1.693(0.56)	1.627(0.54)	0.483(0.07)	0.454(0.02)	0.464(0.04)	0.449(0.01)	0.468(0.01)
	40%	**0.449(0.03)**	1.100(0.61)	1.188(0.59)	**0.277(0.02)**	1.609(0.43)	1.456(0.34)	0.483(0.05)	0.478(0.02)	0.504(0.04)	0.478(0.01)	0.508(0.02)
	50%	**0.492(0.03)**	1.364(0.59)	1.513(0.67)	**0.265(0.01)**	1.972(0.86)	1.366(0.29)	0.562(0.04)	0.501(0.01)	0.515(0.02)	0.499(0.02)	0.546(0.02)

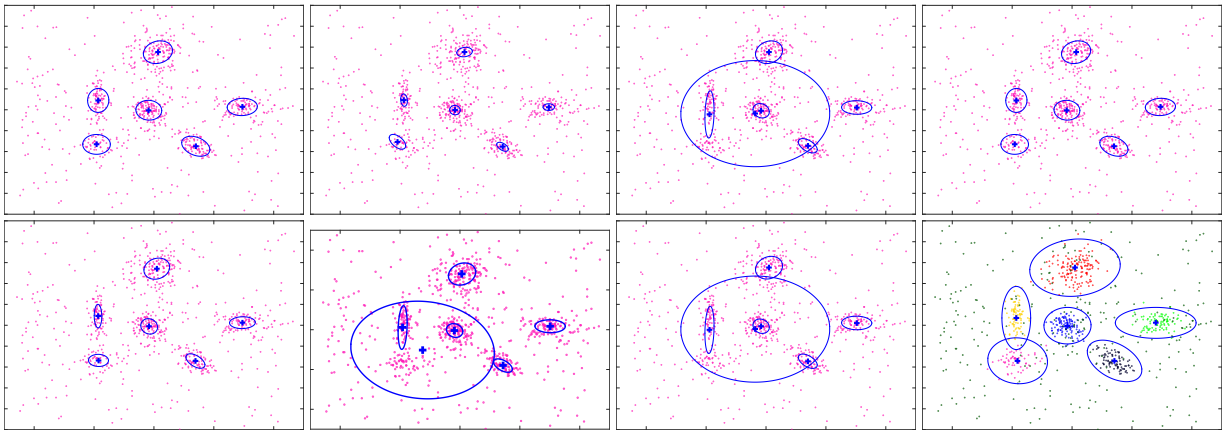


Figure II.2: *Results obtained by fitting mixture models to the SIM-Mixed data in the presence of 50% outliers (see Table II.4). First row (left to right): WD-EM, FWD-EM, GMM, GMM+U. Second row (left to right):: FM-uMST, IGMM, I^{2}GMM, Ground truth.*

To quantitatively evaluate all the tested methods, we chose to use the Davies-Bouldin (DB) index [51]:

$$DB = \frac{1}{K}\sum_{k=1}^{K} R_k, \tag{II.37}$$

where $R_k = \max_{k,k\neq l}\{(S_k + S_l)/d_{kl}\}$, $S_k = n_k^{-1}\sum_{\boldsymbol{x}\in C_k}\|\boldsymbol{x} - \boldsymbol{\mu}_k\|$ is the cluster scatter, n_k is the number of samples in cluster k, $\boldsymbol{\mu}_k$ is the cluster center, and $d_{kl} = \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|$. A low value of the DB index means that the clusters are far from each other with respect to their scatter, and therefore the discriminative power is higher. Since the algorithms are randomly initialized, we repeat each experiment 20 times and compute the mean and standard deviation of the DB index for each experiment. Table II.2 summarizes the results obtained with the MNIST, WAV, BCW, and Letter Recognition datasets. The proposed WD-EM method yields the best results for the WAV and BCW data, while the I^{2}GMM method yields the best results for the MNIST data. It is interesting to notice that the non-parametric methods K-means, NCUT and HAC yield the best and second best results for the Letter Recognition data.

For completeness we also provide the micro F_1 scores (also used in [38]) obtained with the MNIST, WAV, BCW and Letter Recognition datasets in Table II.3. Based on this classification score, the proposed WD-EM method yields the best results for the WAV and BCW data, while the I^{2}GMM yields the best results for the Letter Recognition data, and

the IGMM method yields the best results for the MNIST data. This comparison also shows that I^{2}GMM, GMM and GMM+U yield similar scores.

An interesting feature of the proposed weighted-data clustering algorithms is their robustness in finding good clusters in the presence of outliers. To illustrate this ability we ran a large number of experiments by adding outliers, drawn from a uniform distribution, to the four simulated datasets, *e.g.* Table II.4 and Figure II.2. A comparison between WD-EM, FWD-EM, and the state-of-art clustering techniques mentioned above, with different percentages of outliers, is provided. As it can be easily observed in these tables, GMM+U performs extremely well in the presence of outliers, which is not surprising since the simulated outliers are drawn from a uniform distribution. Overall, the proposed WD-EM method is the second best performing method. Notice the very good performance of the Ncut method for the SIM-overlapped data. Among all these methods, only GMM+U and WD-EM offer the possibility to characterize the outliers using two very different strategies.The GMM+U model simply pulls them in an *outlier class* based on the posterior probabilities. The WD-EM algorithm iteratively updates the posterior probabilities of the weights, and the final posteriors, (II.17), allow to implement a simple outlier detection mechanism. Another important remark is that WD-EM systematically outperforms FWD-EM, which fully justifies the proposed weighted-data model. Figure II.2 shows results of fitting the mixture models to SIM-mixed data drawn from a Gaussian mixture and contaminated with 50% outliers drawn from a uniform distribution. These plots show that GMM, IGMM, and I^{2}GMM find five components corresponding to data clusters while they also fit a component onto the outliers, roughly centered on the data set.

II.9 Audio-Visual Clustering

In this section we illustrate the effectiveness of our method to deal with audio-visual data which belong to the heterogenous type of data, *i.e.* gathered with different sensors, having different noise statistics, and different sources of errors. Prior to clustering one needs to represent audio and visual observations in the same Euclidean space, *e.g.* Figure II.3. Without loss of generality we adopt the sound-source localization method of [52] that performs 2D direction of arrival (DOA) estimation followed by mapping the estimated sound-source direction onto the image plane: a DOA estimate therefore corresponds to a pixel location in the image plane. To find visual features, we use an upper-body detector [53] that provides an approximate localization of human heads, followed by lip localization using facial landmark detection [54]. The rationale of combining upper-body detection with facial landmark localization is that, altogether this yields a detection and localization algorithm that is much more robust to head pose than the vast majority of face detection methods.

Let $\boldsymbol{A} = \{\boldsymbol{a}_j\}_{j=1}^{n_a} \in \mathbb{R}^2$ and $\boldsymbol{V} = \{\boldsymbol{v}_j\}_{j=1}^{n_v} \in \mathbb{R}^2$ denote the set of auditory and visual observations respectively. To initialize the weight variables, we use (II.36) in the following way. An auditory sample is given a high initial weight if it has many visual samples as neighbors, or $w_{a_i} = \sum_{\boldsymbol{v}_j \in \boldsymbol{V}} \exp(-d^2(\boldsymbol{a}_i, \boldsymbol{v}_j)/\sigma)$. Visual weights are initialized in an analogous way, $w_{v_i} = \sum_{\boldsymbol{a}_j \in \boldsymbol{A}} \exp(-d^2(\boldsymbol{v}_i, \boldsymbol{a}_j)/\sigma)$. As illustrated below, this *cross-modal* weighting scheme favors clusters composed of both auditory and visual observations. We recorded three sequences:

- The *fake speaker* (FS) sequence, *e.g.* first and second rows of Figure II.4, consists of two persons facing the camera and the microphones. While the person onto the right emits speech signals (counting from "one" to "ten") the person onto the left performs fake lip, facial, and head movements as he would speak.
- The *moving speakers* (MS) sequence, *e.g.* third and fourth rows of Figure II.4, consists of two persons that move around while they are always facing the cameras and microphones. The persons take speech turns but there is a short overlap between the two auditory signals.
- The *cocktail party* (CP) sequence, *e.g.* fifth and sixth rows of Figure II.4, consists of four persons engaged in an informal dialog. The persons wander around and turn their heads towards the active speaker; occasionally two persons speak simultaneously. Moreover the speakers do not always face the camera, hence face and lip detection/localization are unreliable.

The visual data are gathered with a single camera and the auditory data are gathered with two microphones plugged into the ears of an acoustic dummy head, referred to as *binaural audition*. The visual data are recorded at 25 video frames per second (FPS). The auditory data are gathered and processed in the following way. First, the short-time Fourier transform (STFT) is applied to the left- and right-microphone signals which are sampled at 48 KHz. Second, the left and right spectrograms thus obtained are combined to yield a binaural spectrogram from which a sound-source DOA is estimated. A spectrogram composed of 512 frequency bins is obtained by applying the STFT over a sliding window of width 0.064 s and shifted along the signal with 0.008 s hops. An audio frame, or 512 frequency bins, is associated with each window, hence there are 125 audio frames per second (with 0.056 ms overlap between consecutive frames). Both the visual and audio frames are further grouped into temporal segments of width 0.4 s, hence there are 10 visual frames and 50 audio frames in each segment.

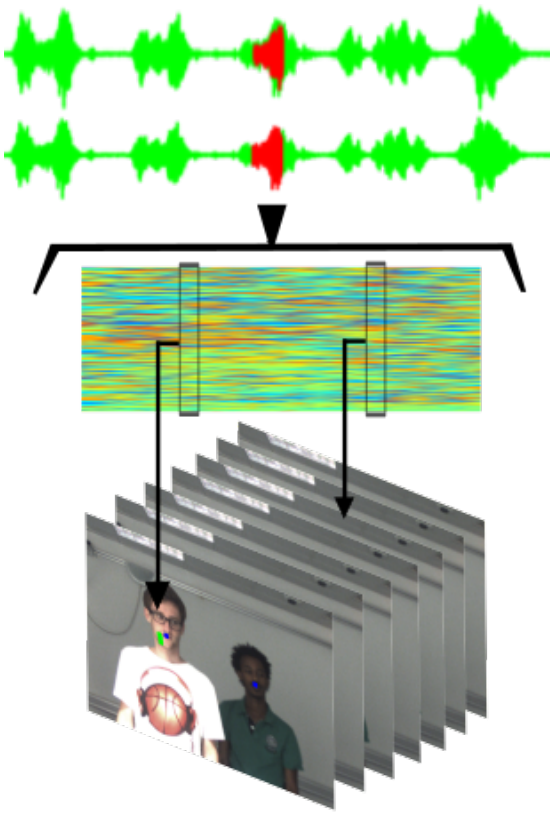


Figure II.3: *Audio-visual data acquisition and alignment. Top: left- and right-microphone signals. A temporal segment of 0.4 s is outlined in red. Middle: Binaural spectrogram that corresponds to the outlined segment. This spectrogram is composed of 50 binaural vectors, each one being associated with an audio frame (shown as a vertical rectangle). Bottom: video frames associated with a segment. A sound-source direction of arrival (DOA) is extracted from each binaural vector and mapped onto the image plane, hence each green dot in the image plane corresponds to a DOA.*

As already mentioned, we follow the method of [52] to extract a sound-source DOA from each audio frame. In order to increase the robustness of audio localization, a voice activity detector (VAD) [55] is first applied to each frame, such that not all the frames have DOA estimates associated with them. On an average there are 40 audio DOA observations per segment. The FS sequence contains 28 segments, the MS sequence contains 43 segments, while the CP sequence contains 115 segments. The left hand sides of Figure II.4 show the central frame of a segment with all the visual features (blue) and auditory features (green) available within that segment.

We tested the proposed WD-EM algorithm on these audio-visual data as well as the GMM+U [47] and FM-uMST [25] algorithms. We chose to compare our method with these two methods for the following reasons. Firstly, all three methods are based on finite mixtures and hence they can use a model selection criterion to estimate the number of components in the mixture that best approximates clusters in the data. This is important since the number of persons and of active speakers among these persons are not known in advance. Secondly, as demonstrated in the previous section, these three methods yield robust clustering in the presence of outliers.

WD-EM uses the MML criterion for model selection as described in Section II.6. We implemented a model selection criterion based on BIC to optimally select the number of components with GMM+U and FM-uMST. While each algorithm yields an optimal number of components for each audio-visual segment, not all them contain a sufficient number of audio and visual observations, such that the component can be associated with an active speaker. Therefore, we apply a simple two-step strategy, firstly to decide whether a component is *audio-visual*, *audio-only*, or *visual-only*, and secondly to select the best audio-visual components. Let n_v and n_a be the total number of visual and audio observations in a segment. We start by assigning each observation to a component: let n_a^k and n_v^k be the number of audio and visual observations associated with component k. Let $r_k = \min\{n_a^k, n_v^k\}/(n_a+n_v)$ measure the audio-visual relevance of a component. If $r_k \geq s$ then component k corresponds to an active speaker, with s being a fixed threshold.

Figure II.4 shows examples of applying the WD-EM, GMM+U and FM-uMST algorithms to the three sequences. One may notice that, while the visual observations (blue) are very accurate and form small *lumps* around the moving lips of a speaker (or of a fake speaker), audio observations (green) are very noisy and have different statistics; this is due to the presence of reverberations (the ceiling in particular) and of other sound sources, such as computer fans. The ground-truth active speaker is shown with a yellow frame. The data clusters obtained by the three methods

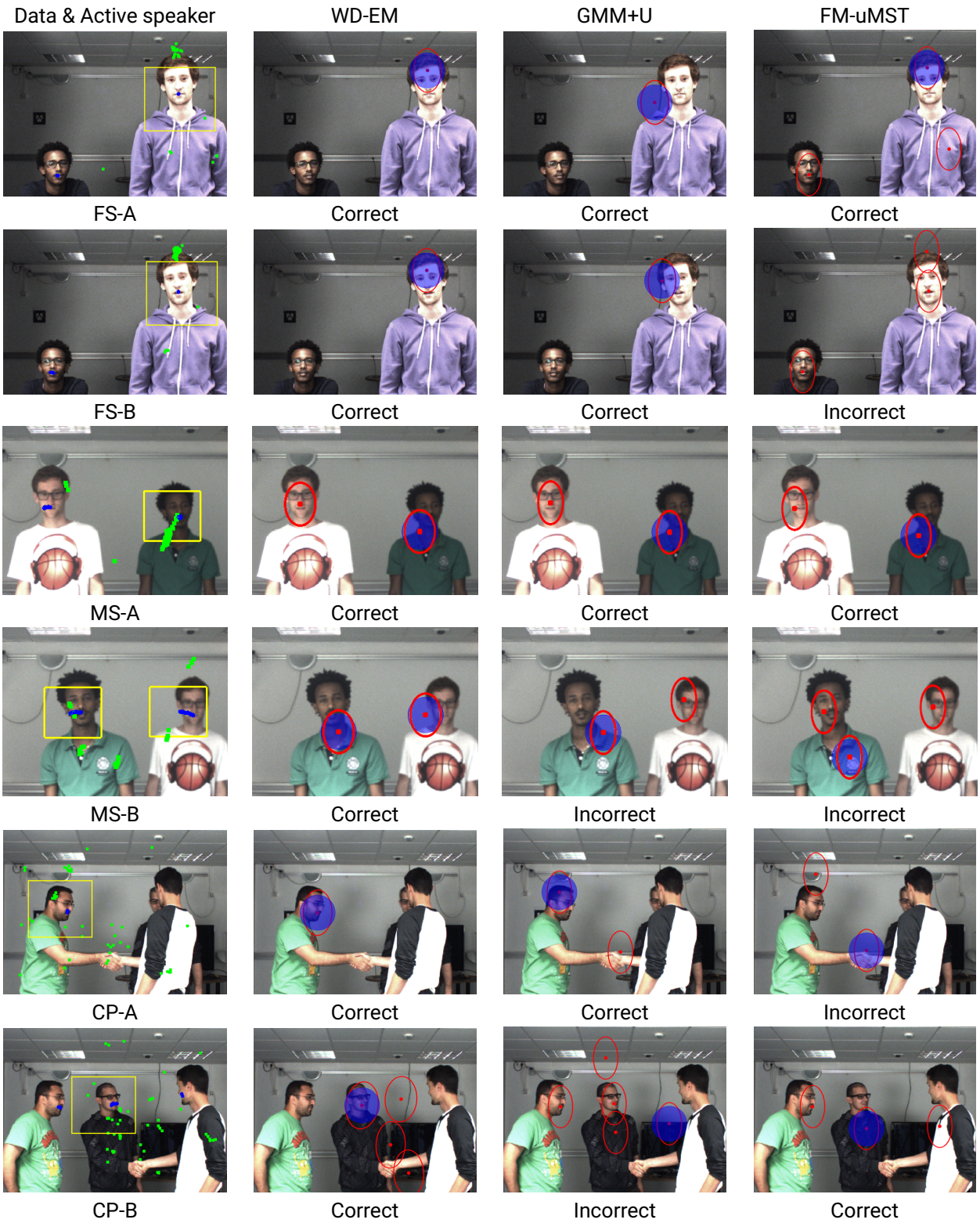


Figure II.4: *Results obtained on the fake speaker (FS), moving speaker (MS) and cocktail party (CP) sequences. The first column shows the audio (green) and visual (blue) observations, as well as a yellow bounding box that shows the ground-truth active speaker. The second, third and fourth columns show the mixture components obtained with the WD-EM, GMM+U and FM-uMST methods, respectively. The blue disks mark components that correspond to correct detections of active speakers, namely whenever there is an overlap between a component and the ground-truth bounding box.*

Table II.5: *The correct detection rates (CDR) obtained with the three methods for three scenarios: fake speaker (FS), moving speakers (MS), and cocktail party (CP).*

Scenario	# Segments	WD-EM	GMM-U [47]	FM-uMST [25]
FS	28	100.00%	100.00%	71.43%
MS	43	83.87%	61.90%	72.22%
CP	115	65.66%	52.48%	49.57%

are shown with red ellipses. A blue disk around a cluster center designates an audio-visual cluster. Altogether, one may notice that the proposed method outperforms the two other methods. An interesting feature of WD-EM is that the weights give more importance to the accurate visual data (because of the low-variance groups of observations available with these data) and hence the audio-visual cluster centers are pulled towards the visual data (lip locations in these examples).

To further quantify the performance of the three methods, we carefully annotated the data. For each segment, we identified the active speaker and we precisely located the speaker's lips. Let $\boldsymbol{x}_g$ be the ground-truth lip location. We assign $\boldsymbol{x}_g$ to a component by computing the maximum responsibility (II.15) of $\boldsymbol{x}_g$. When $\boldsymbol{x}_g$ is assigned to an audio-visual cluster, an active speaker is said to be correctly detected if the posterior probability of $\boldsymbol{x}_g$ is equal or greater than $1/K$, where K is the number of components. Table II.5 summarizes the results obtained with the three methods.

II.10 Conclusions

We presented a weighted-data Gaussian mixture model. We derived a maximum-likelihood formulation and we devised two EM algorithms, one that uses fixed weights (FWD-EM) and another one with weights modeled as random variables (WD-EM). While the first algorithm appears to be a straightforward generalization of standard EM for Gaussian mixtures, the second one has a more complex structure. We showed that the expectation and maximization steps of the proposed WD-EM admit closed-form solutions and hence the algorithm is extremely efficient. Moreover, WD-EM performs much better than FWD-EM which fully justifies the proposed generative probabilistic model for the weights. We extended the MML-based model selection criterion proposed in [32] to the weighted-data Gaussian mixture model and we proposed an algorithm that finds an optimal number of components in the data. Interestingly, the WD-EM algorithm compares favorably with several state-of-the-art parametric and non-parametric clustering methods: it performs particularly well in the presence of a large number of outliers, *e.g.* up to 50% of outliers. Hence, the proposed formulation belongs to the robust category of clustering methods.

We also applied WD-EM to the problem of clustering heterogenous/multimodal data sets, such as audio-visual data. We briefly described the audio-visual fusion problem and how it may be cast into a challenging audio-visual clustering problem, *e.g.* how to associate human faces with speech signals and how to detect and localize active speakers in complex audio-visual scenes. We showed that the proposed algorithm yields better audio-visual clustering results than two other finite-mixture models, and this for two reasons: (i) it is very robust to noise and to outliers and (ii) it allows a cross-modal weighting scheme. Although not implemented, the proposed model has many other interesting features when dealing with multimodal data: it enables to balance the importance of the modalities, to emphasize one modality, or to use any prior information that might be available, for example by giving high weight priors to visual data corresponding to face/lip localization.

Chapter *III*

Non-linear Regression for Acoustico-Articulatory Speaker Adaptation

Abstract *This chapter addresses the adaptation of an acoustic-articulatory inversion model of a reference speaker to the voice of another source speaker, using a limited amount of audio-only data. In this study, the articulatory-acoustic relationship of the reference speaker is modeled by a Gaussian mixture model and inference of articulatory data from acoustic data is made by the associated Gaussian mixture regression (GMR). To address speaker adaptation, we propose two different models based on GMR: the integrated-cascaded or IC-GMR and the joint or J-GMR. We present the two models, derive the respective EM algorithms for learning the parameters, and discuss the similarities and differences between the models and algorithms. We provide an extensive evaluation of the IC-GMR and J-GMR on both synthetic acoustic-articulatory data and on the multi-speaker MOCHA EMA database. We compare various GMR-based adaptation models, and discuss their respective merits.*

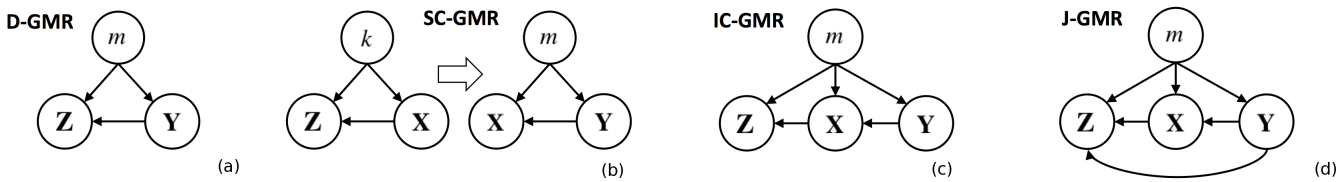


Figure III.1: *Graphical representation of the generative models associated to the D-, SC-, IC-, and J-GMR. In the present applicative framework, $\mathbf{Y}$ is a reference articulatory feature vector, $\mathbf{X}$ is a reference acoustic feature vector, and $\mathbf{Z}$ is a source acoustic feature vector.*

Chapter Pitch

Methodological contribution Two tri-variate mixture models working with missing data. The models differ in the way they consider a link between the first and third random subvectors of the joint random vector (see IC-GMR and J-GMR on Figure III.1), represented in orange in the following:

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{m=1}^{M} p(m)p(\boldsymbol{y}|m)p(\boldsymbol{x}|\boldsymbol{y}, m)p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}, m). \tag{III.1}$$

Applicative task Regression adaptation in general, acoustic-articulatory inversion in particular.

Interesting insight While the two models are strongly related to each other, the corresponding learning algorithms are not. Indeed, in order to keep low computational cost, the EM for IC-GMM must be constructed from the Gaussian-linear model, and not as a particular case of the EM for J-GMM.

Dissemination Both the IC-GMM and the J-GMM were published at IEEE Transactions on Audio, Speech and Language Processing, see [56], [57].

III.1 Introduction

Gaussian Mixture Regression (GMR) is an efficient regression technique derived from the well-known Gaussian Mixture Model (GMM) [18]. The GMR is widely used in different areas of speech processing, e.g. voice conversion [58], [59], in image processing, e.g. head pose estimation from depth data [60], generation of hand writing [61], and in robotics [17], [62], [63]. In the present chapter, we consider the application of GMR to the *speech acoustic-articulatory inversion problem*, i.e. estimating trajectories of speech articulators (jaws, lips, tongue, palate) from speech acoustic data [64]–[66]. Such model can be used in the context of pronunciation training to automatically animate a virtual talking head displaying the internal speech articulators, using only the speaker's voice. Such acoustic-articulatory GMR is generally trained on a large dataset of input-output joint observations recorded on a single speaker, later on referred to as the *reference* speaker. Using this reference GMR with the speech signal produced by a new speaker (hereafter referred to as the *source* speaker) can lead to poorly estimated articulatory trajectories. Indeed, because of the differences in the voice characteristics and in the speech production strategies across speakers, the new input data does not follow the statistical distribution of the reference acoustic data. Therefore, we address the problem of GMR *speaker adaptation*: We consider a GMR adaptation process that can be used to easily adapt a virtual talking head to any new speaker. Moreover, the adaptation process must be designed to work with a tiny set of *input-only*, i.e. acoustic, observations from the source speaker (in practice using a few sentences), in order to guarantee a user-friendly non-invasive system. Indeed, in real-world applications collecting data from a new user comes at high cost, especially for articulatory data.

The general speaker adaptation and normalization problem has been considered in, e.g., [67], [68]. In order to address this problem in the specific GMR framework, [69] proposed to adapt the model parameters related to input observations using two state-of-the-art adaptation techniques for GMM, namely: maximum a posteriori (MAP) [70] and maximum likelihood linear regression (MLLR) [71]. Then, we proposed a general framework called cascaded GMR (C-GMR) and derived two implementations [56]. The first one, referred to as Split-C-GMR (SC-GMR), is a simple chaining of two separate GMRs: a first GMR maps the source acoustic feature vector, denoted $\mathbf{Z}$, into a reference acoustic feature vector, denoted $\mathbf{X}$, and then a second GMR maps $\mathbf{X}$ into the output articulatory feature vector, denoted $\mathbf{Y}$, lying in the reference speaker articulatory space (see Fig. III.1-(b)). The second implementation, referred to as Integrated-C-GMR (IC-GMR) combines the two successive pings in a single probabilistic model (see Fig. III.1-(c)). Indeed, the $\mathbf{Z}$-to-$\mathbf{X}$ and $\mathbf{X}$-to-$\mathbf{Y}$ mappings are integrated at the mixture component level, sharing the $\mathbf{X}$ space. Importantly, the EM algorithm associated to the IC-GMR model [56] uses the general methodology of *missing data* [4], [72], explicitly taking into account the tiny amount of adaptation data from the source speaker. Specifically, the source data consisted in a small subset of the sentences of the reference training set, and the complement of this subset was considered missing. The IC-GMR was shown to provide superior performance to the SC-GMR and also to a direct GMR between $\mathbf{Z}$ and $\mathbf{Y}$ that disregards the $\mathbf{X}$ data (D-GMR, see Fig. III.1-(a)).

As seen in Fig. III.1-(c), the IC-GMR does not explicitly model any direct statistical dependency between $\mathbf{Z}$ and $\mathbf{Y}$ (i.e. in the graphical model, there is no arrow between $\mathbf{Z}$ and $\mathbf{Y}$). In other words, the cascade is "forced" to pass through $\mathbf{X}$, the reference speaker's acoustic space. In a general manner, adding such link would enable the output $\mathbf{Y}$ to be jointly inferred from $\mathbf{Z}$ and $\mathbf{X}$. In the above-mentioned limited parallel dataset strategy [56] (the source data consist in a small subset of the sentences of the reference training set) the acoustics of the source and the reference speaker are not physically linked, but they share the same phonetic content. Therefore, adding the $\mathbf{Z}$-$\mathbf{Y}$ link to the IC-GMR model enables to exploit the correlation associated to the shared phonetic content. Even if the direct $\mathbf{ZY}$ correlation happened to be weaker than the other cross-correlations ($\mathbf{ZX}$ and $\mathbf{XY}$), the impact of exploiting this direct link and thus estimating $\mathbf{Y}$ *jointly from* $\mathbf{X}$ *and* $\mathbf{Z}$ cannot be assessed with the IC-GMR. We also propose to use a joint multi-variate GMM on $\{\mathbf{Z}, \mathbf{X}, \mathbf{Y}\}$, and we can thus refer to this model as Joint GMM (J-GMM), and to the associated regressor as J-GMR.

The research question addressed: "Is there any benefit of explicitly modeling a direct link between the source speaker's acoustics ($\mathbf{Z}$) and the reference speaker's articulation ($\mathbf{Y}$), with special emphasis on the case of very limited amount of adaptation data?" To this aim:

- We present two GMM-based regression models: a "cascaded" GMR which disables the link between $\mathbf{Z}$ and $\mathbf{Y}$ and a "joint" (standard) GMR which enables this link. Both models are presented in the case of missing data, since the adaptation dataset is much smaller.
- We then provide the inference and learning procedures for both models, meaning the inference equation as well as the EM algorithm. These are then used to provide an comprehensive theoretical comparison between the "cascaded" IC-GMR and "joint" J-GMR models.
- Finally, wee provide an extensive evaluation of both GMR-based models on synthetic acoustic-articulatory data as well as on the multi-speaker MOCHA EMA database.

The remaining of this chapter is organized as follows. Section III.2 gives a brief technical presentation of the GMR speaker adaptation problem in the present acoustic-articulatory inversion contextm and presents two straightforward models: D-GMR and SC-GMR. Section III.3 presents the IC-GMR models, which integrates the link-less "cascacded" strategy within the mixture model, rather than having two different mixture models as in SC-GMR. The EM corresponding to the IC-GMR is presented in Section III.4. Section III.5 presents and discusses the J-GMM and the associated J-GMR inference equation. The complete derivation of the corresponding EM algorithm is presented in Section III.6. The theoretical differences between the IC-GMR and J-GMR models are discussed in Section III.7. In Section III.8, we evaluate the practical performance of the proposed J-GMR under the task of speech acoustic-to-articulatory inversion with two different datasets, one synthetic and one of real data, and compare it to the performance of the D-GMR, SC-GMR, and IC-GMR. We discuss the research question risen above in the light of the experimental results. Section III.9 concludes the chapter.

III.2 Gaussian Mixture Regression

III.2.1 Definitions, notations and working hypothesis

Let us consider a GMM and the associated GMR between realizations of input $\mathbf{X}$ and output $\mathbf{Y}$ random (column) vectors, of arbitrary finite dimension. In the present speech acoustic-to-articulatory inversion framework, $\mathbf{Y}$ is an articulatory feature vector and $\mathbf{X}$ is a corresponding acoustic feature vector, both from the reference speaker. Let us define $\boldsymbol{J} = [\mathbf{X}^\top, \mathbf{Y}^\top]^\top$ where $^\top$ denotes the transpose operator. Let $p(\mathbf{X} = \mathbf{x}; \boldsymbol{\Theta}_\mathbf{X})$ denote the probability density function (PDF) of $\mathbf{X}$.[III.1] Let $\mathcal{N}(\mathbf{x}; \mu_\mathbf{X}, \boldsymbol{\Sigma}_{\mathbf{XX}})$ denote the Gaussian distribution evaluated at $\mathbf{x}$ with mean vector $\mu_\mathbf{X}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{XX}}$. Let $\boldsymbol{\Sigma}_{\mathbf{XY}}$ denote the cross-covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$ and $\boldsymbol{\Lambda}_{\mathbf{XX}}$ the precision matrix of $\mathbf{X}$ (similarly for cross-terms). With these notations, the PDF of a GMM on $\boldsymbol{J}$ writes:

$$p(\boldsymbol{j}; \boldsymbol{\Theta}_{\boldsymbol{J}}) = \sum_{m=1}^{M} p(m)\mathcal{N}(\boldsymbol{j}; \mu_{\boldsymbol{J},m}, \boldsymbol{\Sigma}_{\mathbf{JJ},m}), \tag{III.2}$$

where M is the number of components, $p(m) = \pi_m \geq 0$ and $\sum_{m=1}^{M} \pi_m = 1$. Let $\mathcal{D}_{\mathbf{xy}} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$ denote a large training dataset of N i.i.d. vector pairs drawn from the $(\mathbf{X},\mathbf{Y})$ distribution. In practice, these data are feature vectors extracted from synchronized articulatory and acoustic recordings of the reference speaker. The parameters of the above GMM reference model are estimated from $\mathcal{D}_{\mathbf{xy}}$, using an EM algorithm. Then, inference of $\mathbf{y}$ given a new observed value $\mathbf{x}$ can be performed by the minimum mean squared error (MMSE) estimator, which is the posterior mean $\hat{\mathbf{y}} = \mathrm{E}[\mathbf{Y}|\mathbf{x}]$:

$$\hat{\mathbf{y}} = \sum_{m=1}^{M} p(m|\mathbf{x}) \left(\mu_{\mathbf{Y},m} + \boldsymbol{\Sigma}_{\mathbf{YX},m}\boldsymbol{\Sigma}_{\mathbf{XX},m}^{-1}(\mathbf{x} - \mu_{\mathbf{X},m})\right), \tag{III.3}$$

with $p(m|\mathbf{x}) = \frac{\pi_m \mathcal{N}(\mathbf{x}; \mu_{\mathbf{X},m}, \boldsymbol{\Sigma}_{\mathbf{XX},m})}{\sum_{k=1}^{M} \pi_k \mathcal{N}(\mathbf{x}; \mu_{\mathbf{X},k}, \boldsymbol{\Sigma}_{\mathbf{XX},k})}$. Alternatively, one may consider maximum a posteriori (MAP) inference using $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$.

Let us now consider a new input vector $\mathbf{Z}$ following a different statistical distribution that the one of $\mathbf{X}$. Here, $\mathbf{Z}$ is an acoustic feature vector from the source speaker, to which the reference GMR has to be adapted. We assume that a tiny dataset $\mathcal{D}_\mathbf{z}$ of new input vectors $\mathbf{z}$ is available for the adaptation. As in [56], we assume that $\mathcal{D}_\mathbf{z}$ can be aligned with a subset of the reference input dataset: This requires that the new speaker pronounces a subset of the sentences contained in $\mathcal{D}_{\mathbf{xy}}$ and that these new recordings are time-aligned with the corresponding recordings of the reference speaker (*e.g.* using dynamic time warping (DTW) techniques). Since the working hypothesis is that the data tuples are i.i.d., we can reorder the dataset and write without loss of generality $\mathcal{D}_\mathbf{z} = \{\mathbf{z}_n\}_{n=1}^{N_0}$, with $N_0 \ll N$.

III.2.2 D-GMR and SC-GMR

In this section, we briefly recall the two basic approaches for GMR adaptation, namely D-GMR and SC-GMR in Fig. III.1.

The first one is a direct $\mathbf{Z}$-to-$\mathbf{Y}$ GMR (D-GMR). Inference of $\mathbf{y}$ given an observed value $\mathbf{z}$ is done using (III.3), replacing $\mathbf{x}$ and $\mathbf{X}$ with $\mathbf{z}$ and $\mathbf{Z}$:

$$\hat{\mathbf{y}} = \sum_{m=1}^{M} p(m|\mathbf{z}) \left(\mu_{\mathbf{Y},m} + \boldsymbol{\Sigma}_{\mathbf{YZ},m}\boldsymbol{\Sigma}_{\mathbf{ZZ},m}^{-1}(\mathbf{z} - \mu_{\mathbf{Z},m})\right), \tag{III.4}$$

with $p(m|\mathbf{z}) = \frac{\pi_m \mathcal{N}(\mathbf{z}; \mu_{\mathbf{Z},m}, \boldsymbol{\Sigma}_{\mathbf{ZZ},m})}{\sum_{k=1}^{M} \pi_k \mathcal{N}(\mathbf{z}; \mu_{\mathbf{Z},k}, \boldsymbol{\Sigma}_{\mathbf{ZZ},k})}$. The parameters are trained with $\mathcal{D}_{\mathbf{zy}} = \{\mathbf{z}_n, \mathbf{y}_n\}_{n=1}^{N_0}$.

[III.1] In the following, for concision we omit $\mathbf{X}$ and we may omit $\boldsymbol{\Theta}_\mathbf{X}$, depending on the context.

The second model is an instance of the cascaded GMR. As mentioned in the introduction, the Split-Cascaded GMR (SC-GMR) consists of chaining two distinct GMRs: a $\mathbf{Z}$-to-$\mathbf{X}$ GMR followed by the reference $\mathbf{X}$-to-$\mathbf{Y}$ GMR. The inference equation thus consists in chaining $\hat{\mathbf{x}} = \mathrm{E}[\mathbf{X}|\mathbf{z}]$ and $\hat{\mathbf{y}} = \mathrm{E}[\mathbf{Y}|\hat{\mathbf{x}}]$, where both expectations follow (III.3) with their respective parameters. Note that the two GMRs may have a different number of mixture components. Note also that the first GMR is trained with the N_0 samples of $\mathcal{D}_{\mathbf{zx}} = \{\mathbf{z}_n, \mathbf{x}_n\}_{n=1}^{N_0}$, while the second GMR is the reference GMR trained with the N samples of $\mathcal{D}_{\mathbf{xy}}$.

III.3 Integrated Cascaded GMR

We now present the integrated cascaded GMR (IC-GMR) model, see Figure III.1 that we propose to address the present speaker adaptation problem. Then, we discuss the specific way the EM algorithm is to be used in this context. The technical derivation of this algorithm is given in the next section.

III.3.1 Definition of the mixture model

The core idea of the IC-GMR model is to combine spectral conversion and acoustic-articulatory inversion into a single GMR-based mapping process. Very importantly, this is made at the component level of the GMR, i.e. *within the mixture*, as opposed to the SC-GMR of Section III.2. In other words, the plugged "conversion + inversion" components share the same component assignment variable m, as illustrated by the graphical model shown in Fig. III.1 (c). The goal is to benefit from the partitioning of the acoustic-articulatory space of the reference speaker (i.e. $\boldsymbol{X}$-$\boldsymbol{Y}$) which is assumed to be well estimated, when proceeding to the source speaker adaptation. Contrary to the SC-GMR, the structure of the $\boldsymbol{Z}$-to-$\boldsymbol{X}$ conversion process is thus here constrained by the structure of the $\boldsymbol{X}$-to-$\boldsymbol{Y}$ GMR.

The statistical dependencies between $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are here defined as:

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\Theta}) = \sum_{m=1}^{M} p(m)p(\boldsymbol{y}|m, \boldsymbol{\Theta}_{\boldsymbol{Y},m})p(\boldsymbol{x}|\boldsymbol{y}, m, \boldsymbol{\Theta}_{\boldsymbol{X}|\boldsymbol{Y},m})p(\boldsymbol{z}|\boldsymbol{x}, m, \boldsymbol{\Theta}_{\boldsymbol{Z}|\boldsymbol{X},m}), \tag{III.5}$$

with

$$p(m) = \pi_m, \tag{III.6}$$
$$p(\boldsymbol{y}|m, \boldsymbol{\Theta}_{\boldsymbol{Y},m}) = \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{e}_m, \mathbf{R}_m\right), \tag{III.7}$$
$$p(\boldsymbol{x}|\boldsymbol{y}, m, \boldsymbol{\Theta}_{\boldsymbol{X}|\boldsymbol{Y},m}) = \mathcal{N}\left(\boldsymbol{x}|\mathbf{A}_m\boldsymbol{y} + \boldsymbol{b}_m, \mathbf{U}_m\right), \tag{III.8}$$
$$p(\boldsymbol{z}|\boldsymbol{x}, m, \boldsymbol{\Theta}_{\boldsymbol{Z}|\boldsymbol{X},m}) = \mathcal{N}\left(\boldsymbol{z}|\mathbf{C}_m\boldsymbol{x} + \boldsymbol{d}_m, \mathbf{V}_m\right). \tag{III.9}$$

For each component, π_m still represents the prior distribution, $\boldsymbol{e}_m$ and $\mathbf{R}_m$ are respectively the mean vector and covariance matrix of the marginal Gaussian distribution of $\boldsymbol{Y}$, $\mathbf{A}_m$, $\boldsymbol{b}_m$ and $\mathbf{U}_m$ are respectively the transition matrix, constant vector and covariance matrix of the linear-Gaussian conditional pdf model in $(\boldsymbol{X}, \boldsymbol{Y})$, and the same for $\mathbf{C}_m$, $\boldsymbol{d}_m$ and $\mathbf{V}_m$ with $(\boldsymbol{Z}, \boldsymbol{X})$.

III.3.2 Inference equation

Similarly to Section III.2, the minimum MSE estimation $\hat{\boldsymbol{y}}$ of $\boldsymbol{y}$ given $\boldsymbol{z}$ is given by its posterior mean:[III.2]

$$\hat{\boldsymbol{y}} = \mathrm{E}[\boldsymbol{Y}|\boldsymbol{z}] = \int_{\mathbb{R}^{D_Y}} \boldsymbol{y}p(\boldsymbol{y}|\boldsymbol{z})\mathrm{d}\boldsymbol{y}, \tag{III.10}$$

with

$$p(\boldsymbol{y}|\boldsymbol{z}) = \int_{\mathbb{R}^{D_X}} \sum_{m=1}^{M} p(\boldsymbol{x}, \boldsymbol{y}, m|\boldsymbol{z})\mathrm{d}\boldsymbol{x}. \tag{III.11}$$

In the IC-GMR case we have:

$$p(\boldsymbol{x}, \boldsymbol{y}, m|\boldsymbol{z}) = p(m|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, m)p(\boldsymbol{x}|\boldsymbol{z}, m) = p(m|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{x}, m)p(\boldsymbol{x}|\boldsymbol{z}, m), \tag{III.12}$$

[III.2] In this subsection we omit the parameter set in PDF notation for clarity of presentation.

since $\boldsymbol{Y}$ is independent of $\boldsymbol{Z}$ conditionally on $\boldsymbol{X}$ and m [4]–(Section 8.2). Therefore, we have:

$$p(\boldsymbol{y}|\boldsymbol{z}) = \sum_{m=1}^{M} p(m|\boldsymbol{z}) \int_{\mathbb{R}^{D_X}} p(\boldsymbol{y}|\boldsymbol{x}, m) p(\boldsymbol{x}|\boldsymbol{z}, m) \mathrm{d}\boldsymbol{x}. \tag{III.13}$$

At this point, we can insert (III.13) into (III.10). But to go further, we face a problem: the model is expressed in terms of the distributions $p(\boldsymbol{y}|m)$, $p(\boldsymbol{x}|\boldsymbol{y}, m)$, $p(\boldsymbol{z}|\boldsymbol{x}, m)$ and not the "inverse" distributions $p(\boldsymbol{z}|m)$, $p(\boldsymbol{x}|\boldsymbol{z}, m)$, $p(\boldsymbol{y}|\boldsymbol{x}, m)$ as required in (III.13).[III.3] Fortunately, a linear-Gaussian model is "invertible": knowing the Gaussian PDFs $p(\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y})$, the PDFs $p(\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$ are derived easily and form a linear-Gaussian model [4] (p. 93). In the present case, we can chain the inversion across $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{Z}$ to obtain:

$$p(\boldsymbol{y}|\boldsymbol{x}, m, \boldsymbol{\Theta}_{\boldsymbol{Y}|\boldsymbol{X},m}) = \mathcal{N}\left(\boldsymbol{y}|\mathbf{A}^*_m \boldsymbol{x} + \boldsymbol{b}^*_m, \mathbf{U}^*_m\right), \tag{III.14}$$
$$p(\boldsymbol{x}|m, \boldsymbol{\Theta}_{\boldsymbol{X},m}) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{e}^*_m, \mathbf{R}^*_m\right), \tag{III.15}$$
$$p(\boldsymbol{x}|\boldsymbol{z}, m, \boldsymbol{\Theta}_{\boldsymbol{X}|\boldsymbol{Z},m}) = \mathcal{N}\left(\boldsymbol{x}|\mathbf{C}^*_m \boldsymbol{z} + \boldsymbol{d}^*_m, \mathbf{V}^*_m\right), \tag{III.16}$$
$$p(\boldsymbol{z}|m, \boldsymbol{\Theta}_{\boldsymbol{Z},m}) = \mathcal{N}\left(\boldsymbol{z}|\boldsymbol{f}^*_m, \mathbf{P}^*_m\right), \tag{III.17}$$

with

$$\begin{aligned}
\mathbf{U}^*_m &= (\mathbf{R}_m^{-1} + \mathbf{A}_m^\top \mathbf{U}_m^{-1} \mathbf{A}_m)^{-1}, \\
\mathbf{A}^*_m &= \mathbf{U}^*_m \mathbf{A}_m^\top \mathbf{U}_m^{-1}, \quad \boldsymbol{b}^*_m = \mathbf{U}^*_m (\mathbf{R}_m^{-1} \boldsymbol{e}_m - \mathbf{A}_m^\top \mathbf{U}_m^{-1} \boldsymbol{b}_m), \\
\mathbf{R}^*_m &= \mathbf{U}_m + \mathbf{A}_m \mathbf{R}_m \mathbf{A}_m^\top, \quad \boldsymbol{e}^*_m = \mathbf{A}_m \boldsymbol{e}_m + \boldsymbol{b}_m, \\
\mathbf{V}^*_m &= (\mathbf{R}^{*-1}_m + \mathbf{C}_m^\top \mathbf{V}_m^{-1} \mathbf{C}_m)^{-1}, \\
\mathbf{C}^*_m &= \mathbf{V}^*_m \mathbf{C}_m^\top \mathbf{V}_m^{-1}, \quad \boldsymbol{d}^*_m = \mathbf{V}^*_m (\mathbf{R}^{*-1}_m \boldsymbol{e}^*_m - \mathbf{C}_m^\top \mathbf{V}_m^{-1} \boldsymbol{d}_m), \\
\mathbf{P}^*_m &= \mathbf{V}_m + \mathbf{C}_m \mathbf{R}^*_m \mathbf{C}_m^\top, \quad \boldsymbol{f}^*_m = \mathbf{C}_m \boldsymbol{e}^*_m + \boldsymbol{d}_m.
\end{aligned}$$

Now we can calculate (III.10) as:

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \sum_{m=1}^{M} p(m|\boldsymbol{z}) \int_{\mathbb{R}^{D_X}} \left(\int_{\mathbb{R}^{D_Y}} \boldsymbol{y} p(\boldsymbol{y}|\boldsymbol{x}, m) \mathrm{d}\boldsymbol{y} \right) p(\boldsymbol{x}|\boldsymbol{z}, m) \mathrm{d}\boldsymbol{x} = \sum_{m=1}^{M} p(m|\boldsymbol{z}) \int_{\mathbb{R}^{D_X}} (\mathbf{A}^*_m \boldsymbol{x} + \boldsymbol{b}^*_m) p(\boldsymbol{x}|\boldsymbol{z}, m) \mathrm{d}\boldsymbol{x} \\
&= \sum_{m=1}^{M} p(m|\boldsymbol{z}) (\mathbf{A}^*_m (\mathbf{C}^*_m \boldsymbol{z} + \boldsymbol{d}^*_m) + \boldsymbol{b}^*_m),
\end{aligned} \tag{III.18}$$

and finally:

$$\hat{\boldsymbol{y}} = \sum_{m=1}^{M} p(m|\boldsymbol{z}) (\mathbf{A}^*_m \mathbf{C}^*_m \boldsymbol{z} + \mathbf{A}^*_m \boldsymbol{d}^*_m + \boldsymbol{b}^*_m). \tag{III.19}$$

It can be shown that $\mathbf{C}^*_m = \boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{Z},m} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{Z}\boldsymbol{Z},m}$, $\boldsymbol{d}^*_m = \boldsymbol{\mu}_{\boldsymbol{X},m} - \mathbf{C}^*_m \boldsymbol{\mu}_{\boldsymbol{Z},m}$. Therefore, (III.19) is equivalent to :

$$\hat{\boldsymbol{y}} = \sum_{m=1}^{M} p(m|\boldsymbol{z}) (\boldsymbol{\mu}_{\boldsymbol{Y},m} + \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{X},m} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{X}\boldsymbol{X},m} \boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{Z},m} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{Z}\boldsymbol{Z},m} (\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{Z},m})). \tag{III.20}$$

The component weights $p(m|\boldsymbol{z})$ are obtained by applying the classical formula (III.3) with distribution (III.17).

Equation (III.20) exhibits the chaining of $\boldsymbol{Z}$-to-$\boldsymbol{X}$ and $\boldsymbol{X}$-to-$\boldsymbol{Y}$ linear regressions at the mixture component level. This results into a $\boldsymbol{Z}$-to-$\boldsymbol{Y}$ GMR with a specific form of the covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Z},m} = \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{X},m} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{X}\boldsymbol{X},m} \boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{Z},m}. \tag{III.21}$$

Note that these parameters depend on the joint distribution of $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$, and in practice they are estimated from all available $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ data (as we will see below). Even if their inference equation has the same general form, this makes the IC-GMR quite different from the D-GMR, since this latter was obtained from a limited set of N_0 $(\boldsymbol{z}, \boldsymbol{y})$ data only.

III.4 EM algorithm for IC-GMR

In the following, we derive the exact EM algorithm associated to the IC-GMR model presented in the previous section. The aim of the EM algorithm is to maximize the *expected complete-data log-likelihood*, denoted by Q. At each iteration, the E-step computes Q and the M-step maximizes Q with respect to the parameters $\boldsymbol{\Theta}$. The EM algorithm alternates between the E and M steps until convergence.

[III.3] $p(m|\boldsymbol{z})$ can be deduced from $p(\boldsymbol{z}|m)$ using the Bayes formula.

III.4.1 E-step

At iteration $i+1$, $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ is defined as the expected value of the complete data log-likelihood with parameter set $\boldsymbol{\Theta}$. The expectation is taken accordingly to the posterior distribution of latent variables given the observed data and the parameter set at the previous iteration, $\boldsymbol{\Theta}^{(i)}$. In order to derive the Q function we follow the general methodology given in, e.g., [4]–(Section 9.4) and [73]. This leads to:

$$Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}\right) = \sum_{n=1}^{N_0}\sum_{m=1}^{M}\gamma_m^{(i+1)}(\boldsymbol{o}_n)\log p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m) + \sum_{n=N_0+1}^{N}\sum_{m=1}^{M}\frac{1}{p\left(\boldsymbol{j}_n|\boldsymbol{\Theta}_{\mathbf{J}}^{(i)}\right)}\int_{\mathbb{R}^{D_Z}} p\left(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m^{(i)}\right)\log p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m)\mathrm{d}\boldsymbol{z}_n \tag{III.22}$$

where all pdfs are defined in Section III.3.1, $\boldsymbol{j}_n = [\boldsymbol{x}_n^\top \boldsymbol{y}_n^\top]^\top$ and $\boldsymbol{o}_n = [\boldsymbol{x}_n^\top \boldsymbol{y}_n^\top \boldsymbol{z}_n^\top]^\top$ (see the details in Appendix III.10). For $n \in [1, N_0]$,

$$\gamma_m^{(i+1)}(\boldsymbol{o}_n) = \frac{p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m^{(i)})}{p(\boldsymbol{o}_n|\boldsymbol{\Theta}^{(i)})} \tag{III.23}$$

are the so-called *responsibilities* (of component m explaining observation $\boldsymbol{o}_n$) [4]. Note that (III.22) is valid for any trivariate mixture model on $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ (or any bivariate mixture model on $(\mathbf{J}, \boldsymbol{Z})$) with partially missing $\boldsymbol{z}$ data and i.i.d. observations. If we now extend the definition of responsibilities for $n \in [N_0+1, N]$ with:

$$\gamma_m^{(i+1)}(\boldsymbol{j}_n) = \frac{p(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_{\mathbf{J},m}^{(i)})}{p(\boldsymbol{j}_n|\boldsymbol{\Theta}_{\mathbf{J}}^{(i)})}, \tag{III.24}$$

and we use the IC-GMR definition (III.5)–(III.9), (III.22) becomes (see Appendix III.10 for details):

$$\begin{aligned} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}) = \frac{1}{2}\sum_{n=1}^{N_0}\sum_{m=1}^{M}\gamma_m^{(i+1)}(\boldsymbol{o}_n)\,(2\log\pi_m - \log|\mathbf{R}_m| - \log|\mathbf{U}_m| - \log|\mathbf{V}_m| - \mathcal{M}(\boldsymbol{y}_n - \boldsymbol{e}_m; \mathbf{R}_m) - \mathrm{Tr}[\mathbf{V}_m^{-1}\mathbf{V}_m^{(i)}]) \\ -\mathcal{M}(\boldsymbol{x}_n - \mathbf{A}_m\boldsymbol{y}_n - \boldsymbol{b}_m; \mathbf{U}_m) - \mathcal{M}(\boldsymbol{z}_n - \mathbf{C}_m\boldsymbol{x}_n - \boldsymbol{d}_m; \mathbf{V}_m)) + \frac{1}{2}\sum_{n=N_0+1}^{N}\sum_{m=1}^{M}\gamma_m^{(i+1)}(\boldsymbol{j}_n)\,(2\log\pi_m - \log|\mathbf{R}_m| - \log|\mathbf{U}_m| \\ - \log|\mathbf{V}_m| - \mathcal{M}(\boldsymbol{y}_n - \boldsymbol{e}_m; \mathbf{R}_m) - \mathcal{M}(\boldsymbol{x}_n - \mathbf{A}_m\boldsymbol{y}_n - \boldsymbol{b}_m; \mathbf{U}_m) - \mathcal{M}(\mathbf{C}_m^{(i)}\boldsymbol{x}_n + \boldsymbol{d}_m^{(i)} - \mathbf{C}_m\boldsymbol{x}_n - \boldsymbol{d}_m; \mathbf{V}_m), \end{aligned} \tag{III.25}$$

where $\mathcal{M}(\boldsymbol{x}; \mathbf{R}) = \boldsymbol{x}^\top\mathbf{R}^{-1}\boldsymbol{x}$ denotes the Mahalanobis distance of $\boldsymbol{x}$ with matrix $\mathbf{R}$ and Tr stands for the trace operator. The sum in the range $[1, N_0]$ is a direct match of [4]–(9.40), i.e. the classical EM for GMM, while the sum in $[N_0+1, N]$ results from the expectation over the missing data $\boldsymbol{z}_n$.

For $n \in [N_0+1, N]$, let us denote the expected value of $\boldsymbol{Z}_n$ given $\boldsymbol{x}_n$ for the m-th model component by $\boldsymbol{z}'_{nm} = \mathbf{C}_m^{(i)}\boldsymbol{x}_n + \boldsymbol{d}_m^{(i)} = \mu_{\boldsymbol{Z}|\boldsymbol{x}_n,m}^{(i+1)}$. This amounts to replace the missing data with their conditional mean given $\boldsymbol{x}_n$ and the current model parameters. For convenience, let us extend the notation $\boldsymbol{z}'_{nm}$ to the interval $n \in [1, N_0]$ with $\boldsymbol{z}'_{nm} = \boldsymbol{z}_n$ (which does not depend on m here). If, in addition, we denote $\gamma_{nm}^{(i+1)} = \gamma_m^{(i+1)}(\boldsymbol{o}_n)$ for $n \in [1, N_0]$ and $\gamma_{nm}^{(i+1)} = \gamma_m^{(i+1)}(\boldsymbol{j}_n)$ for $n \in [N_0+1, N]$, then $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ can be rewritten as:

$$\begin{aligned} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_{nm}^{(i+1)}(2\log\pi_m - \log|\mathbf{R}_m| - \mathcal{M}(\boldsymbol{y}_n - \boldsymbol{e}_m; \mathbf{R}_m) - \log|\mathbf{U}_m| - \mathcal{M}(\boldsymbol{x}_n - \mathbf{A}_m\boldsymbol{y}_n - \boldsymbol{b}_m; \mathbf{U}_m) \\ - \log|\mathbf{V}_m| - \mathcal{M}(\boldsymbol{z}'_{nm} - \mathbf{C}_m\boldsymbol{x}_n - \boldsymbol{d}_m; \mathbf{V}_m)) - \sum_{m=1}^{M}\left(\sum_{n=N_0+1}^{N}\gamma_{nm}^{(i+1)}\right)\mathrm{Tr}[\mathbf{V}_m^{-1}\mathbf{V}_m^{(i)}]. \end{aligned} \tag{III.26}$$

III.4.2 M-step

In this subsection, we provide the M-step updates for the IC-GMR parameters. The details of the derivations are given in Appendix III.11. Three important properties of the update rules appear. First, they are all closed-form expressions, thus yielding to an intrisically efficient EM algorithm. Second, the dependencies between the update rules do not form a loop. In other words, we first update the parameters that are independent, to later on estimate the rest of them. Third, several auxiliary quantities are shared between different updates, so that calculating these quantities once for all saves computational power. Additionally, this allows to present the update rules more clearly, as follows.

Auxiliary variables are weighted sums of the observations and their outer-products:

$$S_m^{(i+1)} = \sum_{n=1}^{N} \gamma_{nm}^{(i+1)}, \quad S_{\boldsymbol{X},m}^{(i+1)} = \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \boldsymbol{x}_n, \text{ and } \quad S_{\boldsymbol{XX},m}^{(i+1)} = \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \boldsymbol{x}_n \boldsymbol{x}_n^\top. \tag{III.27}$$

The definition of the variables $S_{\boldsymbol{Y},m}^{(i+1)}$, $S_{\boldsymbol{Z'},m}^{(i+1)}$, $S_{\boldsymbol{XY},m}^{(i+1)}$, $S_{\boldsymbol{YY},m}^{(i+1)}$, $S_{\boldsymbol{Z'X},m}^{(i+1)}$ and $S_{\boldsymbol{Z'Z'},m}^{(i+1)}$ follows the same principle.

Priors Maximization of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to the priors is trivial, since it is identical to the GMM case [4] (with of course the responsibilities being calculated from the IC-GMR's PDF). For $m \in [1, M]$, we have:

$$\pi_m^{(i+1)} = \frac{1}{N} S_m^{(i+1)}. \tag{III.28}$$

Constant vectors and transition matrices For $m \in [1, M]$, we have:

$$\boldsymbol{e}_m^{(i+1)} = \frac{1}{S_m^{(i+1)}} S_{\boldsymbol{Y},m}^{(i+1)}, \tag{III.29}$$

and $\mathbf{A}_m$, $\boldsymbol{b}_m$, $\mathbf{C}_m$ and $\boldsymbol{d}_m$ are updated with:

$$\mathbf{A}_m^{(i+1)} = \left(S_{\boldsymbol{XY},m}^{(i+1)} - \frac{1}{S_m^{(i+1)}} S_{\boldsymbol{X},m}^{(i+1)} S_{\boldsymbol{Y},m}^{(i+1)\top}\right)\left(S_{\boldsymbol{YY},m}^{(i+1)} - \frac{1}{S_m^{(i+1)}} S_{\boldsymbol{Y},m}^{(i+1)} S_{\boldsymbol{Y},m}^{(i+1)\top}\right)^{-1}, \boldsymbol{b}_m^{(i+1)} = \frac{1}{S_m^{(i+1)}}\left(S_{\boldsymbol{X},m}^{(i+1)} - \mathbf{A}_m^{(i+1)} S_{\boldsymbol{Y},m}^{(i+1)}\right) \tag{III.30}$$

$$\mathbf{C}_m^{(i+1)} = \left(S_{\boldsymbol{Z'X},m}^{(i+1)} - \frac{1}{S_m^{(i+1)}} S_{\boldsymbol{Z'},m}^{(i+1)} S_{\boldsymbol{X},m}^{(i+1)\top}\right)\left(S_{\boldsymbol{XX},m}^{(i+1)} - \frac{1}{S_m^{(i+1)}} S_{\boldsymbol{X},m}^{(i+1)} S_{\boldsymbol{X},m}^{(i+1)\top}\right)^{-1}, \boldsymbol{d}_m^{(i+1)} = \frac{1}{S_m^{(i+1)}}\left(S_{\boldsymbol{Z'},m}^{(i+1)} - \mathbf{C}_m^{(i+1)} S_{\boldsymbol{X},m}^{(i+1)}\right). \tag{III.31}$$

Note that $\mathbf{A}_m^{(i+1)}$ and $\boldsymbol{b}_m^{(i+1)}$ have the form of the standard weighted-MSE estimates of $\mathbf{A}_m$ and $\boldsymbol{b}_m$ given the $(\boldsymbol{x}, \boldsymbol{y})$ dataset and using the responsibilities as weights. $\mathbf{C}_m^{(i+1)}$ and $\boldsymbol{d}_m^{(i+1)}$ have a similar form but take into account partially missing $\boldsymbol{z}$ data.

Covariance matrices For $m \in [1, M]$, we have:

$$\mathbf{R}_m^{(i+1)} = \boldsymbol{e}_m^{(i+1)} \boldsymbol{e}_m^{(i+1)\top} + \frac{\left(S_{\boldsymbol{YY},m}^{(i+1)} - S_{\boldsymbol{Y},m}^{(i+1)} * \boldsymbol{e}_m^{(i+1)}\right)}{S_m^{(i+1)}}, \tag{III.32}$$

$$\mathbf{U}_m^{(i+1)} = \boldsymbol{b}_m^{(i+1)} \boldsymbol{b}_m^{(i+1)\top} + \frac{1}{S_m^{(i+1)}}\left(S_{\boldsymbol{XX},m}^{(i+1)} + \mathbf{A}_m^{(i+1)} S_{\boldsymbol{YY},m}^{(i+1)} \mathbf{A}_m^{(i+1)\top} - S_{\boldsymbol{XY},m}^{(i+1)} * \mathbf{A}_m^{(i+1)} - \left(S_{\boldsymbol{X},m}^{(i+1)} - \mathbf{A}_m^{(i+1)} S_{\boldsymbol{Y},m}^{(i+1)}\right) * \boldsymbol{b}_m^{(i+1)}\right), \tag{III.33}$$

$$\begin{aligned}\mathbf{V}_m^{(i+1)} = {} & \boldsymbol{d}_m^{(i+1)} \boldsymbol{d}_m^{(i+1)\top} + \frac{1}{S_m^{(i+1)}}\left(S_{\boldsymbol{Z'Z'},m}^{(i+1)} + \mathbf{C}_m^{(i+1)} S_{\boldsymbol{XX},m}^{(i+1)} \mathbf{C}_m^{(i+1)\top}\right. \\ & \left. - S_{\boldsymbol{Z'X},m}^{(i+1)} * \mathbf{C}_m^{(i+1)} - \left(S_{\boldsymbol{Z'},m}^{(i+1)} - \mathbf{C}_m^{(i+1)} S_{\boldsymbol{X},m}^{(i+1)}\right) * \boldsymbol{d}_m^{(i+1)}\right) + \mathbf{V}_m^{(i)} \textstyle\sum_{n=N_0+1}^{N} \gamma_{nm}^{(i+1)},\end{aligned} \tag{III.34}$$

where $\mathbf{P} * \mathbf{Q} = \mathbf{P}\mathbf{Q}^\top + \mathbf{Q}\mathbf{P}^\top$ denotes the symmetrized outer product of $\mathbf{P}$ and $\mathbf{Q}$. Interestingly, (III.29), (III.32), (III.30) and (III.33) correspond to the classical two-variable GMM, whereas (III.31) and (III.34) encode the effect of the missing data. Indeed, all statistics related to $\boldsymbol{Z}$ are computed using the actually observed $\boldsymbol{z}_n$ for $n \in [1, N_0]$ and the expected value $\mu_{\boldsymbol{Z}|\boldsymbol{x}_n,m}^{(i+1)}$ for $n \in [N_0 + 1, N]$.

III.4.3 EM Initialization

In order to infer the articulatory trajectory $\boldsymbol{y}$ from the acoustic features of the source speaker $\boldsymbol{z}$ by means of (III.19), the parameters of the joint model (III.5) need to be estimated from the data. Since (III.5) is a mixture model, this naturally leads to an EM algorithm [4], [6], whose derivation is given in the next section. In general, the initialization of EM algorithms is known to be a crucial phase. In the present study, we propose the following strategy:

- First, the reference GMR is obtained from an extensive set of $(\boldsymbol{x}, \boldsymbol{y})$ data, using the EM algorithm for GMMs.
- Second, we note that the joint marginal distribution of $(\boldsymbol{X}, \boldsymbol{Y})$ obtained by integrating (III.5) over $\boldsymbol{z}$ is given by:

$$p(\boldsymbol{j}|\boldsymbol{\Theta}_{\mathbf{J}}) = \sum_{m=1}^{M} \pi_m p(\boldsymbol{y}|m, \boldsymbol{\Theta}_{\boldsymbol{Y},m}) p(\boldsymbol{x}|\boldsymbol{y}, m, \boldsymbol{\Theta}_{\boldsymbol{X}|\boldsymbol{Y},m}). \tag{III.35}$$

  Since for each m, both the marginal distribution of $\boldsymbol{Y}$ and the conditional distribution of $\boldsymbol{X}|\boldsymbol{y}$ are Gaussian, (III.35) is equivalent to the standard GMM on $(\boldsymbol{X}, \boldsymbol{Y})$ given in (III.2). Therefore, the parameters of (III.35), i.e. $\{\pi_m, \boldsymbol{e}_m, \mathbf{R}_m, \mathbf{A}_m, \boldsymbol{b}_m, \mathbf{U}_m\}_{m=1}^{M}$ are computed from the parameters of the reference GMR.
- Third, $\{\mathbf{C}_m, \boldsymbol{d}_m, \mathbf{V}_m\}_{m=1}^{M}$, i.e., the parameters involving $\boldsymbol{Z}$, are initialized using the N_0 aligned $(\boldsymbol{z}, \boldsymbol{x})$ data. Basically, this is done by evaluating (III.31) with the auxiliary variables involving $\boldsymbol{Z}$ being calculated using observed $\boldsymbol{z}$ data only, i.e. $\boldsymbol{z}_{1:N_0}$, corresponding $\boldsymbol{x}$ data, i.e. $\boldsymbol{x}_{1:N_0}$, and responsibilities for $n \in [1, N_0]$ given by (III.23).
- Finally, after the initialization is done, both the N_0 aligned $(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y})$ data and the remaining $N - N_0$ $(\boldsymbol{x}, \boldsymbol{y})$ data are used to train the IC-GMR. Most importantly, all data are used to *jointly update all IC-GMR parameters*, as opposed to the SC-GMR adaptation, where the reference model remains unchanged, i.e. its parameters are not influenced by the adaptation data $\boldsymbol{z}_{1:N_0}$.

The complete EM algorithm for the IC-GMR, including the initialization step, is schematized in Algorithm 2. The E-step boils down to the calculation of the responsibilities (III.23) and (III.24). The M-step computes the auxiliary quantities defined in (III.27) that speed up the update of the parameters (III.28)–(III.34).

III.5 Joint GMR

In this section we present the proposed *Joint* GMM generative model (J-GMM) and the associated inference equation. We also discuss the relationship with previous works, *i.e.* [74] and [56]. The Joint GMM on $(\mathbf{X},\mathbf{Y},\mathbf{Z})$ is defined as:

$$p(\mathbf{o}) = \sum_{m=1}^{M} p(m) p(\mathbf{o}|m; \boldsymbol{\Theta}_m) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{o}; \mu_m, \boldsymbol{\Sigma}_m), \tag{III.36}$$

where $\boldsymbol{\Theta}_m = \{\mu_m, \boldsymbol{\Sigma}_m\}$ are the parameters of the m-th Gaussian component, and thus $\boldsymbol{\Theta} = \cup_{m=1}^{M} \{\pi_m, \boldsymbol{\Theta}_m\}$. In order to derive the associated inference equation we first compute:

$$p(\mathbf{y}|\mathbf{z}) = \int_{\mathbf{X}} \sum_{m=1}^{M} p(\mathbf{x}, \mathbf{y}, m|\mathbf{z}) d\mathbf{x} = \sum_{m=1}^{M} p(m|\mathbf{z}) p(\mathbf{y}|\mathbf{z}, m). \tag{III.37}$$

Since the conditional and marginal distributions of a Gaussian are Gaussian as well, (III.37) is a GMM. Therefore, the J-GMR inference equation under the MMSE criterion turns out to be identical to the usual expression for a direct $\mathbf{Z}$-to-$\mathbf{Y}$ GMR, *i.e.* (III.4).[III.4] At first sight, this may look a bit strange since this gives the impression of by-passing the information contained in $\mathbf{X}$. However, this is not the case: although its inference equation is identical, the complete "joint" process for GMR adaptation is not equivalent to a GMR build directly from $(\mathbf{z}, \mathbf{y})$ training data, *i.e.* the D-GMR. Indeed, as shown in the next section, the estimation of the J-GMR parameters with the EM algorithm uses all the available data, *i.e.* $\mathcal{D}_{\mathbf{xy}}$ and $\mathcal{D}_{\mathbf{z}}$, hence including all $\mathbf{x}$ data. In summary, the D-GMR and the J-GMR inference equations are identical but these two models differ by their underlying generative model and associated training procedure, leading to different parameter values (even when using the same adaptation dataset).

A J-GMM-based model has already been considered in [74] as the underlying generative model of $\mathbf{O}$ in the present speaker adaptation problem. However, [74] performs MAP inference instead of MMSE inference. More importantly, even if the underlying generative model is a J-GMM, the inference equation in [74] corresponds to the IC-GMR. In details, $p(\mathbf{o})$ in (1-2) of [74] corresponds to (III.36), while $p(\mathbf{y}|\mathbf{z})$ in (6) of [74] corresponds to (III.20). Indeed, this posterior PDF $p(\mathbf{y}|\mathbf{z})$ assumes no direct link between $\mathbf{Z}$ and $\mathbf{Y}$, which is correct for the IC-GMM but incorrect for the J-GMM. Thus the inference in [74] is not consistent with the J-GMM.[III.5]

[III.4] Alternately a MAP estimator can be used by taking the argmax of (III.37).

[III.5] Note that in [74], the details of the derivation of the intermediate form (6) into the GMR form (7) are not provided. In contrast, we provided detailed derivation in [56], where (19) is shown to result into two equivalent forms of a GMR expression (25) and (26). Also, to be fully precise, (7) in [74] corresponds to (26) in [56] up to two differences that we interpret as typos: First, the term $\boldsymbol{\Sigma}_m^{(x,x)}$ in (9) of [74] should be $\boldsymbol{\Sigma}_m^{(x,x)^{-1}}$, see e.g., (5) in [56]; and second, a right-sided term $\boldsymbol{\Sigma}_m^{(x,x)^{-1}}$ is missing in (10) in [74], i.e. $\boldsymbol{\Sigma}_m^{(a,s)} \boldsymbol{\Sigma}_m^{(y,y)^{-1}} \boldsymbol{\Sigma}_m^{(y,x)}$ should be $\boldsymbol{\Sigma}_m^{(a,s)} \boldsymbol{\Sigma}_m^{(y,y)^{-1}} \boldsymbol{\Sigma}_m^{(y,x)} \boldsymbol{\Sigma}_m^{(x,x)^{-1}}$ (see (26) in [56]). Without this matrix, "unnormalized" input data are propagated into the mapping process.

Algorithm 2: EM algorithm for integrated cascaded-GMR (IC-GMR) with partially missing data $\mathcal{Z}$.

Initialization

Use EM for GMM over $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ to compute $\boldsymbol{\Theta}_{\mathbf{J}}^{\text{in}}$:
$\left\{\pi_m^{\text{in}}, \boldsymbol{e}_m^{\text{in}}, \mathbf{R}_m^{\text{in}}, \mathbf{A}_m^{\text{in}}, \boldsymbol{b}_m^{\text{in}}, \mathbf{U}_m^{\text{in}}\right\}_{m=1}^M$.
for $n := 1 : N_0$ **do**
  $\boldsymbol{z}'_{nm} = \boldsymbol{z}_n, \forall m.$
end
for $m := 1 : M$ **do**
  for $n := 1 : N_0$ **do**
    Set γ_{nm}^{in} using (III.24) with $\boldsymbol{\Theta}_{\mathbf{J}}^{\text{in}}$.
  end
  $S_{\boldsymbol{Z}',m}^{\text{in}} = \sum_{n=1}^{N_0} \gamma_{nm}^{\text{in}} \boldsymbol{z}'_{nm}$ $S_{\boldsymbol{Z}'\boldsymbol{X},m}^{\text{in}} = \sum_{n=1}^{N_0} \gamma_{nm}^{\text{in}} \boldsymbol{z}'_{nm} \boldsymbol{x}_n^\top$ $S_{\boldsymbol{Z}'\boldsymbol{Z}',m}^{\text{in}} = \sum_{n=1}^{N_0} \gamma_{nm}^{\text{in}} \boldsymbol{z}'_{nm} (\boldsymbol{z}'_{nm})^\top$
end
Use (III.31), (III.34) with the previous auxiliary variables to compute $\left\{\mathbf{C}_m^{\text{in}}, \boldsymbol{d}_m^{\text{in}}, \mathbf{V}_m^{\text{in}}\right\}_{m=1}^M$.
Set $\boldsymbol{\Theta}^{(0)} = \boldsymbol{\Theta}^{\text{in}}$ and $i = 1$.
while *Not convergence* **do**
  E-step
  for $n := 1 : N_0$ **do**
    $p(\boldsymbol{o}_n|\boldsymbol{\Theta}^{(i)}) = \sum_{m=1}^M p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m^{(i)})$, using (III.5).
    for $m := 1 : M$ **do**
      $\gamma_{nm}^{(i)} = \frac{p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m^{(i)})}{p(\boldsymbol{o}_n|\boldsymbol{\Theta}^{(i)})}$.
    end
  end
  for $n := N_0 + 1 : N$ **do**
    $p(\boldsymbol{j}_n|\boldsymbol{\Theta}^{(i)}) = \sum_{m=1}^M p(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_{\mathbf{J},m}^{(i)})$, using (III.35).
    for $m := 1 : M$ **do**
      $\gamma_{nm}^{(i)} = \frac{p(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_{\mathbf{J},m}^{(i)})}{p(\boldsymbol{j}_n|\boldsymbol{\Theta}_{\mathbf{J}}^{(i)})}$.
      $\boldsymbol{z}'_{nm} = \mathbf{C}_m^{(i)} \boldsymbol{x}_n + \boldsymbol{d}_m^{(i)}$.
    end
  end
  M-step
  for $m := 1 : M$ **compute**
    Auxiliary variables using (III.27)
    $\pi_m^{(i+1)}$ and $e_m^{(i+1)}$ using (III.28) and (III.29)
    $\mathbf{A}_m^{(i+1)}$ and $\boldsymbol{b}_m^{(i+1)}$ using (III.30)
    $\mathbf{C}_m^{(i+1)}$ and $\boldsymbol{d}_m^{(i+1)}$ using (III.31)
    $\mathbf{R}_m^{(i+1)}$,$\mathbf{U}_m^{(i+1)}$and $\mathbf{V}_m^{(i+1)}$using (III.32), (III.33) and (III.34)
  end
  i++
end

Remarkably, (III.5) characterizes the IC-GMR as a particular case of the J-GMR. In Section III.7, we show that this is also true at the mixture model level, i.e. the IC-GMM (III.5) is a particular case of the J-GMM (III.36) with (III.21). The matrix product $\boldsymbol{\Sigma}_{\mathbf{XZ},m}\boldsymbol{\Sigma}_{\mathbf{ZZ},m}^{-1}$ enables to go from $\mathbf{z}$ to $\mathbf{x}$, and then $\boldsymbol{\Sigma}_{\mathbf{YX},m}\boldsymbol{\Sigma}_{\mathbf{XX},m}^{-1}$ enables to go from $\mathbf{x}$ to $\mathbf{y}$, so that the IC-GMR goes from $\mathbf{z}$ to $\mathbf{y}$ "passing through $\mathbf{x}$". In contrast, the J-GMR enables to go directly from $\mathbf{z}$ to $\mathbf{y}$, though again, it is not equivalent to the $\mathbf{Z}$-$\mathbf{Y}$ D-GMR since $\mathbf{x}$ data are used at training time, as is shown in the next section.

III.6 EM algorithm for J-GMR (with missing data)

This section introduces the exact EM algorithm associated to the J-GMM, explicitly handling an incomplete adaptation dataset using the general methodology of missing data. The EM iteratively maximizes the expected complete-data log-likelihood. At iteration $i+1$, the E-step computes the auxiliary function $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$, where $\boldsymbol{\Theta}^{(i)}$ are the parameters

computed at iteration i. The M-step maximizes Q with respect to $\boldsymbol{\Theta}$, obtaining $\boldsymbol{\Theta}^{(i+1)}$. In the following we first describe the E and M steps, then we detail the initialization process. Finally we comment the link between the EM algorithms of the IC-GMM and J-GMM, and the differences between the proposed EM and the EM for J-GMM given in [74]. The associated source code is available at: `https://git.gipsa-lab.grenoble-inp.fr/cgmr.git`.

III.6.1 E-step

In order to derive the auxiliary function $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$, we follow the general methodology given in [4]–(Section 9.4) and [73]. In [56], we have shown that this leads to the general expression:

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}) = \sum_{n=1}^{N_0} \sum_{m=1}^{M} \gamma_{nm}^{(i+1)} \log p(\mathbf{o}_n, m; \boldsymbol{\Theta}_m) + \sum_{n=N_0+1}^{N} \sum_{m=1}^{M} \frac{1}{p(\boldsymbol{j}_n; \boldsymbol{\Theta}_{\boldsymbol{J}}^{(i)})} \int p(\mathbf{o}_n, m; \boldsymbol{\Theta}_m^{(i)}) \log p(\mathbf{o}_n, m; \boldsymbol{\Theta}_m) d\mathbf{z}_n, \quad \text{(III.38)}$$

where

$$\gamma_{nm}^{(i+1)} = \frac{p(\mathbf{o}_n, m; \boldsymbol{\Theta}^{(i)})}{p(\mathbf{o}_n; \boldsymbol{\Theta}^{(i)})}, \quad n \in [1, N_0], \quad \text{(III.39)}$$

are the so-called *responsibilities* (of component m explaining observation $\mathbf{o}_n$) [4]. Eq. (III.38) is valid for any mixture model on i.i.d. vectors $(\boldsymbol{J}, \mathbf{Z})$ with partly missing $\mathbf{z}$ data. Here we study the particular case of the J-GMM. For this aim, we denote $\mu_{\mathbf{Z}|\boldsymbol{j}_n,m}^{(i)}$ the posterior mean of $\mathbf{Z}$ given $\boldsymbol{j}_n$ and given that the data were generated by the m-th Gaussian component with parameters $\boldsymbol{\Theta}_m^{(i)}$:

$$\mu_{\mathbf{Z}|\boldsymbol{j}_n,m}^{(i)} = \mu_{\mathbf{Z},m}^{(i)} + \boldsymbol{\Sigma}_{\mathbf{ZJ},m}^{(i)} \left(\boldsymbol{\Sigma}_{\mathbf{JJ},m}^{(i)}\right)^{-1} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}^{(i)}). \quad \text{(III.40)}$$

Let us define $\mathbf{o}'_{nm} = [\boldsymbol{j}_n^\top, \mu_{\mathbf{Z}|\boldsymbol{j}_n,m}^{(i)\top}]^\top$ if $n \in [N_0+1, N]$, i.e. $\mathbf{o}'_{nm}$ is an "augmented" observation vector in which for $n \in [N_0+1, N]$ the missing data vector $\mathbf{z}_n$ is replaced with (III.40). Let us arbitrarily extend $\mathbf{o}'_{nm}$ with $\mathbf{o}'_{nm} = \mathbf{o}_n$ for $n \in [1, N_0]$, and let us extend the definition of the responsibilities to the incomplete data vectors $\boldsymbol{j}_n$:

$$\gamma_{nm}^{(i+1)} = \frac{p(\boldsymbol{j}_n, m; \boldsymbol{\Theta}_{\boldsymbol{J}}^{(i)})}{p(\boldsymbol{j}_n; \boldsymbol{\Theta}_{\boldsymbol{J}}^{(i)})}, \quad n \in [N_0+1, N]. \quad \text{(III.41)}$$

Then, $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ is given by (III.43). The proof is given in Appendix III.12. The first double sum in (III.43) is similar to the one found in the usual EM for GMM (without missing data), except that for $n \in [N_0+1, N]$ missing $\mathbf{z}$ data are replaced with their estimate using the corresponding $\mathbf{x}$ and $\mathbf{y}$ data and the current parameter values, and responsibilities are calculated using available data only. The second term is a correction term that, as seen below, modifies the estimation of the covariance matrices $\boldsymbol{\Sigma}_m$ in the M-step to take into account the missing data.

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_{nm}^{(i+1)} \left(\log \pi_m - \frac{\log |\boldsymbol{\Sigma}_m| + (\mathbf{o}'_{nm} - \mu_m)^\top \boldsymbol{\Sigma}_m^{-1} (\mathbf{o}'_{nm} - \mu_m)}{2} \right) \quad \text{(III.42)}$$

$$- \frac{1}{2} \sum_{m=1}^{M} \left(\sum_{n=N_0+1}^{N} \gamma_{nm}^{(i+1)} \right) \mathrm{Tr} \left[\boldsymbol{\Lambda}_{\mathbf{ZZ},m} (\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{(i)})^{-1} \right]. \quad \text{(III.43)}$$

III.6.2 M-step

Priors: Maximization of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to the priors π_m is identical to the classical case of GMM without missing data [4]. For $m \in [1, M]$, we have:

$$\pi_m^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)}. \quad \text{(III.44)}$$

Mean vectors: For $m \in [1, M]$, derivating $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to μ_m and setting the result to zero leads to:

$$\mu_m^{(i+1)} = \frac{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \mathbf{o}'_{nm}}{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)}}. \quad \text{(III.45)}$$

This expression is the empirical mean, similar to the classical GMM case, except for the specific definition of observation vectors and responsibilities for $n \in [N_0+1, N]$.

Covariance matrices: Let us first express the trace in (III.43) as a function of $\boldsymbol{\Sigma}_m^{-1}$ by completing $(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{(i)})^{-1}$ with zeros to obtain the matrix $(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{0\,(i)})^{-1}$:

$$(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{0\,(i)})^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{(i)})^{-1} \end{bmatrix}. \tag{III.46}$$

Thus, $\text{Tr}\left[\boldsymbol{\Lambda}_{\mathbf{ZZ},m}(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{(i)})^{-1}\right] = \text{Tr}\left[\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{0\,(i)})^{-1}\right]$, and by canceling the derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{\Sigma}_m^{-1}$ we get:

$$\boldsymbol{\Sigma}_m^{(i+1)} = \frac{1}{\sum_{n=1}^{N}\gamma_{nm}^{(i+1)}}\left[\sum_{n=1}^{N}\gamma_{nm}^{(i+1)}(\mathbf{o}'_{nm}-\mu_m)(\mathbf{o}'_{nm}-\mu_m)^\top + \left(\sum_{n=N_0+1}^{N}\gamma_{nm}^{(i+1)}\right)(\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{0(i)})^{-1}\right]. \tag{III.47}$$

The first term is the empirical covariance matrix and is similar to the classical GMM without missing data, except again for the specific definition of observation vectors and responsibilities for $n \in [N_0+1, N]$. The second term can be seen as an additional correction term that deals with the absence of observed $\mathbf{z}$ data vectors for $n \in [N_0+1, N]$. We remark that $\boldsymbol{\Sigma}_m^{(i+1)}$ depends on all the terms of $\boldsymbol{\Sigma}_m^{(i)}$ obtained at previous iteration, since $\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{(i)} = \left[(\boldsymbol{\Sigma}_m^{(i)})^{-1}\right]_{\mathbf{ZZ}} \neq (\boldsymbol{\Sigma}_{\mathbf{ZZ},m}^{(i)})^{-1}$.

III.6.3 EM Initialization

The initialization of the proposed EM algorithm takes a very peculiar aspect. Indeed, as a result of the nature of the adaptation process, the reference $\mathbf{X}$-$\mathbf{Y}$ GMM model is used to initialize the marginal parameters in $(\mathbf{X}, \mathbf{Y})$. As for the $\mathbf{Z}$ parameters, we adopt the two possible following strategies. Strategy 1 is data-driven: The marginal parameters in $\mathbf{Z}$ are initialized using the adaptation data $\mathcal{D}_{\mathbf{z}} = \{\mathbf{z}_n\}_{n=1}^{N_0}$. The cross-term parameters in $(\mathbf{Z}, \mathbf{X})$ and $(\mathbf{Z}, \mathbf{Y})$ are initialized by constructing the sufficient statistics using $\{\mathbf{z}_n, \mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_0}$. Since the number of adaptation data is limited, and the related statistics may be poorly reliable, we also propose Strategy 2 which is a "blind" strategy: The $\mathbf{ZX}$ cross-covariance matrix is set to the identity matrix, the $\mathbf{ZY}$ cross-covariance matrix is set to zero and the covariance of $\mathbf{Z}$ is set to the covariance of $\mathbf{X}$. As shown in Section III.8, this simple blind initialization exhibited significantly better performance than the one exploiting the statistics of the adaptation set in our experiments, for small adaptation sets. Finally, remember that, whatever the initialization, *all model parameters are jointly updated* by alternating the E and M steps, using both reference data $\mathcal{D}_{\mathbf{xy}}$ and aligned adaptation data $\mathcal{D}_{\mathbf{z}}$.

III.7 Discussion

We have seen in Section III.5 that the IC-GMM is a particular (constrained) version of the J-GMM. However, the EM for the IC-GMM presented in Section III.4 is *not* derivable as a particular case of the EM for J-GMM provided in the previous section. More precisely, if one attempts to estimate the IC-GMR parameters with the EM algorithm introduced in the previous section, one should constrain the M-step by (III.21). Naturally, the complexity of the resulting constrained algorithm would be much higher than the complexity of the (unconstrained) EM of Section III.4. Consequently, even if the IC-GMR and the J-GMR models are closely related, the two learning algorithms are intrinsically different. This difference arises from the fact that the IC-GMM deals with constrained covariance matrices, whereas the J-GMM uses fully free covariance matrices.

We show here that the IC-GMM (III.5) is a particular case of the J-GMM (III.36) with (III.21). Without loss of generality, the density components of the J-GMM can be rewritten as:

$$p(\mathbf{o}|m) = \pi_m p(\mathbf{y}|m)p(\mathbf{x}|\mathbf{y},m)p(\mathbf{z}|\mathbf{x},\mathbf{y},m), \tag{III.48}$$

where all pdfs are Gaussian. Here the conditional pdf of $\mathbf{Z}$ depends on both $\mathbf{x}$ and $\mathbf{y}$, whereas in the IC-GMM it depends only on $\mathbf{x}$ (see Fig. III.1). Setting $p(\mathbf{z}|\mathbf{x},\mathbf{y},m) = p(\mathbf{z}|\mathbf{x},m)$ is equivalent to say that $\mathbf{Z}$ and $\mathbf{Y}$ are *conditionally independent* given $\mathbf{x}$, which can be expressed equivalently as $p(\mathbf{y},\mathbf{z}|\mathbf{x},m) = p(\mathbf{y}|\mathbf{x},m)p(\mathbf{z}|\mathbf{x},m)$ [4]–Section 8.2. Let us denote $\mathbf{U} = [\mathbf{Y}^\top, \mathbf{Z}^\top]^\top$. $p(\mathbf{u}|\mathbf{x},m)$ is a Gaussian pdf with covariance matrix $\boldsymbol{\Sigma}_{\mathbf{UU}|\mathbf{x},m} = \boldsymbol{\Sigma}_{\mathbf{UU},m} - \boldsymbol{\Sigma}_{\mathbf{UX},m}\boldsymbol{\Sigma}_{\mathbf{XX},m}^{-1}\boldsymbol{\Sigma}_{\mathbf{XU},m}$ [4]–Section 2.3. It is easy to show that the block diagonal term of this matrix is $\boldsymbol{\Sigma}_{\mathbf{YZ},m} - \boldsymbol{\Sigma}_{\mathbf{YX},m}\boldsymbol{\Sigma}_{\mathbf{XX},m}^{-1}\boldsymbol{\Sigma}_{\mathbf{XZ},m}$. Therefore, the conditional independence holds if and only if this block-diagonal term is null, i.e. (III.21). Alternately, we can write $p(\mathbf{o}|m)$ as a multivariate Gaussian and decompose the argument of the exponential function: it is then easy to show that $p(\mathbf{z}|\mathbf{x},\mathbf{y},m) = p(\mathbf{z}|\mathbf{x},m)$ for all values of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ (and m), if and only if all entries of $\boldsymbol{\Lambda}_{\mathbf{ZY},m} (= \boldsymbol{\Lambda}_{\mathbf{YZ},m}^\top)$ are zero, for all $m \in [1, M]$. Of course the two conditions are equivalent: Since $\boldsymbol{\Lambda}_{\mathbf{UU},m} = \boldsymbol{\Sigma}_{\mathbf{UU}|\mathbf{x},m}^{-1}$ [4]–(2.79) and (2.82), $\boldsymbol{\Lambda}_{\mathbf{UU},m}$ is block-diagonal if and only if $\boldsymbol{\Sigma}_{\mathbf{UU}|\mathbf{x},m}$ is block-diagonal.

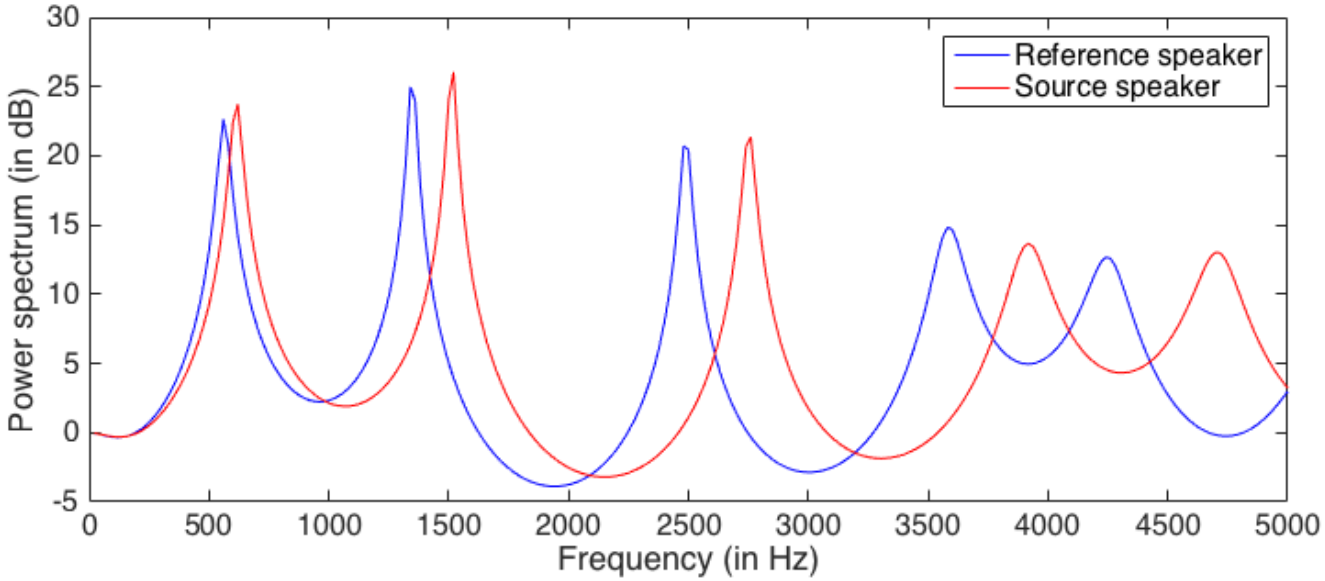


Figure III.2: *Power spectra generated by VLAM for the same articulatory configuration but for two different vocal tract lengths.*

III.8 Experiments

The performance of the J-GMR was evaluated on the speech acoustic-to-articulatory inversion task (i.e. recovering movement of the tongue, lips, jaw and velum from speech acoustics), and compared to the D-GMR, SC-GMR and IC-GMR. Two series of experiments were conducted: the first one on synthetic data, the second one on real data.

III.8.1 Experimental Set Up

Synthetic Data Experiments on synthetic data were conducted using a so-called articulatory synthesizer. This allowed us to carry out a first investigation of the J-GMR behavior by controlling finely the structure of the adaptation dataset (as opposed to the real data of Section III.8.1). A synthetic dataset of vowels was thus generated using the Variable Linear Articulatory Model (VLAM) [75]. VLAM consists of a vocal tract model driven by seven control parameters (lips aperture and protrusion; jaw; tongue body, dorsum and apex; velum). For a given articulatory configuration, VLAM calculates the corresponding area function using 29 tubes of variable length and then deduces the corresponding spectrum using acoustic simulation [76]. Among other articulatory synthesizers, VLAM is of particular interest in our study. Indeed, it integrates a model of the vocal tract growth and enables to generate two different spectra from the same articulatory configuration but different vocal tract length. We used this feature to simulate a parallel acoustic-articulatory dataset for two speakers (reference and source) with different vocal tract length corresponding to speaker age of 25 years and 17 years respectively. The difference in vocal tract length induces a shift of the formants along the frequency axis as illustrated in Fig. III.2. Moreover, this shift is non-linear, justifying the use of a non-linear (or locally linear) mapping model such as the GMR.

We generated a dataset of $(\mathbf{z}, \mathbf{x}, \mathbf{y})$ triplets structured into four clusters simulating the 4 following vowels: /a/, /i/, /u/, /ə/. In these experiments, the spectrum is described by the position and the amplitude of the 4 first formants, which are easily captured on such synthetic data, hence 8-dimensional $\mathbf{x}$ and $\mathbf{z}$ observations. We generated $20,000$ triplets ($5,000$ for each of the 4 vowels). These data are displayed in Fig. III.3 (red points) along with a selection of 467 adaptation vectors (green points), in the two first formant frequencies (F1-F2) plane.

MOCHA EMA Data Experiments on real data were conducted using electromagnetic articulatory (EMA) recordings. We used the publicly available Multichannel Articulatory Database (MOCHA) [77] provided by the Center for Speech Technology Research (University of Edinburgh). It includes acoustic-articulatory data of two speakers: fsew0 (female) and msak0 (male). Both speakers uttered 460 sentences extracted from the British TIMIT corpus, representing 20.6 min of speech for fsew0, and 17.4 min of speech for msak0.

Mel-frequency cepstral coefficients (MFCC) were used here to represent the acoustic content of the speech signal. Each audio observation ($\mathbf{x}$ and $\mathbf{z}$) was a 26-dimensional vector composed of 13 MFCC coefficients and their first derivatives. These vectors were extracted from the 16-kHz speech waveform every 10 ms, leading to a total of about $123,800$ vectors for fsew0 and of about $104,600$ vectors for msak0.

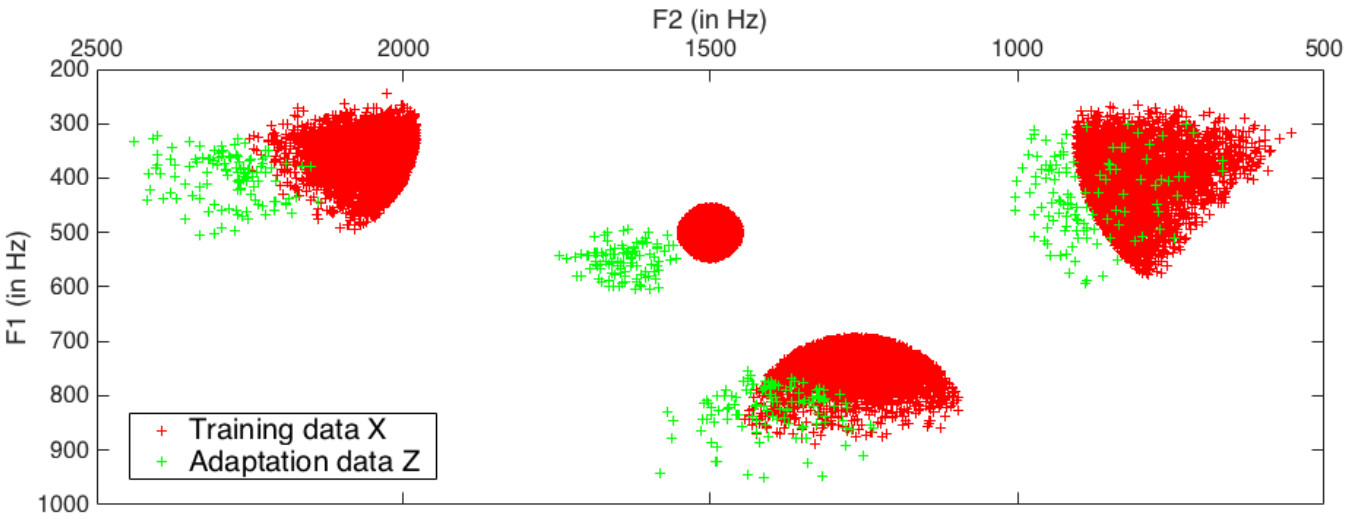


Figure III.3: *Synthetic data generated using VLAM in the F2-F1 acoustic space.*

Regarding the articulatory data, each observation $\mathbf{y}$ was a 14-dimensional vector gathering the 2D coordinates of 7 electromagnetic actuation coils describing the lips, tongue, jaw and velum positions in the midsagittal plane of the reference speaker's vocal tract, every 10 ms. These articulatory data were normalized following the procedure described in [78]. This normalization consists in centering and whitening the data (i.e. subtracting the mean value of each feature and dividing by its standard deviation) on a per-file basis. The mean (resp. standard deviation) of each feature was then low-pass filtered to alleviate the DC drift observed in the raw MOCHA database (see Fig. 3.6, p. 71 in [78]). Note that this has become a de-facto standard procedure, see [64] and [65].

We conducted two series of experiments: adaptation of reference speaker fsew0 to source speaker msak0 (denoted msak0→fsew0) and adaptation of reference speaker mask0 to source speaker fsew0 (denoted fsew0→msak0).

Experimental Protocol For the synthetic data, the complete set of $(\mathbf{z},\mathbf{x},\mathbf{y})$ triplets, are naturally aligned. For the MOCHA data, dynamic time warping (DTW) was used to time-align each of the sentences pronounced by the source speaker with the corresponding sentence pronounced by the reference speaker. The source speaker's acoustics was warped onto the reference speaker's acoustics (and by synchronicity onto the reference speaker's articulatory data).

For the experiments on the synthetic dataset, the EM algorithm for training the reference $\mathbf{X}$-$\mathbf{Y}$ model (and also the $\mathbf{Z}$-$\mathbf{X}$ model for the SC-GMR) was initialized using the k-means algorithm, repeated 5 times (only the best initial model was kept for training). For all EMs, the number of iterations was empirically set to 50. All methods were evaluated under a 30-fold cross-validation protocol: The data was divided in 30 subsets of approximate equal size, 29 subsets were used for training and 1 subset for test, considering all permutations. In each of the 30 folds, $k/30$ of the size of the training set was used for adaptation, with $k \in [1, 10]$. For a given value of k, we conducted 10 experiments with a different adaptation dataset. For each experiment, the optimal number of mixture components (within $M = 2, 4, 8, 12, 16, 20$) was determined using cross-validation during the training of the reference $\mathbf{X}$-$\mathbf{Y}$ model.[III.6] In the majority of these experiments, the optimal value of M was found to be 16. Similarly, the number K of components of the $\mathbf{Z}$-$\mathbf{X}$ model of the SC-GMR was set by cross-validation within the set $\{2, 4, 8, 12, 16\}$.

For the experiments with MOCHA, a similar procedure was used, though with different settings to adapt to the difference in dataset size and dimension. Here, all methods were evaluated under a 5-fold cross-validation protocol (four subsets for training and one subset for test, all of approximate equal size). In each of the five folds, $k/20$ of the size of the training set was used for adaptation, with $k \in [1, 10]$. This results in 50 experiments for each of the two aforementioned configurations (msak0→fsew0 and fsew0→msak0). As for M, the number of mixture components, cross-validation on the reference model for the MOCHA dataset let to an optimal value $M = 128$. However, the results for $M = 128, 64$, and 32 were found to be quite close, which is consistent with the results reported in previous literature [64]. Given that the J-GMR and IC-GMR models have more parameters than the reference model, and are thus more prone to overfitting, we chose to set $M = 32$. As for K, it was set using the same cross-validation procedure as for the synthetic data case.

For both synthetic and real data experiments, the performance was assessed by calculating the average Root Mean Squared Error (RMSE) between the articulatory trajectories estimated from the source speaker's acoustics, and the ones generated by the reference speaker (for the real data experiments, the reference speaker's acoustics and

[III.6] Remember that, in nature, the number of mixture components M of the J-GMR and IC-GMR is imposed by the reference model.

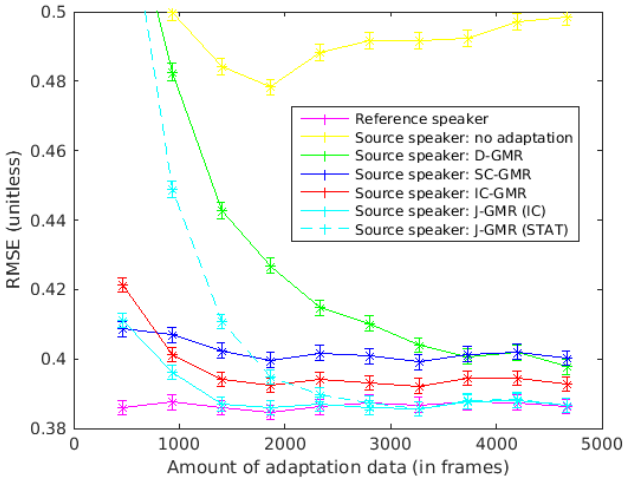


Figure III.4: *RMSE (unitless) of the Z-to-Y mapping as a function of the size of the adaptation data (in number of vectors), for D-GMR, SC-GMR, IC-GMR and J-GMR (lower and upper bounds are given by the X-Y mapping in magenta and the Z-to-Y mapping with no adaptation in yellow; error bars represent 95% confidence intervals.)*

articulatory test data were aligned on the source speaker's acoustics using DTW). 95% confidence interval of RMSE measures were obtained using paired t-tests. For the synthetic data, the vectors were generated independently (i.e. with no temporal structure), hence each vector provides an independent RMSE measure. As for the MOCHA dataset, independence between samples was assumed by considering the average RMSE on 5 consecutive frames, i.e. 50 ms. Note that the RMSE values for the synthetic data are unitless, since the VLAM articulatory data are arbitrary control parameters.

In order to discuss complexity and accuracy issues for the different models, we define the data-to-parameters ratio (DPR) as the total number of (scalar) data divided by the total number of (scalar) parameters to estimate. This simple measure provides prior information on how much the model is prone to overfit: The lower the DPR is (meaning less training data or more complex models) the more the model is prone to overfitting. Table III.1 presents the DPR values for the synthetic dataset and for MOCHA, for each model, and for the two extreme values of N_0 in the reported figures (see below). Note that the DPR is not a performance measure per se; it rather provides a potential explanation for the behavior of the models under evaluation. Indeed, in practice we observed that training models with DPR below 20 is risky, since the overfitting phenomenon may be predominant, impairing the generalization capabilities of the trained model. We can see in Table III.1 that all values are significantly larger than 20, except for the D-GMR with small N_0, as will be discussed later.

Table III.1: *Data-to-parameters ratio for the synthetic dataset and for MOCHA (for both speakers), for all models and for the two extreme values of N_0 reported in Fig. 4 and Fig. 5.*

Data	Synthetic		fsew0→msak0		msak0→fsew0	
N_0	Low	High	Low	High	Low	High
Reference	137	137	121	121	144	144
D-GMR	3	34	6	61	7	71
SC-GMR	89	100	69	88	81	104
IC-GMR	77	87	56	72	67	86
J-GMR	63	70	47	61	56	72

III.8.2 Results and Discussion

Synthetic Data The RMSE for the J-GMR, as well as for the D-GMR, SC-GMR and IC-GMR are plotted in Fig. III.4, as a function of N_0, the size of the adaptation set. The performance of the J-GMR, SC-GMR and IC-GMR are relatively close, and are clearly better than without adaptation and than the D-GMR, especially for low values of N_0. This latter result comes from the fact that the D-GMR exploits only the limited amount of reference speaker's articulatory data that can be associated with the source speaker's audio data, i.e. $\mathcal{D}_{\mathbf{zy}} = \{\mathbf{z}_n, \mathbf{y}_n\}_{n=1}^{N_0}$. As illustrated by the DPR values in Table III.1, this is a quite limited dataset compared to the dataset exploited by the C-GMR family, i.e. $\mathcal{D}_{\mathbf{z}} \cup$

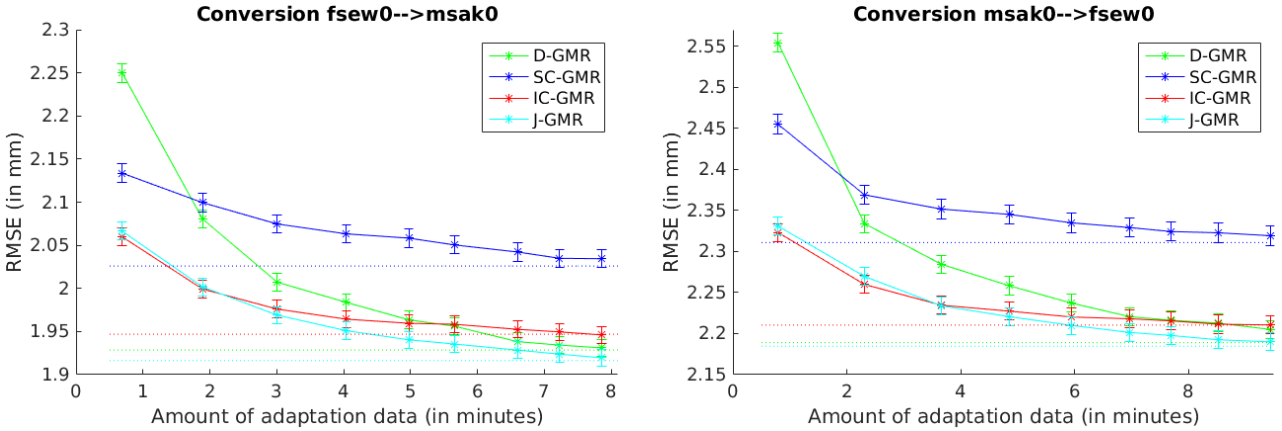


Figure III.5: *RMSE (in mm) with 95% confidence intervals for source speaker fsew0 (left) and msak0 (right) as a function of the amount of adaptation data, for D-, SC-, IC- and J-GMR, and their respective oracle in dotted lines.*

$\mathcal{D}_{\mathbf{xy}} = \{\mathbf{z}_n\}_{n=1}^{N_0} \cup \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$. For low N_0 values, this results in poor performance of the D-GMR, with possible severe overfitting. This tends to validate the benefit of exploiting all available $(\mathbf{x}, \mathbf{y})$ observations during the adaptation process, as done in the C-GMR framework.

As in [56], the IC-GMR performs better than the SC-GMR, except for the lower N_0 value. Importantly, we observe a systematic and statistically significant improvement of the proposed J-GMR over the IC-GMR, for all N_0 values. The gain of J-GMR over IC-GMR is within the approximate range 1.5%–2.5% of RMSE depending on N_0. Subsequently, the J-GMR also clearly outperforms the SC-GMR, except for the lower N_0 for which the difference between J-GMR and SC-GMR is not significant. These results illustrate that the J-GMR is able to better exploit the statistical relations between $\mathbf{z}$, $\mathbf{x}$ and $\mathbf{y}$ data compared to the other C-GMR models. Indeed, while the $\mathbf{Z}$-$\mathbf{X}$ and $\mathbf{X}$-$\mathbf{Y}$ statistical relationships are exploited by the SC-, IC- and J-GMR, only the latter directly exploits the $\mathbf{Z}$-$\mathbf{Y}$ statistical relationship. Therefore, only in the J-GMR the mapping is not forced to pass through $\mathbf{X}$, which is shown to be beneficial in this set of experiments.

Regarding the initialization strategy of the J-GMR, we notice that for the lower range of N_0 values the blind initialization strategy clearly outperforms the one based on the statistics of the adaptation set (denoted with the suffix "(STAT)" in Fig. III.4). This shows that in that case, the amount of adaptation data is not sufficient to calculate reliable statistics to be exploited in model parameter estimation. When the adaptation set grows in size (over approx. $3,000$ vectors), the difference in performance between the two initialization strategies becomes not significant, if any. Therefore, in the following, we will favor the blind initialization strategy.

MOCHA EMA The results of the experiments fsew0→msak0 and msak0→fsew0 on the MOCHA EMA dataset are shown in Figure III.5 respectively. Here also, the curves plot the RMSE against the amount of adaptation data. Similarly to [56] and similarly to the synthetic data experiments, for small adaptation sets, the IC-GMR clearly outperforms the D-GMR model. This is observed for the two source speakers msak0 and fsew0. The same tendency is observed with the proposed J-GMR model since the J-GMR performance is close to the IC-GMR performance (see below). SC-GMR also outperforms D-GMR, but only for the lowest N_0 value, since the difference between SC-GMR and IC-GMR is higher than in the synthetic data case. Altogether, these first general results confirm the results obtained on synthetic data and, again, they can be explained by the fact that the D-GMR exploits only the reference speaker's articulatory data that can be associated with the source speaker's audio data (see the small corresponding DPR values in Table III.1). This corroborates the benefit of (i) relying on an intermediate representation space, for instance the reference acoustic space $\mathbf{X}$, and (ii) exploiting all available $(\mathbf{x}, \mathbf{y})$ observations during the adaptation process. The fact that both J-GMR and IC-GMR clearly outperform SC-GMR everywhere seems to support the interest of a model structure where $\mathbf{X}$ is a single common representation space tied to both input $\mathbf{Z}$ and output $\mathbf{Y}$ at the mixture component level (as already observed for the IC-GMR in [56]).

As for the comparison between J-GMR and IC-GMR, these results also confirm the potential interest of using the J-GMR method over the IC-GMR. Indeed, while the two methods perform closely for tiny amounts of adaptation data, the J-GMR exhibits better results than the IC-GMR for larger amounts of adaptation data. More precisely, we can identify three different zones in the RMSE plots of both source speakers. First the data scarcity zone (below 3 min of adaptation data), where the IC-GMR shows equivalent performance than the J-GMR (for fsew0→mska0 conversion) or slightly better performance but not in a statistically significant manner (for the msak0→fsew0).

Second, the data abundance zone (above 7 min and more than 9 min of adaptation data for fsew0→msak0 and msak0→fsew0 respectively), where the D-GMR has enough data to show competitive performance compared to the J-GMR (see the correct DPR values for the D-GMR for high N_0 in Table III.1). At the same time, the RMSE of the IC-GMR

is here higher than the RMSE of D-GMR and J-GMR in a statistically significant manner. Therefore, it would appear that the constraint associated to the IC-GMR model penalizes its performance when enough adaptation data is available. This would suggest that more data implies more complex underlying links, some of which cannot be captured well by the IC-GMR model. This explanation is reinforced by the results under the so-called "oracle" settings, when all data is used at adaptation time, i.e. $N_0 = N$, which can be seen as the right limit of the plots. The result of the oracle settings for the four models are represented with dotted lines in Figure III.5. We can see that the J-GMR is able to better exploit the overall statistical correlations than the IC-GMR. Interestingly, the J-GMR oracle RMSE is below the D-GMR oracle RMSE, whereas the IC-GMR oracle RMSE is above. Hence, even for large adaptation data, it appears to be a good thing to exploit $\mathbf{x}$ at the mixture component level, but it is not such a good thing to do it in a too constrained manner.

This behavior is also observed, in a somewhat less intense manner, in the third zone (between 3 min and 7/9 min of adaptation data). Here the IC-GMR starts exhibiting worse performance than J-GMR (the difference is statistically significant from 5 min and 7 min of adaptation data for fsew0→msak0 and msak0→fsew0, respectively). At the same time, the D-GMR does not have enough data yet to approach the performance of the J-GMR. Our understanding is that, within this range, the complexity of the adaptation data overwhelms the IC-GMR, while not yet containing enough information to optimally exploit the $\mathbf{Z}$-$\mathbf{Y}$ link.

Overall, the privileged choice for cross-speaker acoustic-articulatory inversion appears to be the J-GMR. Indeed, if not enough adaptation data is available, the J-GMR has equivalent or close performance to the IC-GMR. In case a large amount of adaptation data is available, the J-GMR and the D-GMR perform closely, with a small advantage for the J-GMR, and this is further confirmed by the oracle results. Finally, the J-GMR has proven to be the most effective model in half-way situations between adaptation data scarcity and abundance.

III.9 Conclusions

We presented two models for acoustic-articulatory inversion adaptation. The first model does not consider a link between $\mathbf{Z}$ and $\mathbf{Y}$, and we denote it as IC-GMR. The second models exploits this link, and we refer to is as J-GMR (it is nothing but a tri-variate GMM with missing data). We provided the exact EM training algorithm for both the IC-GMR and the J-GMR, explicitly considering missing input data. We further discussed the theoretical links between these two models, both at the mixture level and at the EM algorithm level. We then applied these models to the cross-speaker acoustic-articulatory inversion task.

The reported experiments on both synthetic and real data show that the J-GMR and IC-GMR outperform the D-GMR, especially for small adaptation datasets. Moreover, we can provide an answer to the question stated in the introduction: Including an explicit link to the probabilistic model between the reference speaker's articulatory space and the source speaker's auditory space is beneficial for the present adaptation task. On the synthetic dataset, the J-GMR outperforms systematically the IC-GMR. On the real data, the J-GMR performs similarly to the IC-GMR for limited adaptation datasets but outperforms the IC-GMR for larger ones. The data-to-parameters ratio of the J-GMR is slightly inferior to the one of the IC-GMR, reflecting a slightly higher complexity of the J-GMR over the IC-GMR. However, in our experimental set-up this difference did not have a negative effect on the performance of the J-GMR.

III.10 Appendix: Derivation of Q for IC-GMR

Q is classically computed by taking the expectation of the complete-data log-likelihood with respect to the posterior distribution of the hidden variables given the observations (and the parameters at previous iteration):

$$Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}\right) = \sum_{n=1}^{N_0}\sum_{m=1}^{M} p\left(m|\boldsymbol{o}_n, \boldsymbol{\Theta}^{(i)}\right) \log p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m) + \sum_{n=N_0+1}^{N}\sum_{m=1}^{M} \int_{\mathbb{R}^{D_Z}} p\left(\boldsymbol{z}_n, m|\boldsymbol{j}_n, \boldsymbol{\Theta}^{(i)}\right) \log p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m) \mathrm{d}\boldsymbol{z}_n. \tag{III.49}$$

With definition (III.23) and multiplying and dividing the terms of the second double sum by $p(\boldsymbol{j}_n, \boldsymbol{\Theta}^{(i)})$, (III.22) follows immediately. Injecting (III.5)–(III.9) into the first double sum of (III.22) leads to the first double sum of (III.25). As for the second double sum, we remark that:

$$\begin{aligned}
&\int_{\mathbb{R}^{D_Z}} p\left(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m^{(i)}\right) \log p(\boldsymbol{o}_n, m|\boldsymbol{\Theta}_m) \mathrm{d}\boldsymbol{z}_n \\
&= p\left(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_m^{(i)}\right) \left[\log p\left(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_m\right) + \quad + \int_{\mathbb{R}^{D_Z}} p\left(\boldsymbol{z}_n, m|\boldsymbol{j}_n, \boldsymbol{\Theta}_m^{(i)}\right) \log p\left(\boldsymbol{z}_n, m|\boldsymbol{j}_n, \boldsymbol{\Theta}_m\right) \mathrm{d}\boldsymbol{z}_n\right].
\end{aligned}$$

Factor $p(\boldsymbol{j}_n|\boldsymbol{\Theta}^{(i)})$ together with $p(\boldsymbol{j}_n, m|\boldsymbol{\Theta}_m^{(i)})$ form the responsibilities (III.24), and the integral term is responsible for the term $-\mathcal{M}(\mathbf{C}_m^{(i)}\boldsymbol{x}_n + \boldsymbol{d}_m^{(i)} - \mathbf{C}_m\boldsymbol{x}_n - \boldsymbol{d}_m; \mathbf{V}_m) - \text{Tr}[\mathbf{V}_m^{-1}\mathbf{V}_m^{(i)}]$ of (III.25), that is equivalent in the case of missing data to the term $-\mathcal{M}(\boldsymbol{z}_n - \mathbf{C}_m\boldsymbol{x}_n - \boldsymbol{d}_m; \mathbf{V}_m)$ present in the first double sum of (III.25).

III.11 Appendix: Maximization of Q for IC-GMR

In this appendix we present the derivations for the M-step. All formulas start by taking the derivative of Q as expressed in (III.26).

Constant vectors and transition matrices For $m \in [1, M]$, we have:

$$\frac{\partial Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})}{\partial \boldsymbol{e}_m} = \mathbf{R}_m^{-1} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{y}_n - \boldsymbol{e}_m).$$

Setting this expression to zero leads to:

$$\boldsymbol{e}_m = \frac{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \boldsymbol{y}_n}{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)}}, \tag{III.50}$$

from which we obtain (III.29). This expression is very similar to the classical GMM case (see [4]), except for the specific definition of the responsibilities for $n \in [N_0 + 1, N]$. In the same line, taking the derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{b}_m$ and setting the result to zero leads to:

$$\boldsymbol{b}_m = \frac{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{x}_n - \mathbf{A}_m \boldsymbol{y}_n)}{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)}}. \tag{III.51}$$

Besides, for $m \in [1, M]$, we have:

$$\frac{\partial Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})}{\partial \mathbf{A}_m} = \mathbf{U}_m^{-1} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{x}_n - \mathbf{A}_m \boldsymbol{y}_n - \boldsymbol{b}_m) \boldsymbol{y}_n^\top.$$

Setting this expression to zero leads to:

$$\mathbf{A}_m = \left(\sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{x}_n - \boldsymbol{b}_m) \boldsymbol{y}_n^\top \right) \left(\sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \boldsymbol{y}_n \boldsymbol{y}_n^\top \right)^{-1}. \tag{III.52}$$

With the notation introduced in (III.27), Equ. (III.51) and (III.52) write:

$$\boldsymbol{b}_m = \frac{1}{S_m^{(i+1)}} \left(S_{\boldsymbol{X},m}^{(i+1)} - \mathbf{A}_m S_{\boldsymbol{Y},m}^{(i+1)} \right) \tag{III.53}$$

and

$$\mathbf{A}_m = \left(S_{\boldsymbol{XY},m}^{(i+1)} - \boldsymbol{b}_m S_{\boldsymbol{Y},m}^{(i+1)\top} \right) S_{\boldsymbol{YY},m}^{(i+1)\,-1}. \tag{III.54}$$

Replacing (III.53) into (III.54) we can deduce the final result for $\mathbf{A}_m$ and $\boldsymbol{b}_m$ given by (III.30)[III.7]. The optimal expression for $\mathbf{C}_m$ and $\boldsymbol{d}_m$ in (III.31) are obtained in the same manner.

Covariance matrices For $m \in [1, M]$, we have:

$$\frac{\partial Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})}{\partial \mathbf{R}_m^{-1}} = \frac{1}{2} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} \left(\mathbf{R}_m - (\boldsymbol{y}_n - \boldsymbol{e}_m)(\boldsymbol{y}_n - \boldsymbol{e}_m)^\top \right).$$

Setting this expression to zero leads to:

$$\mathbf{R}_m = \frac{1}{\sum_{n=1}^{N} \gamma_{nm}^{(i+1)}} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{y}_n - \boldsymbol{e}_m)(\boldsymbol{y}_n - \boldsymbol{e}_m)^\top = \frac{1}{S_m^{(i+1)}} \left(S_{\boldsymbol{YY},m}^{(i+1)} - S_{\boldsymbol{Y},m}^{(i+1)} * \boldsymbol{e}_m + \boldsymbol{e}_m \boldsymbol{e}_m^\top \right).$$

[III.7] Alternately one can solve for $\mathbf{A}_m$ first and place the result in (III.53) to obtain $\boldsymbol{b}_m$. The two solutions are equivalent, including in terms of computational cost.

We recall that $\mathbf{P}*\mathbf{Q} = \mathbf{P}\mathbf{Q}^\top + \mathbf{Q}\mathbf{P}^\top$ denotes the symmetrized outer product of $\mathbf{P}$ and $\mathbf{Q}$. From these equations the result in (III.32) follows immediately. In the same line, taking the derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to $\mathbf{U}_m^{-1}$ and setting the result to zero leads to:

$$\mathbf{U}_m = \frac{1}{S_m^{(i+1)}} \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{x}_n - \mathbf{A}_m \boldsymbol{y}_n - \boldsymbol{b}_m)(\boldsymbol{x}_n - \mathbf{A}_m \boldsymbol{y}_n - \boldsymbol{b}_m)^\top,$$

which drives us to (III.33). These expressions are of course empirical covariance matrices weighted by specific responsibilities. As for the maximization of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)})$ with respect to $\mathbf{V}_m$, we have the additional contribution of the Trace term due to the missing data. Setting the corresponding derivative to zero yields:

$$\mathbf{V}_m = \frac{1}{S_m^{(i+1)}} \left(\left(\sum_{n=N_0+1}^{N} \gamma_{nm}^{(i+1)} \right) \mathbf{V}_m^{(i)} + \sum_{n=1}^{N} \gamma_{nm}^{(i+1)} (\boldsymbol{z}'_{nm} - \mathbf{C}_m \boldsymbol{x}_n - \boldsymbol{d}_m)(\boldsymbol{z}'_{nm} - \mathbf{C}_m \boldsymbol{x}_n - \boldsymbol{d}_m)^\top \right).$$

The second term on the right side is an empirical covariance matrix and, again, it is similar to the classical GMM case [4] except for the specific definition of observation vectors and responsibilities for $n \in [N_0+1, N]$. The first term accounts for the missing data, i.e. $\boldsymbol{z}_n$ for $n \in [N_0+1, N]$. From this last equation (III.34) follows easily.

III.12 Appendix: Calculation of Q for the Joint GMR model

In [56], we provided the general expression (III.38) of Q which is valid for any mixture model of the form $p(\mathbf{o}; \boldsymbol{\Theta}) = \sum_{m=1}^{M} p(m) p(\mathbf{o}|m; \boldsymbol{\Theta}_m)$ parameterized by $\boldsymbol{\Theta}$, and applied on a i.i.d. random vector $\mathbf{O} = [\boldsymbol{J}^\top, \mathbf{Z}^\top]^\top$ with missing $\mathbf{z}$ data for $n \in [N_0+1, N]$. We now further calculate this expression for the J-GMM model defined in (III.36). Injecting (III.36) into (III.22) leads to:

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}) = \sum_{n=1}^{N_0} \sum_{m=1}^{M} \gamma_{nm}^{(i+1)} \left(\log \frac{\pi_m}{|\boldsymbol{\Sigma}_m|^{1/2}} - \frac{1}{2} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}} \right) \tag{III.55}$$

$$+ \sum_{n=N_0+1}^{N} \sum_{m=1}^{M} \frac{1}{p(\boldsymbol{j}_n; \boldsymbol{\Theta}_{\boldsymbol{J}}^{(i)})} \left(\left(\log \frac{\pi_m}{|\boldsymbol{\Sigma}_m|^{1/2}} \right) \int p(\mathbf{o}_n, m; \boldsymbol{\Theta}^{(i)}) d\mathbf{z}_n - \int \frac{1}{2} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} p(\mathbf{o}_n, m; \boldsymbol{\Theta}^{(i)}) \mathrm{d}\mathbf{z}_n \right), \tag{III.56}$$

where $\|\mathbf{x}\|_{\boldsymbol{\Sigma}} = \mathbf{x}^\top \boldsymbol{\Sigma}^{-1} \mathbf{x}$ stands for the Mahalanobis distance, $^{(i)}$ denotes the i-th iteration, and $\gamma_{nm}^{(i+1)}$ are defined in (III.39).

To further develop (III.56), we first notice that for Gaussian vectors we have:

$$\int_{\mathbf{Z}} p(\mathbf{o}_n, m; \boldsymbol{\Theta}^{(i)}) d\mathbf{z}_n = p(\boldsymbol{j}_n, m; \boldsymbol{\Theta}_{\boldsymbol{J}}^{(i)}) = \pi_m \mathcal{N}(\boldsymbol{j}; \mu_{\boldsymbol{J},m}^{(i)}, \boldsymbol{\Sigma}_{\boldsymbol{JJ},m}^{(i)}). \tag{III.57}$$

More importantly, we need to calculate:

$$f(\boldsymbol{j}_n) = \int_{\mathbf{Z}} -\frac{1}{2} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} p(\mathbf{o}_n, m; \boldsymbol{\Theta}^{(i)}) d\mathbf{z}_n = \int_{\mathbf{Z}} -\frac{1}{2} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} \frac{\pi_m e^{-\frac{1}{2}||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m}}}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_m^{(i)}|}} \mathrm{d}\mathbf{z}_n$$

The literature on matrix calculus provides a formula to integrate a quadratic term multiplied by another exponential quadratic term over the complete vector, but not over a subvector. Therefore, we need to separate the terms in $\boldsymbol{j}_n$ and the terms in $\mathbf{z}_n$. Using the precision matrix $\boldsymbol{\Lambda}_m = \boldsymbol{\Sigma}_m^{-1}$, we can first develop the quadratic term as:

$$\begin{aligned} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} = &-(\mathbf{o}_n - \mu_m)^\top \boldsymbol{\Lambda}_m (\mathbf{o}_n - \mu_m) = -(\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\boldsymbol{JJ},m} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}) \\ &- 2(\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\boldsymbol{JZ},m} (\mathbf{z}_n - \mu_{\mathbf{Z},m}) - (\mathbf{z}_n - \mu_{\mathbf{Z},m})^\top \boldsymbol{\Lambda}_{\mathbf{ZZ},m} (\mathbf{z}_n - \mu_{\mathbf{Z},m}), \end{aligned} \tag{III.58}$$

then reorganize it into (see [79]–Section 8.1.6):

$$\begin{aligned} ||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} = &-(\mathbf{o}_n - \mu_m)^\top \boldsymbol{\Lambda}_m (\mathbf{o}_n - \mu_m) = -\left| |\mathbf{z}_n - \mu_{\mathbf{Z},m} + \boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{-1} \boldsymbol{\Lambda}_{\mathbf{ZJ},m} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}) \right| |_{\boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{-1}} \\ &+ (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\boldsymbol{JZ},m} \boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{-1} \boldsymbol{\Lambda}_{\mathbf{ZJ},m} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}) \quad - (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\boldsymbol{JJ},m} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}). \end{aligned} \tag{III.59}$$

In the first term on the right hand side, we can recognize the posterior mean vector of $\mathbf{Z}$ given $\boldsymbol{j}_n$, i.e.:

$$\mu_{\mathbf{Z}|\boldsymbol{j}_n,m} = \mu_{\mathbf{Z},m} - \boldsymbol{\Lambda}_{\mathbf{ZZ},m}^{-1} \boldsymbol{\Lambda}_{\mathbf{ZJ},m} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}) = \mu_{\mathbf{Z},m} + \boldsymbol{\Sigma}_{\mathbf{ZJ},m} \boldsymbol{\Sigma}_{\boldsymbol{JJ},m}^{-1} (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}). \tag{III.60}$$

Besides, the two last terms of (III.59) can be factorized. We can recognize the inverse covariance matrix of $\boldsymbol{J}$ for component m, $\boldsymbol{\Sigma}^{-1}_{\mathbf{JJ},m} = \boldsymbol{\Lambda}_{\mathbf{JJ},m} - \boldsymbol{\Lambda}_{\mathbf{JZ},m}\boldsymbol{\Lambda}^{-1}_{\mathbf{ZZ},m}\boldsymbol{\Lambda}_{\mathbf{ZJ},m}$ [4]–(2.91), and thus we have:

$$||\mathbf{o}_n - \mu_m||_{\boldsymbol{\Sigma}_m} = ||\mathbf{z}_n - \mu_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{ZZ},m}} + ||\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}||_{\boldsymbol{\Sigma}_{\mathbf{JJ},m}}. \tag{III.61}$$

Of course, the same result holds at iteration $i+1$:

$$||\mathbf{o}_n - \mu^{(i)}_m||_{\boldsymbol{\Sigma}^{(i)}_m} = ||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}} + ||\boldsymbol{j}_n - \mu^{(i)}_{\boldsymbol{J},m}||_{\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}}. \tag{III.62}$$

Reinjecting (III.58) and (III.62) into (III.58) leads to:

$$\begin{aligned} f(\boldsymbol{j}_n) = \int_{\mathbf{Z}} \Big(-||\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{JJ},m}} - ||\mathbf{z}_n - \mu_{\mathbf{Z},m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{ZZ},m}} - 2(\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\mathbf{JZ},m}(\mathbf{z}_n - \mu_{\mathbf{Z},m})\Big) \\ \times \frac{\pi_m e^{-\frac{1}{2}||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}}} e^{-\frac{1}{2}||\boldsymbol{j}_n - \mu^{(i)}_{\boldsymbol{J},m}||_{\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}}}}{2\sqrt{(2\pi)^D|\boldsymbol{\Sigma}^{(i)}_m|}} \mathrm{d}\mathbf{z}_n. \end{aligned} \tag{III.63}$$

Separating $\boldsymbol{j}_n$ and $\mathbf{z}_n$, the calculation of $|\boldsymbol{\Sigma}^{(i)}_m|$ can be done by noting that:

$$|\boldsymbol{\Sigma}^{(i)}_m| = |\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}||\boldsymbol{\Sigma}^{(i)}_{\mathbf{ZZ},m} - \boldsymbol{\Sigma}^{(i)}_{\mathbf{ZJ},m}\boldsymbol{\Sigma}^{(i)-1}_{\mathbf{JJ},m}\boldsymbol{\Sigma}^{(i)}_{\mathbf{JZ},m}| = |\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}||\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}|, \tag{III.64}$$

and thus:

$$\begin{aligned} \frac{\pi_m e^{-\frac{1}{2}||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}}} e^{-\frac{1}{2}||\boldsymbol{j}_n - \mu^{(i)}_{\boldsymbol{J},m})||_{\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}}}}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}^{(i)}_m|}} &= \frac{e^{-\frac{1}{2}||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}}}}{\sqrt{(2\pi)^{D_\mathbf{Z}}|\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}|}} \times \frac{\pi_m e^{-\frac{1}{2}||\boldsymbol{j}_n - \mu^{(i)}_{\boldsymbol{J},m})||_{\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}}}}{\sqrt{(2\pi)^{D_\boldsymbol{J}}|\boldsymbol{\Sigma}^{(i)}_{\mathbf{JJ},m}|}} \\ &= \frac{e^{-\frac{1}{2}||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}}}}{\sqrt{(2\pi)^{D_\mathbf{Z}}|\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}|}} \times p(\boldsymbol{j}_n, m; \boldsymbol{\Theta}^{(i)}_{\boldsymbol{J},m}). \end{aligned} \tag{III.65}$$

Therefore, we have:

$$\begin{aligned} f(\boldsymbol{j}_n) = p(\boldsymbol{j}_n, m; \boldsymbol{\Theta}^{(i)}_{\boldsymbol{J}}) \int_{\mathbf{Z}} \Big(-\|\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}\|_{\boldsymbol{\Lambda}^{-1}_{\mathbf{JJ},m}} - ||\mathbf{z}_n - \mu_{\mathbf{Z},m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{ZZ},m}} - 2(\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\mathbf{JZ},m}(\mathbf{z}_n - \mu_{\mathbf{Z},m})\Big) \\ \times \frac{e^{-\frac{1}{2}||\mathbf{z}_n - \mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m}||_{\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}}}}{2\sqrt{(2\pi)^{D_\mathbf{Z}}|\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}|}} \mathrm{d}\mathbf{z}_n. \end{aligned} \tag{III.66}$$

After [79]–(351) and (357), we obtain:

$$\begin{aligned} f(\boldsymbol{j}_n) = p(\boldsymbol{j}_n, m|\boldsymbol{\Theta}^{(i)}_{\boldsymbol{J}})\Big(-\frac{1}{2}||\boldsymbol{j}_n - \mu_{\boldsymbol{J},m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{JJ},m}} - (\boldsymbol{j}_n - \mu_{\boldsymbol{J},m})^\top \boldsymbol{\Lambda}_{\mathbf{JZ},m}(\mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m} - \mu_{\mathbf{Z},m}) \\ -\frac{1}{2}||\mu^{(i)}_{\mathbf{Z}|\boldsymbol{j}_n,m} - \mu_{\mathbf{Z},m}||_{\boldsymbol{\Lambda}^{-1}_{\mathbf{ZZ},m}} - \frac{1}{2}\mathsf{Tr}\left[\boldsymbol{\Lambda}_{\mathbf{ZZ},m}\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}\right]\Big), \end{aligned}$$

which can be reorganized into:

$$f(\boldsymbol{j}_n) = -\frac{1}{2}p(\boldsymbol{j}_n, m; \boldsymbol{\Theta}^{(i)}_{\boldsymbol{J}}) \left(||\mathbf{o}'_{nm} - \mu_m||_{\boldsymbol{\Lambda}^{-1}_m} + \mathsf{Tr}\left[\boldsymbol{\Lambda}_{\mathbf{ZZ},m}\boldsymbol{\Lambda}^{(i)-1}_{\mathbf{ZZ},m}\right]\right). \tag{III.67}$$

Using (III.57), (III.67) and the extended definition of responsibilities for $n \in [N_0+1, N]$, (III.56) can be rewritten into (III.43). □

Chapter *IV*

Robust Deep Regression for Computer Vision

Abstract *We address the problem of how to robustly train a ConvNet for regression, or deep robust regression. Traditionally, deep regression employ the L_2 loss, known to be sensitive to outliers, i.e. samples that either lie at an abnormal distance away from the majority of the training samples, or that correspond to wrongly annotated targets. This means that, during backpropagation, outliers may bias the training process due to the high magnitude of their gradient. In this chapter, we propose DeepGUM: a deep regression model that is robust to outliers thanks to the use of a Gaussian-uniform mixture model. We derive an optimization algorithm that alternates between the unsupervised detection of outliers using expectation-maximization, and the supervised training with cleaned samples using stochastic gradient descent. DeepGUM is able to adapt to a continuously evolving outlier distribution, avoiding to manually impose any threshold on the proportion of outliers in the training set. Extensive experimental evaluations on four different tasks (facial and fashion landmark detection, age and head pose estimation) lead us to conclude that our novel robust technique provides reliability in the presence of various types of noise and protection against a high percentage of outliers.*

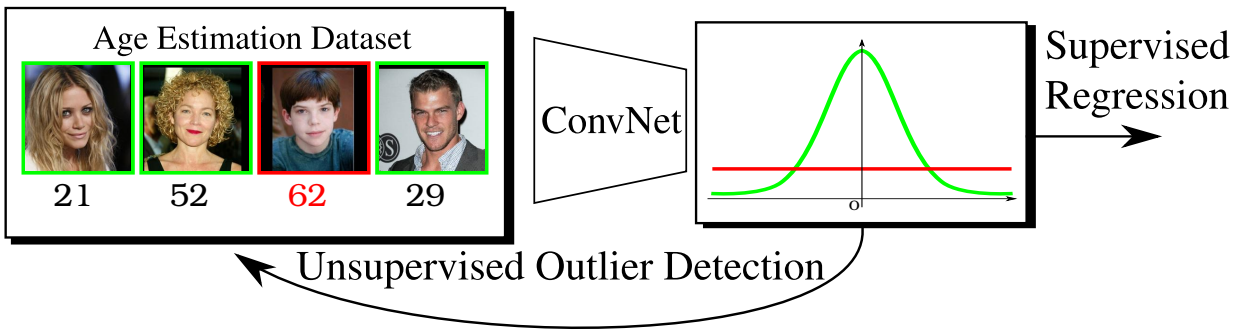


Figure IV.1: *A Gaussian-uniform mixture model is combined with a ConvNet architecture to downgrade the influence of wrongly annotated targets (outliers) on the learning process.*

Chapter Pitch

Methodological contribution A two-component mixture model on top of a pre-trained deep regressor can detect outliers in an unsupervised manner, allowing to automatically clean the training set.

$$\mathcal{L} = \sum_{n=1}^{N} r_n(\boldsymbol{\theta}^{(c)}) ||\boldsymbol{y}_n - \boldsymbol{\phi}(\boldsymbol{i}_n; \boldsymbol{w})||_2^2, \qquad r_n(\boldsymbol{\theta}^{(c)}) \sim \text{ inlier responsibility.}$$

Applicative task Robust deep regression in compute vision.

Interesting insight The responsibilities help back-propagating the gradient of the relevant samples. However, one must use the Euclidean distance and not the Mahalanobis distance in the loss. Otherwise, the most common errors (directions of higher variance) are mitigated and the network does not learn from them.

Dissemination DeepGUM was published at the European Conference on Computer Vision in 2018, see [80].

IV.1 Introduction

For the last decade, deep learning architectures have undoubtably established the state of the art in computer vision tasks such as image classification [81], [82] or object detection [83], [84]. These architectures, e.g. ConvNets, consist of several convolutional layers, followed by a few fully connected layers and by a classification softmax layer with, for instance, a cross-entropy loss. ConvNets have also been used for regression, i.e. predict continuous as opposed to categorical output values. Classical regression-based computer vision methods have addressed human pose estimation [85], age estimation [86], head-pose estimation [87], or facial landmark detection [88], to cite a few. Whenever ConvNets are used for learning a regression network, the softmax layer is replaced with a fully connected layer, with linear or sigmoid activations, and L_2 is often used to measure the discrepancy between prediction and target variables. It is well known that L_2-loss is strongly sensitive to outliers, potentially leading to poor generalization performance [89]. While robust regression is extremely well investigated in statistics, there has only been a handful of methods that combine robust regression with deep architectures.

We propose to mitigate the influence of outliers when deep neural architectures are used to learn a regression function, ConvNets in particular. More precisely, we investigate a methodology specifically designed to cope with two types of outliers that are often encountered: (i) samples that lie at an abnormal distance away from the other training samples, and (ii) wrongly annotated training samples. On the one hand, abnormal samples are present in almost any measurement system and they are known to bias the regression parameters. On the other hand, deep learning requires very large amounts of data and the annotation process, be it either automatic or manual, is inherently prone to errors. These unavoidable issues fully justify the development of robust deep regression.

The proposed method combines the representation power of ConvNets with the principled probabilistic mixture framework for outlier detection and rejection, e.g. Figure IV.1. We propose to use a Gaussian-uniform mixture (GUM) as the last layer of a ConvNet, and we refer to this combination as DeepGUM. The mixture model hypothesizes a Gaussian distribution for inliers and a uniform distribution for outliers. We interleave an EM procedure within stochastic gradient descent (SGD) to downgrade the influence of outliers in order to robustly estimate the network parameters. We empirically validate the effectiveness of the proposed method with four computer vision problems and associated datasets: facial and fashion landmark detection, age estimation, and head pose estimation. The standard regression measures are accompanied by statistical tests that discern between random differences and systematic improvements.

The remainder of the chapter is organized as follows. Section IV.2 describes the related work. Section IV.3 describes in detail the proposed method and the associated algorithm. Section IV.4 describes extensive experiments with several applications and associated datasets. Section IV.5 draws conclusions and discusses the potential of robust deep regression in computer vision.

IV.2 Related Work

Robust regression has long been studied in statistics [89]–[91] and in computer vision [92]–[94]. Robust regression methods have a high *breakdown point*, which is the smallest amount of outlier contamination that an estimator can handle before yielding poor results. Prominent examples are the least trimmed squares, the Theil-Sen estimator or heavy-tailed distributions [95]. Several robust training strategies for artificial neural networks are also available [96], [97].

M-estimators, sampling methods, trimming methods and robust clustering are among the most used robust statistical methods. M-estimators [89] minimize the sum of a positive-definite function of the residuals and attempt to reduce the influence of large residual values. The minimization is carried our with weighted least squares techniques, with no proof of convergence for most M-estimators. Sampling methods [93], such as least-median-of-squares or random sample consensus (RANSAC), estimate the model parameters by solving a system of equations defined for a randomly chosen data subset. The main drawback of sampling methods is that they require complex data-sampling procedures and it is tedious to use them for estimating a large number of parameters. Trimming methods [91] rank the residuals and down-weight the data points associated with large residuals. They are typically cast into a (non-linear) weighted least squares optimization problem, where the weights are modified at each iteration, leading to iteratively re-weighted least squares problems. Robust statistics have also been addressed in the framework of mixture models and a number of robust mixture models were proposed, such as Gaussian mixtures with a uniform noise component [47], [98], heavy-tailed distributions [24], trimmed likelihood estimators [99], [100], or weighted-data mixtures [17]. Importantly, it has been recently reported that modeling outliers with an uniform component yields very good performance [17], [98].

Deep robust classification was recently addressed, e.g. [101] assumes that observed labels are generated from true labels with unknown noise parameters: a probabilistic model that maps true labels onto observed labels is proposed

and an EM algorithm is derived. In [102] is proposed a probabilistic model that exploits the relationships between classes, images and noisy labels for large-scale image classification. This framework requires a dataset with explicit clean- and noisy-label annotations as well as an additional dataset annotated with a noise type for each sample, thus making the method difficult to use in practice. Classification algorithms based on a distillation process to learn from noisy data was recently proposed [103].

Recently, deep regression methods were proposed, e.g. [85], [88], [104]–[106]. Despite the vast robust statistics literature and the importance of regression in computer vision, at the best of our knowledge there has been only one attempt to combine robust regression with deep networks [107], where robustness is achieved by minimizing the Tukey's bi-weight loss function, i.e. an M-estimator. We take a radical different approach and propose to use robust mixture modeling within a ConvNet. We conjecture that while *inlier noise* follows a Gaussian distribution, *outlier errors* are uniformly distributed over the volume occupied by the data. Mixture modeling provides a principled way to characterize data points individually, based on posterior probabilities. We propose an algorithm that interleaves a robust mixture model with network training, i.e. alternates between EM and SGD. EM evaluates data-posterior probabilities which are then used to weight the residuals used by the network loss function and hence to downgrade the influence of samples drawn from the uniform distribution. Then, the network parameters are updated which in turn are used by EM. A prominent feature of the algorithm is that it requires neither annotated outlier samples nor prior information about their percentage in the data. This is in contrast with [102] that requires explicit inlier/outlier annotations and with [107] which uses a fixed hyperparameter ($c = 4.6851$) that allows to exclude from SGD samples with high residuals.

IV.3 Deep Regression with a Robust Mixture Model

We assume that the inlier noise follows a Gaussian distribution while the outlier error follows a uniform distribution. Let $\boldsymbol{i} \in \mathbb{R}^M$ and $\boldsymbol{y} \in \mathbb{R}^D$ be the input image and the output vector with dimensions M and D, respectively, with $D \ll M$. Let ϕ denote a ConvNet with parameters $\boldsymbol{w}$ such that $\boldsymbol{y} = \phi(\boldsymbol{i}, \boldsymbol{w})$. We aim to train a model that detects outliers and downgrades their role in the prediction of a network output, while there is no prior information about the percentage and spread of outliers. The probability of $\boldsymbol{y}$ conditioned by $\boldsymbol{i}$ follows a Gaussian-uniform mixture model (GUM):

$$p(\boldsymbol{y}|\boldsymbol{i}; \boldsymbol{\theta}, \boldsymbol{w}) = \pi \, \mathcal{N}(\boldsymbol{y}; \boldsymbol{\phi}(\boldsymbol{i}; \boldsymbol{w}), \boldsymbol{\Sigma}) + (1 - \pi) \, \mathcal{U}(\boldsymbol{y}; \gamma), \tag{IV.1}$$

where π is the prior probability of an inlier sample, γ is the normalization parameter of the uniform distribution and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ is the covariance matrix of the multivariate Gaussian distribution. Let $\boldsymbol{\theta} = \{\pi, \gamma, \boldsymbol{\Sigma}\}$ be the parameter set of GUM. At training we estimate the parameters of the mixture model, $\boldsymbol{\theta}$, and of the network, $\boldsymbol{w}$. An EM algorithm is used to estimate the former together with the responsibilities r_n, which are plugged into the network's loss, minimized using SGD so as to estimate the later.

IV.3.1 EM Algorithm

Let a training dataset consist of N image-vector pairs $\{\boldsymbol{i}_n, \boldsymbol{y}_n\}_{n=1}^N$. At each iteration, EM alternates between evaluating the expected complete-data log-likelihood (E-step) and updating the parameter set $\boldsymbol{\theta}$ conditioned by the network parameters (M-step). In practice, the E-step evaluates the posterior probability (responsibility) of an image-vector pair n to be an inlier:

$$r_n(\boldsymbol{\theta}^{(i)}) = \frac{\pi^{(i)} \mathcal{N}(\boldsymbol{y}_n; \boldsymbol{\phi}(\boldsymbol{i}_n, \boldsymbol{w}^{(c)}), \boldsymbol{\Sigma}^{(i)})}{\pi^{(i)} \mathcal{N}(\boldsymbol{y}_n; \boldsymbol{\phi}(\boldsymbol{i}_n, \boldsymbol{w}^{(c)}), \boldsymbol{\Sigma}^{(i)}) + (1 - \pi^{(i)}) \gamma^{(i)}}, \tag{IV.2}$$

where (i) denotes the EM iteration index and $\boldsymbol{w}^{(c)}$ denotes the currently estimated network parameters. The posterior probability of the n-th data pair to be an outlier is $1 - r_n(\boldsymbol{\theta}^{(i)})$. The M-step updates the mixture parameters $\boldsymbol{\theta}$ with:

$$\boldsymbol{\Sigma}^{(i+1)} = \sum_{n=1}^{N} r_n(\boldsymbol{\theta}^{(i)}) \boldsymbol{\delta}_n^{(i)} \boldsymbol{\delta}_n^{(i)\top}, \tag{IV.3}$$

$$\pi^{(i+1)} = \sum_{n=1}^{N} r_n(\boldsymbol{\theta}^{(i)}) / N, \tag{IV.4}$$

$$\frac{1}{\gamma^{(i+1)}} = \prod_{d=1}^{D} 2\sqrt{3\left(C_{2d}^{(i+1)} - \left(C_{1d}^{(i+1)}\right)^2\right)}, \tag{IV.5}$$

where $\boldsymbol{\delta}_n^{(i)} = \boldsymbol{y}_n - \boldsymbol{\phi}(\boldsymbol{i}_n; \boldsymbol{w}^{(c)})$, and C_1 and C_2 are the first- and second-order centered data moments computed using ($\delta_{nd}^{(i)}$ denotes the d-th entry of $\boldsymbol{\delta}_n^{(i)}$):

$$C_{1d}^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} \frac{(1 - r_n(\boldsymbol{\theta}^{(i)}))}{1 - \pi^{(i+1)}} \delta_{nd}^{(i)}, \; C_{2d}^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} \frac{(1 - r_n(\boldsymbol{\theta}^{(i)}))}{1 - \pi^{(i+1)}} \left(\delta_{nd}^{(i)}\right)^2. \tag{IV.6}$$

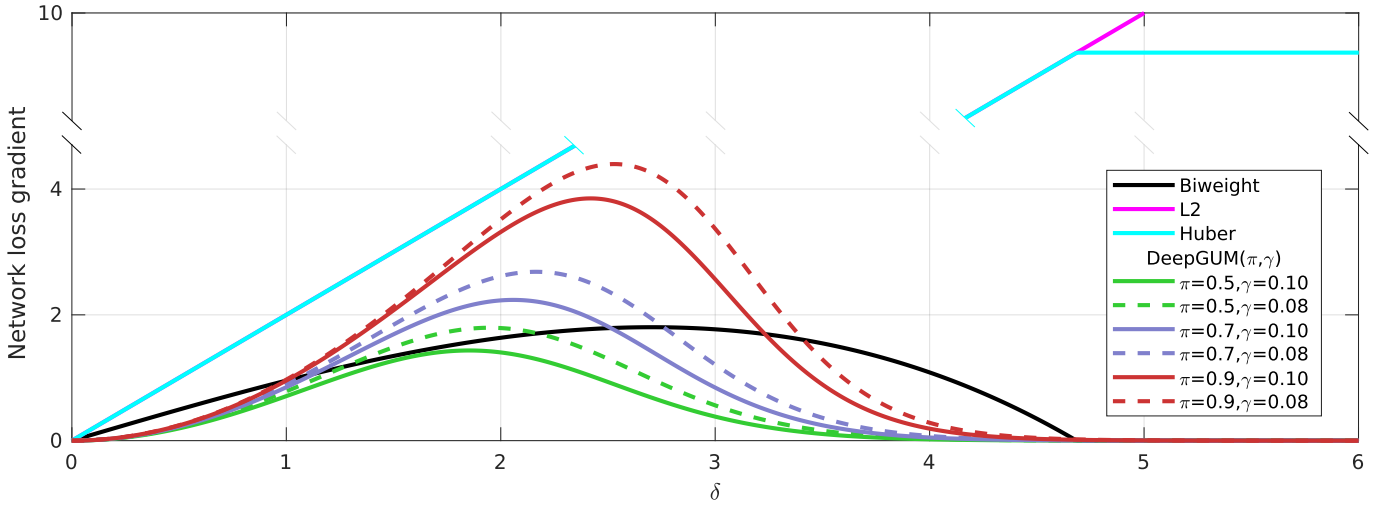


Figure IV.2: *Loss gradients for Biweight (black), Huber (cyan), L_2 (magenta), and DeepGUM (remaining colors). Huber and L_2 overlap up to $\delta = 4.6851$ (the plots are truncated along the vertical coordinate). DeepGUM is shown for different values of π and γ, although in practice they are estimated via EM. The gradients of DeepGUM and Biweight vanish for large residuals. DeepGUM offers some flexibility over Biweight thanks to π and γ.*

The iterative estimation of γ as just proposed has an advantage over using a constant value based on the volume of the data, as done in robust mixture models [98]. Indeed, γ is updated using the actual volume occupied by the outliers, which increases the ability of the algorithm to discriminate between inliers and outliers.

Another prominent advantage of DeepGUM for robustly predicting multidimensional outputs is its flexibility for handling the granularity of outliers. Consider for example to problem of locating landmarks in an image. One may want to devise a method that disregards outlying landmarks and not the whole image. In this case, one may use a GUM model for each landmark category. In the case of two-dimensional landmarks, this induces $D/2$ covariance matrices of size 2 (D is the dimensionality of the target space). Similarly one may use an coordinate-wise outlier model, namely D scalar variances. Finally, one may use an image-wise outlier model, i.e. the model detailed above. This flexibility is an attractive property of the proposed model as opposed to [107] which uses a coordinate-wise outlier model.

IV.3.2 Network Loss Function

As already mentioned we use SGD to estimate the network parameters $\boldsymbol{w}$. Given the updated GUM parameters estimated with EM, $\boldsymbol{\theta}^{(c)}$, the regression loss function is weighted with the responsibility of each data pair:

$$\mathcal{L}_{\text{DEEPGUM}} = \sum_{n=1}^{N} r_n(\boldsymbol{\theta}^{(c)}) ||\boldsymbol{y}_n - \boldsymbol{\phi}(\boldsymbol{i}_n; \boldsymbol{w})||_2^2. \tag{IV.7}$$

With this formulation, the contribution of a training pair to the loss gradient vanishes (i) if the sample is an inlier with small error ($||\boldsymbol{\delta}_n||_2 \to 0, r_n \to 1$) or (ii) if the sample is an outlier ($r_n \to 0$). In both cases, the network will not back propagate any error. Consequently, the parameters $\boldsymbol{w}$ are updated only with inliers. This is graphically shown in Figure IV.2, where we plot the loss gradient as a function of a one-dimensional residual δ, for DeepGUM, Biweight, Huber and L_2. For fair comparison with Biweight and Huber, the plots correspond to a unit variance (i.e. standard normal, see discussion following eq. (3) in [107]). We plot the DeepGUM loss gradient for different values of π and γ to discuss different situations, although in practice all the parameters are estimated with EM. We observe that the gradient of the Huber loss increases linearly with δ, until reaching a stable point (corresponding to $c = 4.6851$ in [107]). Conversely, the gradient of both DeepGUM and Biweight vanishes for large residuals (i.e. $\delta > c$). Importantly, DeepGUM offers some flexibility as compared to Biweight. Indeed, we observe that when the amount of inliers increases (large π) or the spread of outliers increases (small γ), the importance given to inliers is higher, which is a desirable property. The opposite effect takes place for lower amounts of inliers and/or reduced outlier spread.

IV.3.3 Training Algorithm

In order to train the proposed model, we assume the existence of a training and validation datasets, denoted $\mathcal{T} = \{\boldsymbol{i}_n^{\mathsf{T}}, \boldsymbol{y}_n^{\mathsf{T}}\}_{n=1}^{N_{\mathsf{T}}}$ and $\mathcal{V} = \{\boldsymbol{i}_n^{\mathsf{V}}, \boldsymbol{y}_n^{\mathsf{V}}\}_{n=1}^{N_{\mathsf{V}}}$, respectively. The training alternates between the unsupervised EM algorithm of Section IV.3.1 and the supervised SGD algorithm of Section IV.3.2, i.e. Algorithm 3. EM takes as input the training set,

Algorithm 3: DeepGUM training.

input : $\mathcal{T} = (\boldsymbol{i}_n^{\mathsf{T}}, \boldsymbol{y}_n^{\mathsf{T}})_{n=1}^{N_{\mathsf{T}}}$, $\mathcal{V} = \{\boldsymbol{i}_n^{\mathsf{V}}, \boldsymbol{y}_n^{\mathsf{V}}\}_{n=1}^{N_{\mathsf{V}}}$, and $\epsilon > 0$ (convergence threshold).

Initialization: Run SGD on $\mathcal{T}$ to minimize (IV.7). **repeat**

  EM algorithm: Unsupervised outlier detection. **repeat**

    Update the r_n's with (IV.2). Update the mixture parameters with (IV.3), (IV.4), (IV.5).

  until *The parameters $\boldsymbol{\theta}$ are stable.*

  repeat

    Run SGD to minimize $\mathcal{L}_{\text{DEEPGUM}}$ in (IV.7).

  until *Early stop with a patience of K epochs.*

until $\mathcal{L}_{DEEPGUM}$ *grows on* $\mathcal{V}$.

alternates between responsibility evaluation, (IV.2) and mixture parameter update, (IV.3), (IV.4), (IV.5), and iterates until convergence, namely until the mixture parameters do not evolve anymore. The current mixture parameters are used to evaluate the responsibilities of the validation set. The SGD algorithm takes as input the training and validation sets as well as the associated responsibilities. In order to prevent over-fitting, we perform early stopping on the validation set with a patience of K epochs.

Notice that the training procedure requires neither specific annotation of outliers nor the ratio of outliers present in the data. The procedure is initialized by executing SGD, as just described, with all the samples being supposed to be inliers, i.e. $r_n = 1, \forall n$. Algorithm 3 is stopped when $\mathcal{L}_{\text{DEEPGUM}}$ does not decrease anymore. It is important to notice that we do not need to constrain the model to avoid the trivial solution, namely all the samples are considered as outliers. This is because after the first SGD execution, the network can discriminate between the two categories. In the extreme case when DeepGUM would consider all the samples as outliers, the algorithm would stop after the first SGD run and would output the initial model.

Since EM provides the data covariance matrix $\boldsymbol{\Sigma}$, it may be tempting to use the Mahalanobis norm instead of the L_2 norm in (IV.7). The covariance matrix is narrow along output dimensions with low-amplitude noise and wide along dimensions with high-amplitude noise. The Mahalanobis distance would give equal importance to low- and high-amplitude noise dimensions which is not desired. Another interesting feature of the proposed algorithm is that the posterior r_n weights the learning rate of sample n as its gradient is simply multiplied by r_n. Therefore, the proposed algorithm automatically selects a learning rate for each individual training sample.

IV.4 Experiments

The purpose of the experimental validation is two-fold. First, we empirically validate DeepGUM with three datasets that are naturally corrupted with outliers. The validations are carried out with the following applications: fashion landmark detection (Section IV.4.1), age estimation (Section IV.4.2) and head pose estimation (Section IV.4.3). Second, we delve into the robustness of DeepGUM and analyse its behavior in comparison with existing robust deep regression techniques by corrupting the annotations with an increasing percentage of outliers on the facial landmark detection task (Section IV.4.4).

We systematically compare DeepGUM with the standard L_2 loss, the Huber loss and the Biweight loss (used in [107]). In all these cases, we use the VGG-16 architecture [108] pre-trained on ImageNet [109]. We also tried to use the architecture proposed in [107], but we were unable to reproduce the results reported in [107] on the LSP and Parse datasets, using the code provided by the authors. Therefore, for the sake of reproducibility and for a fair comparison between different robust loss functions, we used VGG-16 in all our experiments. In detail, we fine-tune the last convolutional block and both fully connected layers with a mini-batch of size 128 and learning rate set to 10^{-4}. The fine-tuning starts with 3 epochs of L_2 loss, before exploiting either the Biweight, Huber of DeepGUM loss. When using any of these three losses, the network output is normalized with the median absolute deviation (as in [107]), computed on the entire dataset after each epoch. Early stopping with a patience of $K = 5$ epochs is employed and the data is augmented using mirroring.

In order to evaluate the methods, we report the mean absolute error (MAE) between the regression target and the network output over the test set. In addition, we complete the evaluation with statistical tests that allow to point out when the differences between methods are systematic and statistically significant or due to chance. Statistical tests are run per-image regression errors and therefore can only be applied to the methods for which the code is available, and not to average errors reported in the literature; in the latter case, only MAE are made available. In practice, we use the non-parametric Wilcoxon signed-rank test [110] to assess whether the null hypothesis (the median difference between pairs of observations is zero) is true or false. We denote the statistical significance with *, ** or ***, corresponding to a p-value (the probability of the null hypothesis being true) smaller than $p = 0.05$, $p = 0.01$ or

Table IV.1: *Mean absolute error on the upper-body subset of FLD, per landmark and in average. The landmarks are left (L) and right (R) collar (C), sleeve (S) and hem (H). The results of DFA are from [112] and therefore do not take part in the statistical comparison.*

Method	Upper-body landmarks						
	LC	RC	LS	RS	LH	RH	Avg.
DFA [112] (L_2)	15.90	15.90	30.02	29.12	23.07	22.85	22.85
DFA [112] (5 VGG)	10.75	10.75	20.38	19.93	15.90	16.12	15.23
L_2	12.08	12.08	18.87	18.91	16.47	16.40	15.80
Huber [113]	14.32	13.71	20.85	19.57	20.06	19.99	18.08
Biweight [107]	13.32	13.29	21.88	21.84	18.49	18.44	17.88
DeepGUM	11.97***	11.99***	18.59***	18.50***	16.44***	16.29***	15.63***

Figure IV.3: *Sample fashion landmarks detected by DeepGUM.*

$p = 0.001$, respectively. We only report the statistical significance of the methods with the lowest MAE. For instance, A*** means that *the probability that method* A *is equivalent to any other method is less than* $p = 0.001$.

IV.4.1 Fashion Landmark Detection

Visual fashion analysis presents a wide spectrum of applications such as cloth recognition, retrieval, and recommendation. We employ the fashion landmark dataset (FLD) [111] that includes more than $120K$ images, where each image is labeled with eight landmarks. The dataset is equally divided in three subsets: upper-body clothes (6 landmarks), full-body clothes (8 landmarks) and lower-body clothes (4 landmarks). We randomly split each subset of the dataset into test ($5K$), validation ($5K$) and training ($\sim 30K$). Two metrics are used: the mean absolute error (MAE) of the landmark localization and the percentage of failures (landmarks detected further from the ground truth than a given threshold). We employ *landmark-wise* r_n.

Table IV.1 reports the results obtained on the upper-body subset of the fashion landmark dataset (additional results on full-body and lower-body subsets are included in the supplementary material). We report the mean average error (in pixels) for each landmark individually, and the overall average (last column). While for the first subset we can compare with the very recent results reported in [112], for the other there are no previously reported results. Generally speaking, we outperform all other baselines in average, but also in each of the individual landmarks. The only exception is the comparison against the method utilizing five VGG pipelines to estimate the position of the landmarks. Although this method reports slightly better performance than DeepGUM for some columns of Table IV.1, we recall that we are using one single VGG as front-end, and therefore the representation power cannot be the same as the one associated to a pipeline employing five VGG's trained for tasks such as pose estimation and cloth classification that clearly aid the fashion landmark estimation task.

Interestingly, DeepGUM yields better results than L_2 regression and a major improvement over Biweight [107] and Huber [113]. This behavior is systematic for all fashion landmarks and statistically significant (with $p < 0.001$). In order to better understand this behavior, we computed the percentage of outliers detected by DeepGUM and Biweight, which are 3% and 10% respectively (after convergence). We believe that within this difference (7% corresponds to $2.1K$ images) there are mostly "difficult" inliers, from which the network could learn a lot (and does it in DeepGUM) if they were not discarded as happens with Biweight. This illustrates the importance of rejecting the outliers while keeping the inliers in the learning loop, and exhibits the robustness of DeepGUM in doing so. Figure IV.3 displays a few landmarks estimated by DeepGUM.

IV.4.2 Age Estimation

Age estimation from a single face image is an important task in computer vision with applications in access control and human-computer interaction. This task is closely related to the prediction of other biometric and facial attributes,

Method	MAE
L_2	5.75
Huber [113]	5.59
Biweight [107]	5.55
Dex [86]	5.25
DexGUM***	5.14
DeepGUM***	5.08

Figure IV.4: *Results on the CACD dataset: (left) mean absolute error and (right) images considered as outliers by DeepGUM, the corresponding annotation is (left to right): 14, 14, 14, 16, 20, 23 (top row) and 49, 51, 60, 60, 60, 62 (bottom row).*

such as gender, ethnicity, and hair color. We use the cross-age celebrity dataset (CACD) [114] that contains $163,446$ images from $2,000$ celebrities. The images are collected from search engines using the celebrity's name and desired year (from 2004 to 2013). The dataset splits into 3 parts, $1,800$ celebrities are used for training, 80 for validation and 120 for testing. The validation and test sets are manually cleaned whereas the training set is noisy. In our experiments, we report results using *image-wise* r_n.

Apart from DeepGUM, L_2, Biweight and Huber, we also compare to the age estimation method based on deep expectation (Dex) [86], which was the winner of the Looking at People 2015 challenge. This method uses the VGG-16 architecture and poses the age estimation problem as a classification problem followed by a softmax expected value refinement. We report results with two different approaches using Dex. First, our implementation of the original Dex model. Second, we add the GUM model on top the the Dex architecture; we termed this architecture DexGUM.

The table in Figure IV.4 reports the results obtained on the CACD test set for age estimation. We report the mean absolute error (in years) for size different methods. We can easily observe that DeepGUM exhibits the best results: 5.08 years of MAE (0.7 years better than L_2). Importantly, the architectures using GUM (DeepGUM followed by DexGUM) are the ones offering the best performance. This claim is supported by the results of the statistical tests, which say that DexGUM and DeepGUM are statistically better than the rest (with $p < 0.001$), and that there are no statistical differences between them. This is further supported by the histogram of the error included in the supplementary material. DeepGUM considered that 7% of images were outliers and thus these images were undervalued during training. The images in Figure IV.4 correspond to outliers detected by DeepGUM during training, and illustrate the ability of DeepGUM to detect outliers. Since the dataset was automatically annotated, it is prone to corrupted annotations. Indeed, the age of each celebrity is automatically annotated by subtracting the date of birth from the picture time-stamp. Intuitively, this procedure is problematic since it assumes that the automatically collected and annotated images show the right celebrity and that the times-tamp and date of birth are correct. Our experimental evaluation clearly demonstrates the benefit of a robust regression technique to operate on datasets populated with outliers.

IV.4.3 Head Pose Estimation

The McGill real-world face video dataset [87] consists of 60 videos (a single participant per video, 31 women and 29 men) recorded with the goal of studying unconstrained face classification. The videos were recorded in both indoor and outdoor environments under different illumination conditions and participants move freely. Consequently, some frames suffer from important occlusions. The yaw angle (ranging from $-90°$ to $90°$) is annotated using a two-step labeling procedure that, first, automatically provides the most probable angle as well as a degree of confidence, and then the final label is chosen by a human annotator among the plausible angle values. Since the resulting annotations are not perfect it makes this dataset suitable to benchmark robust regression models. As the training and test sets are not separated in the original dataset, we perform a 7-fold cross-validation. We report the fold-wise MAE average and standard deviation as well as the statistical significance corresponding to the concatenation of the test results of the 7 folds. Importantly, only a subset of the dataset is publicly available (35 videos over 60).

In Table IV.2, we report the results obtained with different methods and employ a dagger to indicate when a particular method uses the entire dataset (60 videos) for training. We can easily notice that DeepGUM exhibits the best results compared to the other ConvNets methods (respectively $0.99°$, $0.50°$ and $0.20°$ lower than L_2, Huber and Biweight in MAE). The last three approaches, all using deep architectures, significantly outperform the current state-of-the-art approach [115]. Among them, DeepGUM is significantly better than the rest with $p < 0.001$.

Table IV.2: *Mean average error on the McGill dataset. The results of the first half of the table are directly taken from the respective papers and therefore no statistical comparison is possible.* †*Uses extra training data.*

Method	MAE	RMSE
Xiong et al. [116]†	-	29.81 ± 7.73
Zhu and Ramanan [54]†	-	35.70 ± 7.48
Demirkus et al. [87]†	-	12.41 ± 1.60
Drouard et al. [115]	12.22 ± 6.42	23.00 ± 9.42
L_2	8.60 ± 1.18	12.03 ± 1.66
Huber [113]	8.11 ± 1.08	11.79 ± 1.59
Biweight [107]	7.81 ± 1.31	11.56 ± 1.95
DeepGUM***	7.61 ± 1.00	11.37 ± 1.34

IV.4.4 Facial Landmark Detection

We perform experiments on the LFW and NET facial landmark detection datasets [88] that consist of 5590 and 7876 face images, respectively. We combined both datasets and employed the same data partition as in [88]. Each face is labeled with the positions of five key-points in Cartesian coordinates, namely left and right eye, nose, and left and right corners of the mouth. The detection error is measured with the Euclidean distance between the estimated and the ground truth position of the landmark, divided by the width of the face image, as in [88]. The performance is measured with the failure rate of each landmark, where errors larger than 5% are counted as failures. The two aforementioned datasets can be considered as outlier-free since the average failure rate reported in the literature falls below 1%. Therefore, we artificially modify the annotations of the datasets for facial landmark detection to find the breakdown point of DeepGUM. Our purpose is to study the robustness of the proposed deep mixture model to outliers generated in controlled conditions. We use three different types of outliers:

- Normally Generated Outliers (*NGO*): A percentage of landmarks is selected, regardless of whether they belong to the same image or not, and shifted a distance of d pixels in a uniformly chosen random direction. The distance d follows a Gaussian distribution, $\mathcal{N}(25, 2)$. *NGO* simulates errors produced by human annotators that made a mistake when clicking, thus annotating in a slightly wrong location.
- Local - Uniformly Generated Outliers (*l-UGO*): It follows the same philosophy as *NGO*, sampling the distance d from a uniform distribution over the image, instead of a Gaussian. Such errors simulate human errors that are not related to the human precision, such as not selecting the point or misunderstanding the image.
- Global - Uniformly Generated Outliers (*g-UGO*): As in the previous case, the landmarks are corrupted with uniform noise. However, in *g-UGO* the landmarks to be corrupted are grouped by image. In other words, we do not corrupt a subset of all landmarks regardless of the image they belong to, but rather corrupt all landmarks of a subset of the images. This strategy simulates problems with the annotation files or in the sensors in case of automatic annotation.

The first and the second types of outlier contamination employ *landmark-wise* r_n, while the third uses *image-wise* r_n.

the plots in Figure IV.5 report the failure rate of DeepGUM, Biweight, Huber and L_2 (top) and the outlier detection precision and recall of all except for L_2 (bottom) for the three types of synthetic noise. The precision corresponds to the percentage of training samples classified as outliers that are true outliers; and the recall corresponds to the percentage of outliers that are classified as such. The first conclusion that can be drawn directly from this figure are that, on the one hand, Biweight and Huber systematically present a lower recall than DeepGUM. In other words, DeepGUM exhibits the highest reliability at identifying and, therefore, ignoring outliers during training. And, on the other hand, DeepGUM tends to present a lower failure rate than Biweight, Huber and L_2 in most of the scenarios contemplated.

Regarding the four most-left plots, *l-UGO* and *g-UGO*, we can clearly observe that, while for limited amounts of outliers (i.e. $< 10\%$) all methods report comparable performance, DeepGUM is clearly superior to L_2, Biweight and Huber for larger amounts of outliers. We can also safely identify a breakdown point of DeepGUM on *l-UGO* at $\sim 40\%$. This is inline with the reported precision and recall for the outlier detection task. While for Biweight and Huber, both decrease when increasing the number of outliers, these measures are constantly around 99% for DeepGUM (before 40% for *l-UGO*). The fact that the breakdown point of DeepGUM under *g-UGO* is higher than 50% is due to fact that the a priori model of the outliers (i.e. uniform distribution) corresponds to the way the data is corrupted.

For *NGO*, the corrupted annotation is always around the ground truth, leading to a failure rate smaller than 7% for all methods. We can see that all four methods exhibit comparable performance up to 30% of outliers. Beyond

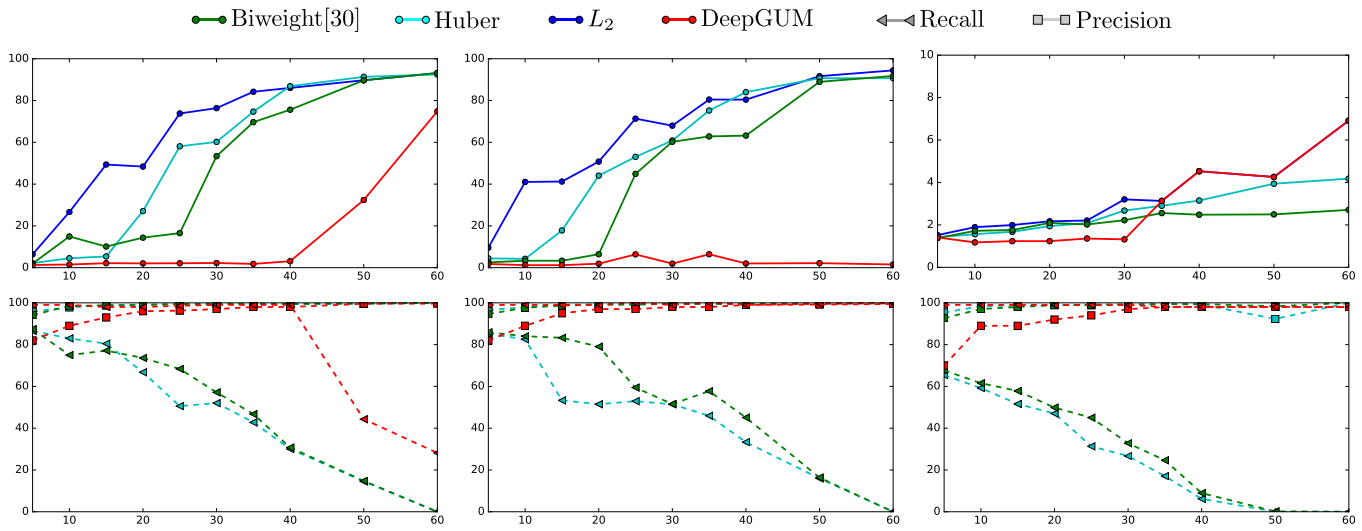


Figure IV.5: *Evolution of the failure rate (top) when augmenting the noise for the 3 types of outliers considered (from left to right: l-UGO, g-UGO and NGO). We also display the corresponding precisions and recalls in percentage (bottom) for the outlier class. Best seen in color.*

that threshold, Biweight outperforms the other methods in spite of presenting a progressively lower recall and a high precision (i.e. Biweight identifies very few outliers, but the ones identified are true outliers). This behavior is also exhibited by Huber. Regarding DeepGUM, we observe that in this particular setting the results are aligned with L_2. This is because the SGD procedure is not able to find a better optimum after the first epoch and therefore the early stopping mechanism is triggeres and SFD output the initial network, which corresponds to L_2. We can conclude that the strategy of DeepGUM, consisting in removing all points detected as outliers, is not effective in this particular experiment. In other words, having more noisy data is better than having only few clean data in this particular case of 0-mean highly correlated noise. Nevertheless, we consider an attractive property of DeepGUM the fact that it can automatically identify these particular cases and return an acceptable solution.

IV.5 Conclusions

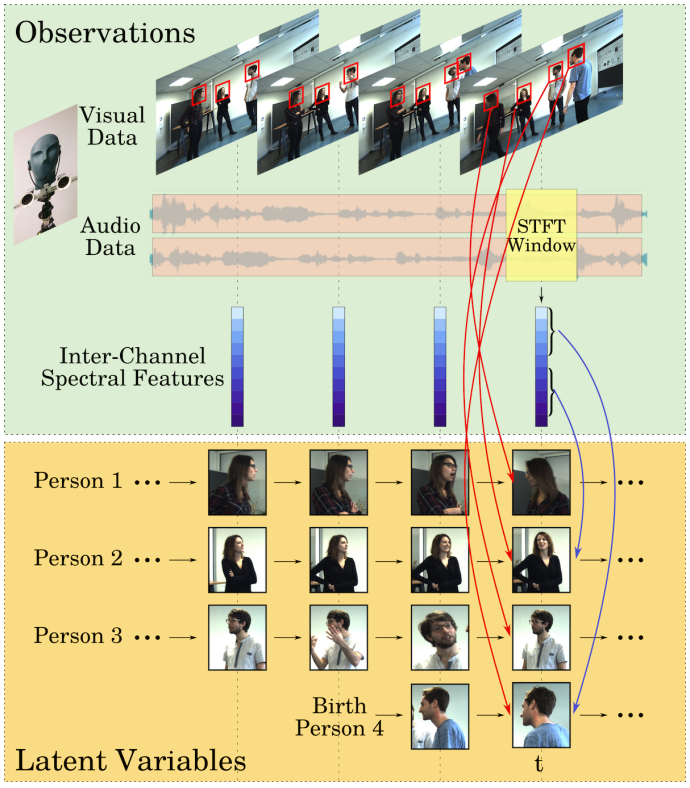
This chapter introduced a deep robust regression learning method that uses a Gaussian-uniform mixture model. The novelty resides in combining a probabilistic robust mixture model with deep learning in a jointly trainable fashion. In this context, previous studies only dealt with the classical L_2 loss function or Tukey's Biweight function, an M-estimator robust to outliers [107]. Our proposal yields better performance than previous deep regression approaches by proposing a novel technique, and the derived optimization procedure, that alternates between the unsupervised task of outlier detection and the supervised task of learning network parameters. The experimental validation addresses four different tasks: facial and fashion landmark detection, age estimation, and head pose estimation. We have empirically shown that DeepGUM (i) is a robust deep regression approach that does not need to rigidly specify *a priori* the distribution (number and spread) of outliers, (ii) exhibits a higher breakdown point than existing methods when the outliers are sampled from a uniform distribution (being able to deal with more than 50% of outlier contamination without providing incorrect results), and (iii) is capable of providing comparable or better results than current state-of-the-art approaches in the four aforementioned tasks. Finally, DeepGUM could be easily used to remove undesired samples that arise from tedious manual annotation. It could also deal with highly unusual training samples inherently present in automatically collected huge datasets, a problem that is currently addressed using error-prone and time-consuming human supervision.

Part B

Computationally Intractable $p(\mathbf{x})$

Chapter *V*

Variational Expectation-Maximisation for Audio-Visual Multi-Speaker Tracking



Abstract *In this chapter we address the problem of tracking multiple speakers via the fusion of visual and auditory information. We propose to exploit the complementary nature and roles of these two modalities in order to accurately estimate smooth trajectories of the tracked persons, to deal with the partial or total absence of one of the modalities over short periods of time, and to estimate the acoustic status – either speaking or silent – of each tracked person over time. We propose to cast the problem at hand into a generative audio-visual fusion (or association) model formulated as a latent-variable temporal graphical model. This may well be viewed as the problem of maximizing the posterior joint distribution of a set of continuous and discrete latent variables given the past and current observations, which is intractable. We propose a variational inference model which amounts approximating the joint distribution with a factorized distribution. The solution takes the form of a closed-form expectation maximization procedure. We describe in detail the inference algorithm, we evaluate its performance and we compare it with several baseline methods. These experiments show that the proposed audio-visual tracker performs well in informal meetings involving a time-varying number of people.*

Chapter Pitch

Methodological contribution An audio-visual multi-observation multi-source linear dynamical system is proposed, and a variational EM is derived to exploit this model in a computationally efficient manner.

Applicative task Audio-visual multi-person tracking.

Interesting insight Thanks to the variational approximation, the inference is performed by alternating N – the number of sources – Kalman filters/smoothers with T – the number of frames – GMM E-steps. In short, alternating tracking and clustering.

Dissemination The multi-source AV tracker was published in IEEE Transactions on Pattern Analysis and Machine Intelligence [117].

V.1 Introduction

We address the problem of tracking multiple speakers via the fusion of visual and auditory information [118]–[124]. We propose to exploit the complementary nature of these two modalities in order to accurately estimate the position of each person at each time step, to deal with the partial or total absence of one of the modalities over short periods of time, and to estimate the acoustic status, either speaking or silent, of each tracked person. We propose to cast the problem at hand into a generative audio-visual fusion (or association) model formulated as a latent-variable temporal graphical model. We propose a tractable solver via a variational approximation.

We are particularly interested in tracking people involved in informal meetings and social gatherings.In this type of scenarios, participants wander around, cross each other, move in and out the camera field of view, take speech turns, etc. Acoustic room conditions, e.g. reverberation, and overlapping audio sources of various kinds drastically deteriorate or modify the microphone signals. Likewise, occluded persons, lighting conditions and mid-range camera distance complicate the task of visual processing. It is therefore impossible to gather reliable and continuous flows of visual **and** audio observations. Hence one must design a fusion and tracking method that is able to deal with intermittent visual and audio data.

We propose a multi-speaker tracking method based on a dynamic Bayesian model that fuses audio and visual information over time from their respective observation spaces. This may well be viewed as a generalization of single-observation and single-target Kalman filtering – which yields an exact recursive solution – to multiple observations and multiple targets, which makes the exact recursive solution computationally intractable. We propose a variational approximation of the joint posterior distribution over the continuous variables (positions and velocities of tracked persons) and discrete variables (observation-to-person associations) at each time step, given all the past and present audio and visual observations. The proposed approximation consists on factorizing the joint distribution. We obtain a variational expectation maximisation (VEM) algorithm that is not only computationally tractable, but also very efficient.

In general, multiple object tracking consists of the temporal estimation of the kinematic state of each object, i.e. position and velocity. In computer vision, local descriptors are used to better discriminate between objects, e.g. person detectors/descriptors based on hand-crafted features [125] or on deep neural networks [126]. If the tracked objects emit sounds, their states can be inferred as well using sound-source localization techniques combined with tracking, e.g. [127]. These techniques are often based on the estimation of the sound's direction of arrival (DOA) using a microphone array, e.g. [128], or on a steered beamformer [127]. DOA estimation can be carried out either in the temporal domain [129], or in the spectral (Fourier) domain [130]. However, spectral-domain DOA estimation methods are more robust than temporal-domain methods, in particular in the presence of background noise and reverberation [131], [132]. The multiple sound-source localization and tracking method of [127] combines a steered beamformer with a particle filter. The loudest sound source is detected first, the second loudest one is next detected, etc., and up to four sources. This leads to many false detections. Particle filtering is combined with source-to-track assignment probabilities in order to determine whether a newly detected source is a false detection, a source that is currently being tracked, or a new source. In practice, this method requires several empirically defined thresholds.

Via proper camera-microphone calibration, audio and visual observations can be aligned such that a DOA corresponds to a 2D location in the image plane. We adopt the audio-visual alignment method of [133], which learns a mapping from the space spanned by *inter-channel spectral features* (audio features) to the space of source locations, which in our case corresponds to the image plane. Interestingly, the method of [133] estimates both this mapping and its inverse via a closed-form EM algorithm. Moreover, this allows us to exploit the richness of representing acoustic signals in the short-time Fourier domain [134] and to extract noise- and reverberation-free audio features [131].

We propose to represent the audio-visual fusion problem via two sets of independent variables, i.e. visual-feature-to-person and audio-feature-to-person sets of assignment variables. An interesting characteristic of this way of doing is that the proposed tracking algorithm can choose to use visual features, audio features, or a combination of both, and this choice can be made independently for every person and for every time step. Indeed, audio and visual information are rarely available simultaneously and continuously. Visual information suffers from limited camera field-of-view, occlusions, false positives, missed detections, etc. Audio information is often corrupted by room acoustics, environmental noise and overlapping acoustic signals. In particular speech signals are sparse, non-stationary and are emitted intermittently, with silence intervals between speech utterances. Hence a robust audio-visual tracking must explicitly take into account the temporal sparsity of the two modalities and this is exactly what we propose.

We use the AV16.3 [135] and the AVDIAR [136] datasets to evaluate the performance of the proposed audio-visual tracker. We use the Multiple Object Tracking (MOT) metrics and the Optimal Sub Pattern Assignment fo Tracks (OSPA-T) metrics to quantitatively assess method performance. MOT and in particular MOTA (tracking accuracy), which combines false positives, false negatives, identity switches, by comparing the estimated tracks with the ground-truth trajectories, is a commonly used score to assess the quality of a multiple person tracker.[V.1] OSPA-T measures the

[V.1] `https://motchallenge.net/`

distance between two point sets and hence it is also useful to compare ground-truth tracks with estimated tracks in the context of multi-target tracking [137]. We use MOT and OSPA-T metrics to compare our method with two recently proposed audio-visual tracking methods [121], [124] and with a visual tracker [125]. An interesting outcome of the proposed method is that speaker diarization, i.e. who speaks when, can be coarsely inferred from the tracking output, thanks to the audio-feature-to-person assignment variables. The speaker diarization results obtained with our method are compared with two other methods [136], [138] based on the Diarization Error Rate (DER) score.

The remainder of the chapter is organized as follows. Section V.2 describes the related work. Section V.3 describes in detail the proposed formulation. Section V.4 describes the proposed variational approximation and Section V.5 details the variational expectation-maximization procedure. The algorithm implementation is described in Section V.6. Tracking results and comparisons with other methods are reported in Section V.7. Finally, Section V.8 draws a few conclusions.[V.2]

V.2 Related Work

In computer vision, there is a long history of multiple object tracking methods. While these methods provide interesting insights concerning the problem at hand, a detailed account of existing visual trackers is beyond our scope. Several audio-visual tracking methods were proposed in the recent past, e.g. [118]–[120], [139]. These papers proposed to use approximate inference of the filtering distribution using Markov chain Monte Carlo particle filter sampling (MCMC-PF). These methods cannot provide estimates of the accuracy and merit of each modality with respect to each tracked person.

More recently, audio-visual trackers based on particle filtering and probability hypothesis density (PHD) filters were proposed, e.g. [121]–[124], [140]–[142]. [123] used DOAs of audio sources to guide the propagation of particles, and combined the filter with a mean-shift algorithm to reduce the computational complexity. Some PHD filter variants were proposed to improve the tracking performance [140], [141]. The method of [121] also used DOAs of active audio sources to give more importance to particles located around DOAs. Along the same line of thought, [124] proposed a mean-shift sequential Monte Carlo PHD (SMC-PHD) algorithm that used audio information to improve the performance of a visual tracker. This implies that the persons being tracked must emit acoustic signals continuously and that multiple-source audio localization is reliable enough for proper audio-visual alignment.

PHD-based tracking methods are computationally efficient but their inherent limitation is that they are unable to associate observations to tracks. Hence they require an external post-processing mechanism that provides associations. Also, in the case of PF-based audio-visual filtering, the number of tracked persons must be set in advance and sampling can be a computational burden. In contrast, the proposed variational formulation embeds association variables within the model, uses a birth process to estimate the initial number of persons and to add new ones along time, and an explicit dynamic model yields smooth trajectories.

Another limitation of the methods proposed in [118], [120], [123], [140]–[142] is that they need as input a continuous flow of audio and visual observations. To some extent, this is also the case with [121], [124], where only the audio observations are supposed to be continuous. All these methods showed good performance in the case of the AV16.3 dataset [135] in which the participants spoke simultaneously and continuously – which is somehow artificial. The AV16.3 dataset was recorded in a specially equipped meeting room using three cameras that generally guarantee that frontal views of the participants were always available. This contrasts with the AVDIAR dataset which was recorded with one sensor unit composed of two cameras and six microphones. The AVDIAR scenarios are composed of participants that take speech turns while they look at each other, hence they speak intermittently and they do not always face the cameras.

Recently, we proposed an audio-visual clustering method [17] and an audio-visual speaker diarization method [136]. The weighted-data clustering method of [17] analyzed a short time window composed of several audio and visual frames and hence it was assumed that the speakers were static within such temporal windows. Binaural audio features were mapped onto the image plane and were clustered with nearby visual features. There was no dynamic model that allowed to track speakers. The audio-visual diarization method [136] used an external multi-object visual tracker that provided trajectories for each tracked person. The audio-feature-space to image-plane mapping [133] was used to assign audio information to each tracked person at each time step. Diarization itself was modeled with a binary state variable (speaking or silent) associated with each person. The diarization transition probabilities (state dynamics) were hand crafted, with the assumption that the speaking status of a person was independent of all the other persons. Because of the small number of state configurations, i.e. $\{0,1\}^N$ (where N is the maximum number of tracked persons), the MAP solution could be found by exhaustively searching the state space. In Section V.7.2 we use the AVDIAR recordings to compare our diarization results with the results obtained with [136].

[V.2] Supplemental materials are available at `https://team.inria.fr/perception/research/var-av-track/`

The variational Bayesian inference method proposed we propose may well be viewed as a multimodal generalization of variational expectation maximization algorithms for multiple object tracking using either visual-only information [125] or audio-only information [143], [144]. We show that these models can be extended to deal with observations living in completely different mathematical spaces. Indeed, we show that two (or several) different data-processing pipelines can be embedded and treated on an equal footing in the proposed formulation. Special attention is given to audio-visual alignment and to audio-to-person assignments: (i) we learn a mapping from the space of audio features to the image plane, as well as the inverse of this mapping, which are integrated in the proposed generative approach, and (ii) we show that the an increase in the number of assignment variables, due to the use of two modalities, do not affect the complexity of the algorithm. Absence of observed data of any kind or erroneous data are carefully modeled: this enables the algorithm to deal with intermittent observations, whether audio, visual, or both. This is probably one of the most prominent features of the method, in contrast with most existing audio-visual tracking methods which require continuous and simultaneous flows of visual and audio data. Moreover, we show that our tracker can be used for audio-visual speaker diarization [136].

V.3 Proposed Model

V.3.1 Mathematical Definitions and Notations

Unless otherwise specified, uppercase letters denote random variables while lowercase letters denote their realizations, *e.g.* $p(X = x)$, where $p(\cdot)$ denotes either a probability density function (pdf) or a probability mass function (pmf). For the sake of conciseness we generally write $p(x)$. Vectors are written in slanted bold, *e.g.* $\boldsymbol{X}, \boldsymbol{x}$, whereas matrices are written in bold, *e.g.* $\mathbf{Y}, \mathbf{y}$. Video and audio data are assumed to be synchronized, and let t denote the common frame index. Let N be the upper bound of the number of persons that can simultaneously be tracked at any time t, and let $n \in \{1 \dots N\}$ be the person index. Let $n = 0$ denote *nobody*. A t subscript denotes variable concatenation at time t, e.g. $\mathbf{X}_t = (\boldsymbol{X}_{t1}, \dots, \boldsymbol{X}_{tn}, \dots, \boldsymbol{X}_{tN})$, and the subscript $1:t$ denotes concatenation from 1 to t, e.g. $\mathbf{X}_{1:t} = (\mathbf{X}_1, \dots, \mathbf{X}_t)$.

Let $\boldsymbol{X}_{tn} \in \mathcal{X} \subset \mathbb{R}^2$, $\boldsymbol{Y}_{tn} \in \mathcal{Y} \subset \mathbb{R}^2$ and $\boldsymbol{W}_{tn} \in \mathcal{W} \subset \mathbb{R}^2$ be three latent variables that correspond to the 2D position, 2D velocity and 2D size (width and height) of person n at t, respectively. Typically, $\boldsymbol{X}_{tn}$ and $\boldsymbol{W}_{tn}$ are the center and size of a bounding box of a person while $\boldsymbol{Y}_{tn}$ is the velocity of $\boldsymbol{X}_{tn}$. Let $\boldsymbol{S}_t = \{(\boldsymbol{X}_{tn}^\top, \boldsymbol{W}_{tn}^\top, \boldsymbol{Y}_{tn}^\top)^\top\}_{n=1}^N \subset \mathbb{R}^6$ be the complete set of continuous latent variables at t, where $^\top$ denotes the transpose operator. Without loss of generality, we assume that a person is characterized with the bounding box of her/his head and the center of this bounding box is assumed to be the location of the corresponding speech source.

We now define the observations.

At each time t there are M_t visual observations and K_t audio observations. Let $\mathbf{f}_t = \{\boldsymbol{f}_{tm}\}_{m=1}^{M_t}$ and $\mathbf{g}_t = \{\boldsymbol{g}_{tk}\}_{k=1}^{K_t}$ be realizations of the visual and audio observed random variables $\{\boldsymbol{F}_{tm}\}_{m=1}^{M_t}$ and $\{\boldsymbol{G}_{tk}\}_{k=1}^{K_t}$, respectively. Visual observations, $\boldsymbol{f}_{tm} = (\boldsymbol{v}_{tm}^\top, \boldsymbol{u}_{tm}^\top)^\top$, correspond to the bounding boxes of detected faces, namely the concatenation of the bounding-box center, width and height, $\boldsymbol{v}_{tm} \in \mathcal{V} \subset \mathbb{R}^4$, and of a feature vector $\boldsymbol{u}_{tm} \in \mathcal{H} \subset \mathbb{R}^d$ that describes the photometric content of that bounding box, i.e. a d-dimensional face descriptor. Audio observations, $\boldsymbol{g}_{tk}$, correspond to inter-channel spectral features, where k is a frequency sub-band index. Let's assume that there are K sub-bands, that $K_t \leq K$ sub-bands are *active* at t, i.e. sub-bands with sufficient signal energy, and that there are J frequencies per sub-band. Hence, $\boldsymbol{g}_{tk} \in \mathbb{R}^{2J}$ corresponds to the real and imaginary parts of J complex-valued Fourier coefficients. It is well established that inter-channel spectral features $\{\boldsymbol{g}_{tk}\}_{k=1}^{K_t}$ contain audio-source localization information, which is what is needed for tracking. These audio features are obtained by applying the multi-channel audio processing method described below. Note that both the number of visual and of audio observations at t, M_t and K_t, vary over time. Let $\mathbf{o}_{1:t} = (\mathbf{o}_1, \dots, \mathbf{o}_t)$ denote the set of observations from 1 to t, where $\mathbf{o}_t = (\mathbf{f}_t, \mathbf{g}_t)$.

Finally, we define the assignment variables of the proposed latent variable model. There is an assignment variable (a discrete random variable) associated with each observed variable. Namely, let A_{tm} and B_{tk} be associated with $\boldsymbol{f}_{tm}$ and with $\boldsymbol{g}_{tk}$, respectively, e.g. $p(A_{tm} = n)$ denotes the probability of assigning visual observation m at t to person n. Note that $p(A_{tm} = 0)$ and $p(B_{tk} = 0)$ are the probabilities of assigning visual observation m and audio observation k to none of the persons, or to nobody. In the visual domain, this may correspond to a false detection while in the audio domain this may correspond to an audio signal that is not uttered by a person. There is an additional assignment variable, C_{tk} that is associated with the audio generative model described in Section V.3.4. The assignment variables are jointly denoted with $\boldsymbol{Z}_t = (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{C}_t)$.

V.3.2 The Filtering Distribution

We remind that the objective is to estimate the positions and velocities of participants (multiple person tracking) and, possibly, to estimate their speaking status (speaker diarization). The audio-visual multiple-person tracking problem

is cast into the problems of estimating the filtering distribution $p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})$ and of inferring the state variable $\mathbf{S}_t$. Subsequently, speaker diarization can be obtained from audio-feature-to-person information via the estimation of the assignment variables $\boldsymbol{B}_{tk}$ (Section V.6.3).

We reasonably assume that the state variable $\mathbf{S}_t$ follows a first-order Markov model, and that the visual and audio observations only depend on $\mathbf{S}_t$ and $\boldsymbol{Z}_t$. By applying Bayes rule, one can then write the filtering distribution of $(\mathbf{s}_t, \boldsymbol{z}_t)$ as:

$$p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t}) \propto p(\mathbf{o}_t|\mathbf{s}_t, \boldsymbol{z}_t)p(\boldsymbol{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{o}_{1:t-1}), \tag{V.1}$$

with:

$$p(\mathbf{o}_t|\mathbf{s}_t, \boldsymbol{z}_t) = p(\mathbf{f}_t|\mathbf{s}_t, \boldsymbol{a}_t)p(\mathbf{g}_t|\mathbf{s}_t, \boldsymbol{b}_t, \boldsymbol{c}_t), \tag{V.2}$$

$$p(\boldsymbol{z}_t|\mathbf{s}_t) = p(\boldsymbol{a}_t)p(\boldsymbol{b}_t)p(\boldsymbol{c}_t|\boldsymbol{s}_t, \boldsymbol{b}_t), \tag{V.3}$$

$$p(\mathbf{s}_t|\mathbf{o}_{1:t-1}) = \int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{o}_{1:t-1})d\mathbf{s}_{t-1}. \tag{V.4}$$

Eq. (V.2) is the joint (audio-visual) observed-data likelihood. Visual and audio observations are assumed independent conditionally to $\mathbf{S}_t$, and their distributions will be detailed in Sections V.3.3 and V.3.4, respectively.[V.3] Eq. (V.3) is the prior distribution of the assignment variable. The observation-to-person assignments are assumed to be a priori independent so that the probabilities in (V.3) factorize as:

$$p(\boldsymbol{a}_t) = \prod_{m=1}^{M_t} p(a_{tm}), \tag{V.5}$$

$$p(\boldsymbol{b}_t) = \prod_{k=1}^{K_t} p(b_{tk}), \tag{V.6}$$

$$p(\boldsymbol{c}_t|\mathbf{s}_t, \boldsymbol{b}_t) = \prod_{k=1}^{K_t} p(c_{tk}|\boldsymbol{s}_{tn}, B_{tk} = n). \tag{V.7}$$

It makes sense to assume that these distributions do not depend on t and that they are uniform. The following notations are introduced: $\eta_{mn} = p(A_{tm} = n) = 1/(N+1)$ and $\rho_{kn} = p(B_{tk} = n) = 1/(N+1)$. The probability $p(c_{tk}|\boldsymbol{s}_{tn}, B_{tk} = n)$ is discussed below (Section V.3.4).

Eq. (V.4) is the predictive distribution of $\mathbf{s}_t$ given the past observations, i.e. from 1 to $t-1$. The state dynamics in (V.4) are modeled with a linear-Gaussian first-order Markov process. Moreover, it is assumed that the dynamics are independent over speakers:

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{s}_{tn}; \mathbf{D}\boldsymbol{s}_{t-1\,n}, \boldsymbol{\Lambda}_{tn}), \tag{V.8}$$

where $\boldsymbol{\Lambda}_{tn}$ is the dynamics' covariance matrix and $\mathbf{D}$ is the state transition matrix, given by:

$$\mathbf{D} = \begin{pmatrix} \mathbf{I}_{4\times4} & \mathbf{I}_{2\times2} \\ \mathbf{0}_{2\times4} & \mathbf{I}_{2\times2} \end{pmatrix}.$$

As described in Section V.4 below, an important feature of the proposed model is that the predictive distribution (V.4) at frame t is computed from the state dynamics model (V.8) and an approximation of the filtering distribution $p(\mathbf{s}_{t-1}|\mathbf{o}_{1:t-1})$ at frame $t-1$, which also factorizes across speaker. As a result, the computation of (V.4) factorizes across speakers as well.

V.3.3 The Visual Observation Model

As already mentioned above (Section V.3.1), a visual observation $\boldsymbol{f}_{tm}$ consists of the center, width and height of a bounding box, namely $\boldsymbol{v}_{tm} \in \mathcal{V} \subset \mathbb{R}^4$, as well as of a feature vector $\boldsymbol{u}_{tm} \in \mathcal{H} \subset \mathbb{R}^d$ describing the region inside the bounding box. Since the velocity is not observed, a 4×6 projection matrix $\mathbf{P}_f = (\mathbf{I}_{4\times4}\ \mathbf{0}_{4\times2})$ is used to project $\boldsymbol{s}_{tn}$ onto $\mathcal{V}$. Assuming that the M_t visual observations $\{\boldsymbol{f}_{tm}\}_{m=1}^{M_t}$ available at t are independent, and that the appearance

[V.3] We will see that $\mathbf{G}_t$ depends on $\mathbf{X}_t$ but depends neither on $\mathbf{W}_t$ nor on $\mathbf{Y}_t$, and $\mathbf{F}_t$ depends on $\mathbf{X}_t$ and $\mathbf{W}_t$ but not on $\mathbf{Y}_t$.

representation of a person is independent of his/her position in the image, e.g. CNN-based embedding, the visual likelihood in (V.2) is defined as:

$$p(\mathbf{f}_t|\mathbf{s}_t, \boldsymbol{a}_t) = \prod_{m=1}^{M_t} p(\boldsymbol{v}_{tm}|\mathbf{s}_t, a_{tm}) p(\boldsymbol{u}_{tm}|\mathbf{h}, a_{tm}), \tag{V.9}$$

where the observed bounding-box centers, widths, heights, and feature vectors are drawn from the following distributions:

$$p(\boldsymbol{v}_{tm}|\mathbf{s}_t, A_{tm} = n) = \begin{cases} \mathcal{N}(\boldsymbol{v}_{tm}; \mathbf{P}_f \boldsymbol{s}_{tn}, \boldsymbol{\Phi}_{tm}) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{v}_{tm}; \mathrm{vol}(\mathcal{V})) & \text{if } n = 0, \end{cases} \tag{V.10}$$

$$p(\boldsymbol{u}_{tm}|\mathbf{h}, A_{tm} = n) = \begin{cases} \mathcal{B}(\boldsymbol{u}_{tm}; \boldsymbol{h}_n) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{u}_{tm}; \mathrm{vol}(\mathcal{H})) & \text{if } n = 0, \end{cases} \tag{V.11}$$

where $\boldsymbol{\Phi}_{tm} \in \mathbb{R}^{4\times 4}$ is a covariance matrix quantifying the measurement error in the bounding-box center and size, $\mathcal{U}(\cdot; \mathrm{vol}(\cdot))$ is the uniform distribution with $\mathrm{vol}(\cdot)$ being the volume of the support of the variable, $\mathcal{B}(\cdot; \boldsymbol{h})$ is the Bhattacharya distribution [145], and $\mathbf{h} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N) \in \mathbb{R}^{d\times N}$ is a set of prototype feature vectors that model the appearances of the N persons.

V.3.4 The Audio Observation Model

It is well established in the recent audio signal processing literature that inter-channel spectral features encode sound-source localization information [130], [131], [133]. Therefore, observed audio features, $\mathbf{g}_t = \{\boldsymbol{g}_{tk}\}_{k=1}^{K_t}$ are obtained by considering all the pairs of a microphone array. Audio observations depend neither on the size of the bounding box $\boldsymbol{w}_t$, nor on the velocity $\boldsymbol{y}_t$. Indeed, we note that the velocity of a sound source (a moving person) is of about 1 meter/second, which is negligible compared to the speed of sound. Moreover, the inter-microphone distance is small compared to the source-to-microphone distance, hence the Doppler effect, if any, is similar across microphones. Hence one can replace $\boldsymbol{s}$ with $\boldsymbol{x} = \mathbf{P}_g \boldsymbol{s}$ in the equations below, with $\mathbf{P}_g = (\mathbf{I}_{2\times 2}\ \mathbf{0}_{2\times 4})$. By assuming independence across frequency sub-bands (indexed by k), the audio likelihood in (V.2) can be factorized as:

$$p(\mathbf{g}_t|\mathbf{s}_t, \boldsymbol{b}_t, \boldsymbol{c}_t) = \prod_{k=1}^{K_t} p(\boldsymbol{g}_{tk}|\boldsymbol{x}_{tb_{tk}}, b_{tk}, c_{tk}). \tag{V.12}$$

While the inter-channel spectral features $\boldsymbol{g}_{tk}$ contain localization information, in complex acoustic environments there is no explicit transformation that maps a source location onto an inter-channel spectral feature. We therefore make recourse to modeling this mapping via learning a non-linear regression. We use the method of [131] to extract audio features and the piecewise-linear regression model of [133] to learn a mapping between the space of audio-source locations and the space of audio features. The method of [133] belongs to the mixture of experts (MOE) class of models and hence it embeds well in our latent-variable mixture model. Let $\{h_{kr}\}_{r=1}^{r=R}$ be a set of linear regressions, such that the r-th linear transformation h_{kr} maps $\boldsymbol{x} \in \mathbb{R}^2$ onto $\boldsymbol{g}_k \in \mathbb{R}^{2J}$ for the frequency sub-band k. It follows that (V.12) writes:

$$p(\boldsymbol{g}_{tk}|\boldsymbol{x}_{tn}, B_{tk} = n, C_{tk} = r) = \begin{cases} \mathcal{N}(\boldsymbol{g}_{tk}; h_{kr}(\boldsymbol{x}_{tn}), \boldsymbol{\Sigma}_{kr}) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{g}_{tk}; \mathrm{vol}(\mathcal{G})) & \text{if } n = 0, \end{cases} \tag{V.13}$$

where $\boldsymbol{\Sigma}_{kr} \in \mathbb{R}^{2J\times 2J}$ is a covariance matrix that captures the linear-mapping error and C_{tk} is a discrete random variable, such that $C_{tk} = r$ means that the audio feature $\boldsymbol{g}_{tk}$ is generated through the r-th linear transformation.

Please consult Appendix V.9 for details on how the parameters of the linear transformations h_{kr} are learned from a training dataset.

V.4 Variational Approximation

Direct estimation of the filtering distribution $p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})$ is computationally intractable. Consequently, evaluating expectations over this distribution is intractable as well. We overcome this problem via variational inference and associated EM closed-form solver [4], [7]. More precisely $p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})$ is approximated with the following factorized form:

$$p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t}) \approx q(\mathbf{s}_t, \boldsymbol{z}_t) = q(\mathbf{s}_t) q(\boldsymbol{z}_t), \tag{V.14}$$

which implies

$$q(\mathbf{s}_t) = \prod_{n=1}^{N} q(\boldsymbol{s}_{tn}),\ q(\boldsymbol{z}_t) = \prod_{m=1}^{M_t} q(a_{tm}) \prod_{k=1}^{K} q(b_{tk}, c_{tk}), \tag{V.15}$$

where $q(A_{tm} = n)$ and $q(B_{tk} = n, C_{tk} = r)$ are the variational posterior probabilities of assigning visual observation m to person n and audio observation k to person n, respectively. The proposed variational approximation (V.14) amounts to break the conditional dependence of $\mathbf{S}$ and $\boldsymbol{Z}$ with respect to $\mathbf{o}_{1:t}$ which causes the computational intractability. Note that the visual, $\boldsymbol{A}_t$, and audio, $\boldsymbol{B}_t, \boldsymbol{C}_t$, assignment variables are independent, that the assignment variables for each observation are also independent, and that B_{tk} and C_{tk} are conditionally dependent on the audio observation. This factorized approximation makes the calculation of $p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})$ tractable. The optimal solution is given by an instance of the variational expectation maximization (VEM) algorithm [4], [7], which alternates between two steps:

- *Variational* E-step: the approximate log-posterior distribution of each one of the latent variables is estimated by taking the expectation of the complete-data log-likelihood over the remaining latent variables, i.e. (V.16), (V.17), and (V.18) below, and
- M-step: model parameters are estimated by maximizing the variational expected complete-data log-likelihood.[V.4]

In the case of the proposed model the latent variable log-posteriors write:

$$\log q(\boldsymbol{s}_{tn}) = \mathbb{E}_{q(\boldsymbol{z}_t) \prod_{\ell \neq n} q(\boldsymbol{s}_{t\ell})}[\log p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})] + \text{const}, \tag{V.16}$$

$$\log q(a_{tm}) = \mathbb{E}_{q(\mathbf{s}_t) \prod_{\ell \neq m} q(a_{t\ell}) \prod_k q(b_{tk}, c_{tk})}[\log p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})] + \text{const}, \tag{V.17}$$

$$\log q(b_{tk}, c_{tk}) = \mathbb{E}_{q(\mathbf{s}_t) \prod_m q(a_{tm}) \prod_{\ell \neq k} q(b_{t\ell}, c_{t\ell})}[\log p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})] + \text{const}. \tag{V.18}$$

A remarkable consequence of the factorization (V.14) is that $p(\mathbf{s}_{t-1}|\mathbf{o}_{1:t-1})$ is replaced with $q(\mathbf{s}_{t-1}) = \prod_{n=1}^{N} q(\boldsymbol{s}_{t-1\,n})$, consequently (V.4) becomes:

$$p(\mathbf{s}_t|\mathbf{o}_{1:t-1}) \approx \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) \prod_{n=1}^{N} q(\boldsymbol{s}_{t-1\,n}) d\mathbf{s}_{t-1}. \tag{V.19}$$

It is now assumed that the variational posterior distribution $q(\boldsymbol{s}_{t-1\,n})$ is Gaussian with mean $\boldsymbol{\mu}_{t-1\,n}$ and covariance $\boldsymbol{\Gamma}_{t-1\,n}$:

$$q(\boldsymbol{s}_{t-1\,n}) = \mathcal{N}(\boldsymbol{s}_{t-1\,n}; \boldsymbol{\mu}_{t-1\,n}, \boldsymbol{\Gamma}_{t-1\,n}). \tag{V.20}$$

By substituting (V.20) into (V.19) and combining it with (V.8), the predictive distribution (V.19) becomes:

$$p(\mathbf{s}_t|\mathbf{o}_{1:t-1}) \approx \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{s}_{tn}; \mathbf{D}\boldsymbol{\mu}_{t-1\,n}, \mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top + \boldsymbol{\Lambda}_{tn}). \tag{V.21}$$

Note that the above distribution factorizes across persons. Now that all the factors in (V.1) have tractable expressions, a VEM algorithm can be derived.

V.5 Variational Expectation Maximization

The proposed VEM algorithm iterates between an E-S-step, an E-Z-step, and an M-step on the following grounds.

V.5.1 E-S-step

the per-person variational posterior distribution of the state vector $q(\boldsymbol{s}_{tn})$ is evaluated by developing (V.16). The joint posterior $p(\mathbf{s}_t, \boldsymbol{z}_t|\mathbf{o}_{1:t})$ in (V.16) is the product of (V.2), (V.3) and (V.21). We thus first sum the logarithms of (V.2), of (V.3) and of (V.21). Then we ignore the terms that do not involve $\boldsymbol{s}_{tn}$. Evaluation of the expectation over all the latent variables except $\boldsymbol{s}_{tn}$ yields the following Gaussian distribution:

$$q(\boldsymbol{s}_{tn}) = \mathcal{N}(\boldsymbol{s}_{tn}; \boldsymbol{\mu}_{tn}, \boldsymbol{\Gamma}_{tn}), \tag{V.22}$$

[V.4] Even if the M-step is in closed-form, the inference is based on the variational posterior distributions. Therefore, the M-step could also be regarded as *variational*.

with:

$$\boldsymbol{\Gamma}_{tn} = \Bigg(\underbrace{\sum_{k=1}^{K}\sum_{r=1}^{R} \beta_{tknr} \mathbf{P}_g^\top \mathbf{L}_{kr}^\top \boldsymbol{\Sigma}_{kr}^{-1} \mathbf{L}_{kr} \mathbf{P}_g}_{\#1} + \underbrace{\sum_{m=1}^{M_t} \alpha_{tmn} \mathbf{P}_f^\top \boldsymbol{\Phi}_{tm}^{-1} \mathbf{P}_f}_{\#2} + \underbrace{\left(\boldsymbol{\Lambda}_{tn} + \mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top\right)^{-1}}_{\#3} \Bigg)^{-1}, \tag{V.23}$$

and with:

$$\boldsymbol{\mu}_{tn} = \boldsymbol{\Gamma}_{tn} \Bigg(\underbrace{\sum_{k=1}^{K}\sum_{r=1}^{R} \beta_{tknr} \mathbf{P}_g^\top \mathbf{L}_{kr}^\top \boldsymbol{\Sigma}_{kr}^{-1} (\boldsymbol{g}_{kr} - \boldsymbol{l}_{kr})}_{\#1} + \underbrace{\sum_{m=1}^{M_t} \alpha_{tmn} \mathbf{P}_f^\top \boldsymbol{\Phi}_{tm}^{-1} \boldsymbol{v}_{tm}}_{\#2} + \underbrace{\left(\boldsymbol{\Lambda}_{tn} + \mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top\right)^{-1} \mathbf{D}\boldsymbol{\mu}_{t-1\,n}}_{\#3} \Bigg), \tag{V.24}$$

where $\alpha_{tmn} = q(A_{tm} = n)$ and $\beta_{tknr} = q(B_{tk} = n, C_{tk} = r)$ are computed in the E-Z-step below. A key point is that, because of the recursive nature of the formulas above, it is sufficient to make the Gaussian assumption at $t = 1$, i.e. $q(\boldsymbol{s}_{1n}) = \mathcal{N}(\boldsymbol{s}_{1n}; \boldsymbol{\mu}_{1n}, \boldsymbol{\Gamma}_{1n})$, whose parameters may be easily initialized. It follows that $q(\boldsymbol{s}_{tn})$ is Gaussian at every frame.

We note that both (V.23) and (V.24) are composed of three terms: the first (#1), second (#2) and third terms (#3) of (V.23) and of (V.24) correspond to the audio, visual, and past cumulated information contributions to the precision matrix and the mean vector, respectively. Remind that the covariance $\boldsymbol{\Phi}_{tm}$ is associated with the visual observed variable in (V.10). Matrices $\mathbf{L}_{kr}$ and vectors $\boldsymbol{l}_{kr}$ characterize the piecewise affine mappings from the space of person locations to the space of audio features, i.e. Appendix V.9, and covariances $\boldsymbol{\Sigma}_{kr}$ capture the errors that are associated with both audio measurements and the piecewise affine approximation in (V.13). A similar interpretation holds for the three terms of (V.24).

V.5.2 E-Z-step

by developing (V.17), along the same reasoning as above, we obtain the following closed-form expression for the variational posterior distribution of the visual assignment variable:

$$\alpha_{tmn} = q(A_{tm} = n) = \frac{\tau_{tmn}\eta_{mn}}{\sum_{i=0}^{N} \tau_{tmi}\eta_{mi}}, \tag{V.25}$$

where τ_{tmn} is given by:

$$\tau_{tmn} = \begin{cases} \mathcal{N}(\boldsymbol{v}_{tm}; \mathbf{P}_f\boldsymbol{\mu}_{tn}, \boldsymbol{\Phi}_{tm}) e^{-\frac{1}{2}\mathrm{tr}\left(\mathbf{P}_f^\top \boldsymbol{\Phi}_{tm}^{-1} \mathbf{P}_f \boldsymbol{\Gamma}_{tn}\right)} \, \mathcal{B}(\boldsymbol{u}_{tm}; \boldsymbol{h}_n) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{v}_{tm}; \mathrm{vol}(\mathcal{V})) \mathcal{U}(\boldsymbol{u}_{tm}; \mathrm{vol}(\mathcal{H})) & \text{if } n = 0. \end{cases}$$

Similarly, for the variational posterior distribution of the audio assignment variables, developing (V.18) leads to:

$$\beta_{tknr} = q(B_{tk} = n, C_{tk} = r) = \frac{\kappa_{tknr}\rho_{kn}\pi_r}{\sum_{i=0}^{N}\sum_{j=1}^{R} \kappa_{tkij}\rho_{ki}\pi_j}, \tag{V.26}$$

where κ_{tknr} is given by:

$$\kappa_{tknr} = \begin{cases} \mathcal{N}(\boldsymbol{g}_{tk}; \mathbf{L}_{kr}\mathbf{P}_g\boldsymbol{\mu}_{tn} + \boldsymbol{l}_{kr}, \boldsymbol{\Sigma}_{kr}) e^{-\frac{1}{2}\mathrm{tr}\left(\mathbf{P}_g^\top \mathbf{L}_{kr}^\top \boldsymbol{\Sigma}_{kr}^{-1} \mathbf{L}_{kr} \mathbf{P}_g \boldsymbol{\Gamma}_{tn}\right)} \mathcal{N}(\tilde{\boldsymbol{x}}_{tn}; \boldsymbol{\nu}_r, \boldsymbol{\Omega}_r) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{g}_{tk}; \mathrm{vol}(\mathcal{G})) & \text{if } n = 0. \end{cases} \tag{V.27}$$

To obtain (V.27), an additional approximation is made. Indeed, the logarithm of (39) in Appendix V.9 is part of the complete-data log-likelihood and the denominator of this formula contains a weighted sum of Gaussian distributions. Taking the expectation of this term is not tractable because of the denominator. Based on the dynamical model (V.8), we replace the state variable $\boldsymbol{x}_{tn}$ in (39) with a "naive" estimate $\tilde{\boldsymbol{x}}_{tn}$ predicted from the position and velocity inferred at $t-1$: $\tilde{\boldsymbol{x}}_{tn} = \boldsymbol{x}_{t-1\,n} + \boldsymbol{y}_{t-1\,n}$.

V.5.3 M-step

The entries of the covariance matrix of the state dynamics, $\boldsymbol{\Lambda}_{tn}$, are the only parameters that need be estimated. To this aim, we develop $\mathbb{E}_{q(\mathbf{s}_t)q(\boldsymbol{z}_t)}[\log p(\mathbf{s}_t, \boldsymbol{z}_t | \mathbf{o}_{1:t})]$ and ignore the terms that do not depend on $\boldsymbol{\Lambda}_{tn}$. We obtain:

$$J(\boldsymbol{\Lambda}_{tn}) = \mathbb{E}_{q(\boldsymbol{s}_{tn})}\big[\log \mathcal{N}(\boldsymbol{s}_{tn}; \mathbf{D}\boldsymbol{\mu}_{t-1\,n}, \mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top + \boldsymbol{\Lambda}_{tn})\big],$$

which can be further developed as:

$$J(\boldsymbol{\Lambda}_{tn}) = \log|\mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top + \boldsymbol{\Lambda}_{tn}| + \mathrm{Tr}\big((\mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top + \boldsymbol{\Lambda}_{tn})^{-1}\left((\boldsymbol{\mu}_{tn} - \mathbf{D}\boldsymbol{\mu}_{t-1\,n})(\boldsymbol{\mu}_{tn} - \mathbf{D}\boldsymbol{\mu}_{t-1\,n})^\top + \boldsymbol{\Gamma}_{tn}\right)\big). \tag{V.28}$$

Hence, by differentiating (V.28) with respect to $\boldsymbol{\Lambda}_{tn}$ and equating to zero, we obtain:

$$\boldsymbol{\Lambda}_{tn} = \boldsymbol{\Gamma}_{tn} - \mathbf{D}\boldsymbol{\Gamma}_{t-1\,n}\mathbf{D}^\top + (\boldsymbol{\mu}_{tn} - \mathbf{D}\boldsymbol{\mu}_{t-1\,n})(\boldsymbol{\mu}_{tn} - \mathbf{D}\boldsymbol{\mu}_{t-1\,n})^\top. \tag{V.29}$$

Algorithm 4: Variational audio-visual tracking (VAVIT).

Input: visual observations $\mathbf{f}_{1:t} = \{\mathbf{v}_{1:t}, \xi_{1:t}\}$;
audio observations $\mathbf{g}_{1:t}$;
Output: Parameters of $q(\mathbf{s}_{1:t})$: $\{\boldsymbol{\mu}_{1:t,n}, \boldsymbol{\Gamma}_{1:t,n}\}_{n=0}^{N}$ (the estimated position of each person n is given by the two first entries of $\boldsymbol{\mu}_{1:t,n}$);
Person speaking status for $1:t$
Initialization;
for $t = 1$ *to end* **do**
  Gather visual and audio observations at frame t;
  Perform voice activity detection;
  Initialization of E-Z step;
  for $iter = 1$ *to* N_{iter} **do**
    E-Z-step (vision):
    for $m \in \{1, ..., M_t\}$ **do**
      for $n \in \{0, ..., N_t\}$ **do**
        Evaluate $q(A_{tm} = n)$ with (V.25);
      end
    end
    E-Z-step (audio):
    for $k \in \{1, ..., K_t\}$ **do**
      for $n \in \{0, ..., N_t\}$ *and* $r \in \{1, ..., R\}$ **do**
        Evaluate $q(B_{tk} = n, C_{tk} = r)$ with (V.26) and (V.27);
      end
    end
    E-S-step:
    for $n \in \{1, ..., N_t\}$ **do**
      Evaluate $\boldsymbol{\Gamma}_{tn}$ and $\boldsymbol{\mu}_{tn}$ with (V.23) and (V.24);
    end
    M-step: Evaluate $\boldsymbol{\Lambda}_{tn}$ with (V.29);
  end
  Perform birth (see Section V.6.2);
  Output the results;
end

V.6 Algorithm Implementation

The VEM procedure above will be referred to as VAVIT which stands for *variational audio-visual tracking*, and pseudo-code is shown in Algorithm 4. In theory, the order in which the two expectation steps are executed is not important. In practice, the issue of initialization is crucial. In our case, it is more convenient to start with the E-Z step rather than with the E-S step because the former is easier to initialize than the latter (see below). We start by explaining how the algorithm is initialized at $t = 1$ and then how the E-Z-step is initialized at each iteration. Next, we explain in detail the birth process. An interesting feature of the proposed method is that it allows to estimate who speaks when (i.e. perform speaker diarization) which is explained in detail at the end of the section.

V.6.1 Initialization

At $t = 1$ one must provide initial values for the parameters of the distributions (V.22), namely $\boldsymbol{\mu}_{1n}$ and $\boldsymbol{\Gamma}_{1n}$ for all $n \in \{1 \dots N\}$. These parameters are initialized as follows. The means are initialized at the image center and the covariances are given very large values, such that the variational distributions $q(\boldsymbol{s}_{1n})$ are non-informative. Once these parameters are initialized, they remain constant for a few frames, i.e. until the birth process is activated (see Section V.6.2 below).

As already mentioned, it is preferable to start with the E-Z-step than with the E-S-step because the initialization of the former is straightforward. Indeed, the E-S-step (Section V.5) requires current values for the posterior probabilities (V.25) and (V.27) which are estimated during the E-Z-step and which are both difficult to initialize. Conversely, the E-Z-step only requires current mean values, $\boldsymbol{\mu}_{tn}$, which can be easily initialized by using the model dynamics (V.8), namely $\boldsymbol{\mu}_{tn} = \mathbf{D}\boldsymbol{\mu}_{t-1n}$.

V.6.2 Birth Process

We now explain in detail the birth process, which is executed at the start of the tracking to initialize a latent variable for each detected person, as well as at any time t to detect new persons. The birth process considers B consecutive visual frames. At t, with $t > B$, we consider the set of visual observations assigned to $n = 0$ from $t - B$ to t, namely observations whose posteriors (V.25) are maximized for $n = 0$ (at initialization all the observations are in this case). We then build observation sequences from this set, namely sequences of the form $(\tilde{\boldsymbol{v}}_{m_{t-B}}, \ldots, \tilde{\boldsymbol{v}}_{m_t})_{\tilde{n}} \in \mathcal{B}$, where m_t indexes the set of observations at t assigned to $n = 0$ and $\tilde{n}$ indexes the set $\mathcal{B}$ of all such sequences. Notice that the birth process only uses the bounding-box center, width and size, $\boldsymbol{v}$, and that the descriptor $\boldsymbol{u}$ is not used. Hence the birth process is only based on the smoothness of an observed sequence of bounding boxes. Let's consider the marginal likelihood of a sequence $\tilde{n}$, namely:

$$\begin{aligned}\mathcal{L}_{\tilde{n}} &= p((\tilde{\boldsymbol{v}}_{m_{t-B}}, \ldots, \tilde{\boldsymbol{v}}_{m_t})_{\tilde{n}}) \\ &= \int \cdots \int p(\tilde{\boldsymbol{v}}_{m_{t-B}} | \boldsymbol{s}_{t-B\,\tilde{n}}) \ldots p(\tilde{\boldsymbol{v}}_{m_t} | \boldsymbol{s}_{t\,\tilde{n}}) p(\boldsymbol{s}_{t\,\tilde{n}} | \boldsymbol{s}_{t-1\,\tilde{n}}) \ldots p(\boldsymbol{s}_{t-B+1\,\tilde{n}} | \boldsymbol{s}_{t-B\,\tilde{n}}) p(\boldsymbol{s}_{t-B\,\tilde{n}}) d\boldsymbol{s}_{t-B:t\,\tilde{n}},\end{aligned} \tag{V.30}$$

where $\boldsymbol{s}_{t,\tilde{n}}$ is the latent variable already defined and $\tilde{n}$ indexes the set $\mathcal{B}$. All the probability distributions in (V.30) were already defined, namely (V.8) and (V.10), with the exception of $p(\boldsymbol{s}_{t-B,\tilde{n}})$. Without loss of generality, we can assume that the latter is a normal distribution centered at $\tilde{\boldsymbol{v}}_{m_t}$ and with a large covariance. Therefore, the evaluation of (V.30) yields a closed-form expression for $\mathcal{L}_{\tilde{n}}$. A sequence $\tilde{n}$ generated by a person is likely to be smooth and hence $\mathcal{L}_{\tilde{n}}$ is high, while for a non-smooth sequence the marginal likelihood is low. A newborn person is therefore created from a sequence of observations $\tilde{n}$ if $\mathcal{L}_{\tilde{n}} > \tau$, where τ is a user-defined parameter. As just mentioned, the birth process is executed to initialize persons as well as along time to add new persons. In practice, in (V.30) we set $B = 3$ and hence, from $t = 1$ to $t = 4$ all the observations are initially assigned to $n = 0$.

V.6.3 Speaker Diarization

Speaker diarization consists of assigning temporal segment of speech to persons [146]. We introduce a binary variable χ_{tn} such that $\chi_{tn} = 1$ if person n speaks at time t and $\chi_{tn} = 0$ otherwise. Traditionally, speaker diarization is based on the following assumptions. First, it is assumed that speech signals are sparse in the time-frequency domain. Second, it is assumed that each time-frequency point in such a spectrogram corresponds to a single speech source. Therefore, the proposed speaker diarization method is based on assigning time-frequency points to persons.

In the case of the proposed model, speaker diarization can be coarsely inferred from frequency sub-bands in the following way. The posterior probability that the speech signal available in the frequency sub-band k at frame t was uttered by person n, given the audio observation $\boldsymbol{g}_{tk}$, is:

$$p(B_{tk} = n | \boldsymbol{g}_{tk}) = \sum_{r=1}^{R} p(B_{tk} = n, C_{tk} = r | \boldsymbol{g}_{tk}), \tag{V.31}$$

where B_{tk} is the audio assignment variable and C_{tk} is the affine-mapping assignment variable defined in Section V.3.4 and in Appendix V.9. Using the variational approximation (V.26), this probability becomes:

$$p(B_{tk} = n | \boldsymbol{g}_{tk}) \approx \sum_{r=1}^{R} q(B_{tk} = n, C_{tk} = r) = \sum_{r=1}^{R} \beta_{tknr}, \tag{V.32}$$

and by accumulating probabilities over all the frequency sub-bands, we obtain the following formula:

$$\chi_{tn} = \begin{cases} 1 & \text{if } \frac{1}{K_t} \sum_{k=1}^{K_t} \sum_{r=1}^{R} \beta_{tknr} \geq \gamma \\ 0 & \text{otherwise,} \end{cases} \tag{V.33}$$

where γ is a user-defined threshold. Note that there is no dynamic model associated with diarization: χ_{tn} is estimated independently at each frame and for each person. More sophisticated diarization models can be found in [136], [147].

V.7 Experiments

V.7.1 Experimental Protocol

The AVDIAR Dataset We used the AVDIAR[V.5] dataset [136] to evaluate the performance of the proposed audio-visual tracking method. This dataset is challenging in terms of audio-visual analysis. There are several participants involved

[V.5] https://team.inria.fr/perception/avdiar/

in informal conversations while wandering around. They are in between two and four meters away from the audio-visual recording device. They take speech turns and often there are speech overlaps. They turn their faces away from the camera. The dataset is annotated as follows: The visual annotations comprise the centers, widths and heights of two bounding boxes for each person and in each video frame, a face bounding box and an upper-body bounding box. An identity (a number) is associated with each person through the entire dataset. The audio annotations comprise the speech status of each person over time (speaking or silent), with a minimum speech duration of 0.2 s. The audio source locations correspond to the centers of the face bounding boxes.

The dataset was recorded with a sensor composed of two cameras and six microphones, but only one camera is used in the experiments described below. The videos were recorded at 25 FPS. The frame resolution is of 1920×1200 pixels corresponding to a field of view of $97^\circ \times 80^\circ$. The microphone signals are sampled at 16 kHz. The dataset was recorded into two different rooms, *living-room* and *meeting-room*, e.g. Fig. V.2 and Fig. V.3. These two rooms have quite different lighting conditions and acoustic properties (size, presence of furniture, background noise, etc.). Altogether there are 18 sequences associated with living-room (26927 video frames) and 6 sequences with meeting-room (6031 video frames). Additionally, there are two training datasets, $\mathcal{T}_1$ and $\mathcal{T}_2$ (one for each room) that contain input-output pairs of multichannel audio features and audio-source locations that allow to estimate the parameters of (V.13) using the method of [133]. This yields a mapping between source locations in the image plane, $\boldsymbol{x}$, and audio features, $\boldsymbol{g}$. Audio feature extraction is described in detail below.

One interesting characteristic of the proposed tracking is its flexibility in dealing only with visual data, only with audio data, or with visual and audio data. Moreover, the algorithm is able to automatically switch from unimodal (audio or visual) to multimodal (audio and visual). In order to quantitatively assess the performance and merits of each one of these variants we used two configurations:

- *Full camera field of view (FFOV)*: The entire horizontal field of view of the camera, i.e. 1920 pixels, or 97°, is being used, such that visual and audio observations, **if any**, are simultaneously available, and
- *Partial camera field of view (PFOV)*: The horizontal field of view is restricted to 768 pixels (or 49°) and there are two *blind* strips (576 pixels each) on its left- and right-hand sides; the *audio field of view* remains unchanged, 1920 pixels, or 97°.

The PFOV configuration allows us to test scenarios in which a participant may leave the camera field of view and still be heard. Notice that since ground-truth annotations are available for the full field of view, it is possible to assess the performance of the tracker using audio observations only, as well as to analyse the behavior of the tracker when it switches from audio-only tracking to audio-visual tracking.

The AV16.3 Dataset We also used the twelve recordings of the AV16.3 dataset [135] to evaluate the proposed method and to compare it with [121] and with [124]. The dataset was recorded in a meeting room. The videos were recorded at 25 FPS with three cameras fixed on the room ceiling. The image resolution is of 288×360 pixels. The audio signals were recorded with two eight-microphone circular arrays, both placed onto a table top, and sampled at 16 kHz. In addition, the dataset comes with internal camera calibration parameters, as well as with external calibration parameters, namely camera-to-camera and microphone-array-to-camera calibration parameters. We note that the scenarios associated with AV16.3 are somehow artificial in the sense that *the participants speak simultaneously and continuously*. This stays in contrast with the AVDIAR recordings where people take speech turns in informal conversations.

Audio Features In the case of AVDIAR, the STFT (short-time Fourier transform) [134] is applied to each microphone signal using a 16 ms Hann window (256 audio samples per window) and with an 8 ms shift between successive windows (50% overlap), leading to 128 frequency bins and to 125 audio FPS. Inter-microphone spectral features are then computed using [132]. These features – referred to in [132] as *direct-path relative transfer function (DP-RTF) features* – are robust against background noise and against reverberations, hence they do not depend on the acoustic properties of the recording room, as they encode the direct path from the audio source to the microphones. Nevertheless, they may depend on the orientation of the speaker's face. If the microphones are positioned behind a speaker, the direct-path sound wave (from the speaker to the microphones) propagates through the speaker's head, hence it is attenuated. This may have a negative impact on the direct-to-reverberation ratio. Here we assume that, altogether, this has a limited effect.

The audio features are averaged over five audio frames in order to be properly aligned with the video frames. The feature vector is then split into $K = 16$ sub-bands, each sub-band being composed of $J = 8$ frequencies; sub-bands with low energy are disregarded. This yields the set of audio observations at t, $\{\boldsymbol{g}_{tk}\}_{k=1}^{K_t}$, $K_t \leq K$ (see Section V.3.4 and Appendix V.9).

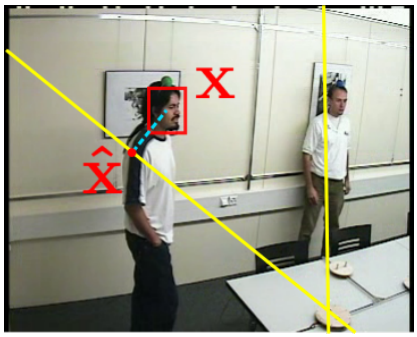


Figure V.1: *This figure displays two DOAs, associated with one microphone array (bottom left), projected onto the image plane, and illustrates the geometric relationship between a DOA and the current location of a speaker.*

Interestingly, the computed inter-microphone DP-RTF features can be mapped onto the image plane and hence they can be used to estimate directions of arrival (DOAs). Please consult [133] for more details. Alternatively, one can compute DOAs explicitly from time differences of arrival (TDOAs) between the microphones of a microphone array, provided that the inter-microphone geometry is known. The disadvantage is that DOAs based on TDOAs assume free-field acoustic-wave propagation and hence they don't have a built-in reverberation model. Moreover, if the camera parameters are known and if the camera location (extrinsic parameters) is known in the coordinate frame of the microphone array, as is the case with the AV16.3 dataset, it is possible to project DOAs onto the image plane. We use the multiple-speaker DOA estimator of [148] as it provides accurate results for the AV16.3 sensor setup [135]. Let d_{tk} be the line corresponding to the projection of a DOA onto the image plane and let $\boldsymbol{x}_{tn}$ be the location of person n at time t. It is straightforward to determine the point $\hat{\boldsymbol{x}}_{tk} \in d_{tk}$ the closest to $\boldsymbol{x}_{tn}$, e.g. Fig. V.1. Hence the inter-channel spectral features $\{\boldsymbol{g}_{tk}\}_{k=1}^{K_t}$ are replaced with $\{\hat{\boldsymbol{x}}_{tk}\}_{k=1}^{K_t}$ and (V.13) is replaced with:

$$p(\hat{\boldsymbol{x}}_{tk}|\boldsymbol{x}_{tn}, B_{tk} = n) = \begin{cases} \mathcal{N}(\hat{\boldsymbol{x}}_{tk}; \boldsymbol{x}_{tn}, \sigma\mathbf{I}) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\hat{\boldsymbol{x}}_{tk}; \text{vol}(\mathcal{X})) & \text{if } n = 0, \end{cases} \tag{V.34}$$

where $\sigma\mathbf{I}$ is an isotropic covariance that models the uncertainty of the DOA, e.g. Fig. V.4, third row.

Visual Features In both AVDIAR and AV16.3 datasets participants do not always face the cameras and hence face detection is not robust. Instead we use the person detector of [149] from which we infer a body bounding-box and a head bounding-box. We use the person re-identification CNN-based method [150] to extract an embedding (i.e. a person descriptor) from the body bounding-box. This yields the feature vectors $\{\boldsymbol{u}_{tm}\}_{m=1}^{M_t} \subset \mathbb{R}^{2048}$ (Section V.3.3). Similarly, the center, width and height of the head bounding-box yield the observations $\{\boldsymbol{v}_{tm}\}_{m=1}^{M_t} \subset \mathbb{R}^4$ at each frame t.

Evaluation Metrics We used standard multi-object tracking (MOT) metrics [151] to quantitatively evaluate the performance of the proposed tracking algorithm. The multi-object tracking accuracy (MOTA) is the most commonly used metric for MOT. It is a combination of false positives (FP), false negatives (FN; i.e. missed persons), and identity switches (IDs), and is defined as:

$$\text{MOTA} = 100\left(1 - \frac{\sum_t(\text{FP}_t + \text{FN}_t + \text{IDs}_t)}{\sum_t \text{GT}_t}\right), \tag{V.35}$$

where GT stands for the ground-truth person trajectories. After comparison with GT trajectories, each estimated trajectory can be classified as mostly tracked (MT) and mostly lost (ML) depending on whether a trajectory is covered by correct estimates more than 80% of the time (MT) or less than 20% of the time (ML). In the tables below, MT and ML indicated the percentage of ground-truth tracks under each situation.

In addition to MOT, we also used the OSPA-T metric [137]. OSPA-T is based on a distance between two point sets and combines various aspects of tracking performance, such as timeliness, track accuracy, continuity, data associations and false tracks. It should be noted that OSPA-T involves a number of parameters whose values must be provided in advance. We used the publicly available code provided by one of the authors of [137] for computing the OSPA-T scores in all the experimental evaluations reported below.[V.6]

[V.6] `http://ba-tuong.vo-au.com/codes.html`

Table V.1: *OSPA-T and MOT scores for the living-room sequences (full camera field of view)*

Method	OSPA-T(↓)	MOTA(↑)	FP(↓)	FN(↓)	IDs(↓)	MT(↑)	ML(↓)
[121]	28.12	10.37	44.64 %	43.95%	732	20%	7.5 %
[124]	30.03	18.96	8.13 %	72.09%	581	17.5%	52.5%
[125]	**14.79**	**96.32**	**1.77%**	**1.79%**	**80**	**92.5%**	**0%**
VAVIT	17.05	96.03	1.85%	2.0%	86	**92.5%**	**0%**

Table V.2: *OSPA-T and MOT scores for the meeting-room sequences (full camera field of view).*

Method	OSPA-T (↓)	MOTA(↑)	FP(↓)	FN(↓)	IDs(↓)	MT(↑)	ML(↓)
[121]	5.76	62.43	18.63%	17.19%	297	70.59 %	**0%**
[124]	7.83	28.48	0.93%	69.68%	155	0 %	52.94%
[125]	**3.02**	**98.50**	**0.25%**	**1.11%**	**25**	**100.00%**	**0%**
VAVIT	3.57	98.16	0.38%	1.27%	32	**100.00%**	**0%**

Table V.3: *OSPA-T and MOT scores for the living-room sequences (partial camera field of view).*

Method	OSPA-T(↓)	MOTA(↑)	FP(↓)	FN(↓)	IDs(↓)	MT(↑)	ML(↓)
[121]	28.14	17.82	36.86%	42.88%	1722	32.50%	7.5%
[124]	29.73	20.61	5.54%	72.45%	989	12.5%	40%
[125]	22.25	66.39	**0.48%**	32.95%	**129**	45%	7.5%
VAVIT	**21.77**	**69.62**	8.97%	**21.18%**	152	**70%**	**5%**

Table V.4: *OSPA-T and MOT scores for the meeting-room sequences (partial camera field of view).*

Method	OSPA-T(↓)	MOTA(↑)	FP(↓)	FN(↓)	IDs(↓)	MT(↑)	ML(↓)
[121]	7.23	29.04	23.05%	45.19 %	461	29.41%	17.65%
[124]	8.17	26.95	1.05%	70.62%	234	5.88%	52.94%
[125]	**5.80**	64.24	**0.43%**	35.18%	**24**	36.84%	15.79%
VAVIT	5.81	**65.27**	5.07%	**29.5%**	26	**47.37%**	**10.53%**

Table V.5: *OSPA-T and MOT scores obtained with the AV16.3 dataset.*

Method	OSPA-T (↓)	MOTA(↑)	FP(↓)	FN(↓)	IDs(↓)	MT(↑)	ML(↓)
[124]	17.28	36.4	16.72%	42.22%	765	11.11%	**0%**
[125]	13.32	82.9	**5.29%**	11.5 %	51	85.2%	**0%**
VAVIT	**10.88**	**84.1**	6.51%	**9.18%**	**29**	**92.6%**	**0%**

In our experiments, the threshold of overlap to consider that a ground truth is covered by an estimation is set to 0.1 intersection over union (IoU). In the PFOV configuration, we need to evaluate the audio-only tracking, i.e. the speakers are in the blind areas. As mentioned before, audio localization is less accurate than visual localization. Therefore, for evaluating the audio-only tracker we relax by a factor of two the expected localization accuracy with respect to the audio-visual localization accuracy.

V.7.2 Results and Discussion

Benchmarking with Baseline Methods To quantitatively evaluate its performance, we benchmarked the proposed method with two state-of-the-art audio-visual tracking methods. The first one is the audio-assisted video adaptive

particle filtering (AS-VA-PF) method of [121], and the second one is the sparse audio-visual mean-shift sequential Monte-Carlo probability hypothesis density (AV-MSSMC-PHD) method of [124].

Notice that both these methods do not make recourse to a person detector as they use a tracking-by-detection paradigm. This stays in contrast with our method which uses a person detector and probabilistically assigns each detection to each person. In principle, the baseline methods can be modified to accept person detection as visual information. However, we did not modify the baseline methods and used the software provided by the authors of [121] and [124].

Sound locations are used to reshape the typical Gaussian noise distribution of particles in a propagation step, then [121] uses the particles to weight the observation model. [124] uses audio information to improve the performance and robustness of a visual SMC-PHD filter. Both [121] and [124] require input from a multiple sound-source localization (SSL) algorithm. In the case of AVDIAR recordings, the multi-speaker localization method proposed in [132] is used to provide input to [121] and [124].[V.7] In the case of AV16.3 recordings the method of [135] is used to provide DOAs to [121], [124] and to our method, as explained above.

We also compare the proposed method with a visual multiple-person tracker, more specifically the *online Bayesian variational tracker* (OBVT) of [125], which is based on a similar variational inference as the one presented here. In [125] visual observations were provided by color histograms. In our benchmark, for the sake of fairness, the proposed tracker and [125] share the same visual observations.

The OSPA-T and MOT scores obtained with these methods as well as the proposed method are reported in Table V.1, Table V.2, Table V.3, Table V.4, and Table V.5. The symbols $\uparrow$ and $\downarrow$ indicate higher the better and lower the better, respectively. In the case of AVDIAR, we report results with both meeting-room and living-room in the two configurations: FFOV, Table V.1 and Table V.2 and PFOV, Table V.3 and Table V.4. In the case of AV16.3 we report results with the twelve recordings commonly used by audio-visual tracking algorithms, Table V.5.

The most informative metrics are OSPA-T and MOTA (MOT accuracy) and one can easily see that both [125] and the proposed method outperform the other two methods. The poorer performance of both [121] and [124] for all the configurations is generally explained by the fact that these two methods expect audio and visual observations to be simultaneously available. In particular, [121] is not robust against visual occlusions, which leads to poor IDs (identity switches) scores.

The AV-MSSMC-PHD method [124] uses audio information in order to count the number of speakers. In practice, we noticed that the algorithm behaves differently with the two datasets. In the case of AVDIAR, we noticed that the algorithm assigns several visible participants to the same audio source, since in most of the cases there is only one active audio source at a time. In the case of AV16.3 the algorithm performs much better, since participants speak simultaneously and continuously. This explains why both FN (false negatives) and IDs (identity switches) scores are high in the case of AVDIAR, i.e. Tables V.1, V.2, and V.3.

One can notice that in the case of FFOV, [125] and the proposed method yield similar results in terms of OSPA-T and MOT scores: both methods exhibit low OSPA-T, FP, FN and IDs scores and, consequently, high MOTA scores. Moreover, they have very good MT and ML scores (out of 40 sequences 37 are mostly tracked, 3 are partially tracked, and none is mostly lost). As expected, the inferred trajectories are more accurate for visual tracking (whenever visual observations are available) than for audio-visual tracking: indeed, the latter fuses visual and audio observations which slightly degrades the accuracy because audio localization is less accurate than visual localization.

As for the PFOV configuration (Table V.3 and Table V.4), the proposed algorithm yields the best MOTA scores both for meeting-room and for living-room. Both [121] and [124] have difficulties when visual information is not available: both these algorithms fail to track speakers when they walk outside the visual field of view. While [124] can detect a speaker when it re-enters the visual field of view, [121] cannot. Obviously, the visual-only tracker [125] fails outside the camera field of view.

Audio-Visual Tracking Examples We now provide and discuss results obtained with three AVDIAR recordings and one AV16.3 recording, namely the FFOV recording Seq13-4P-S2-M1 (Fig. V.2), the PFOV recordings Seq19-2P-S1M1 (Fig. V.3) and Seq22-1P-S0M1 (Fig. V.5), and the seq45-3p-1111 recording of AV16.3 (Fig. V.4).[V.8] All these recordings are challenging in terms of audio-visual tracking: participants are seated, then they stand up or they wander around. In the case of AVDIAR, some participants take speech turns and interrupt each other, while others remain silent.

The first rows of Fig. V.2, Fig. V.3 and Fig. V.4 show four frames sampled from two AVDIAR recordings and one AV16.3 recording, respectively. The second rows show ellipses of constant density that correspond to visual uncertainty (covariances). The third rows show the audio uncertainty. The audio uncertainties (covariances) are much

[V.7] The authors of [121] and [124] kindly provided their software packages.

[V.8] `https://team.inria.fr/perception/research/variational_av_tracking/`

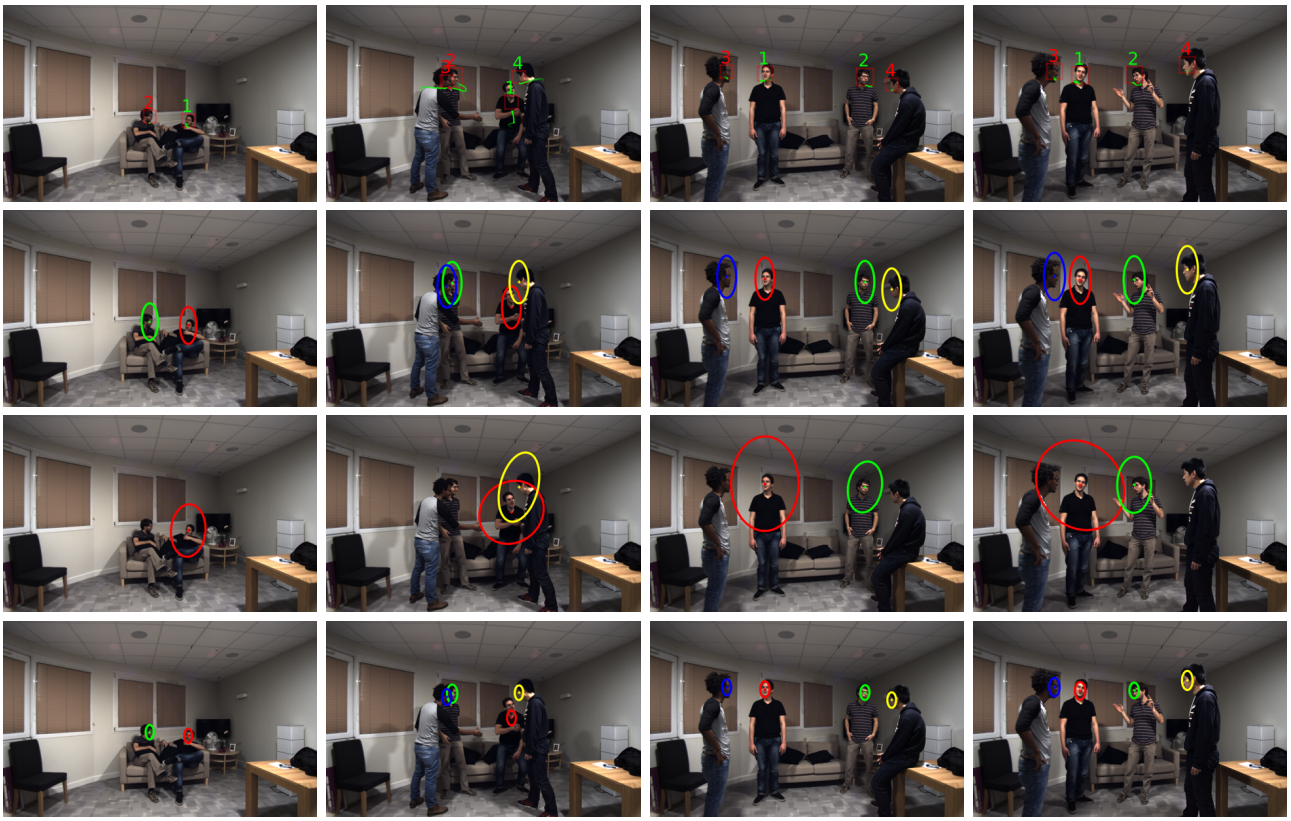


Figure V.2: *Four frames sampled from Seq13-4P-S2M1 (living room). First row: green digits denote speakers while red digits denote silent participants. Second, third and fourth rows: the ellipses visualize the visual, audio, and dynamic covariances, respectively, of each tracked person. The tracked persons are color-coded: green, yellow, blue, and red.*

Figure V.3: *Four frames sampled from Seq19-2P-S1M1 (living room). The camera field of view is limited to the central strip. Whenever the participants are outside the central strip, the tracker entirely relies on audio observations and on the model's dynamics.*

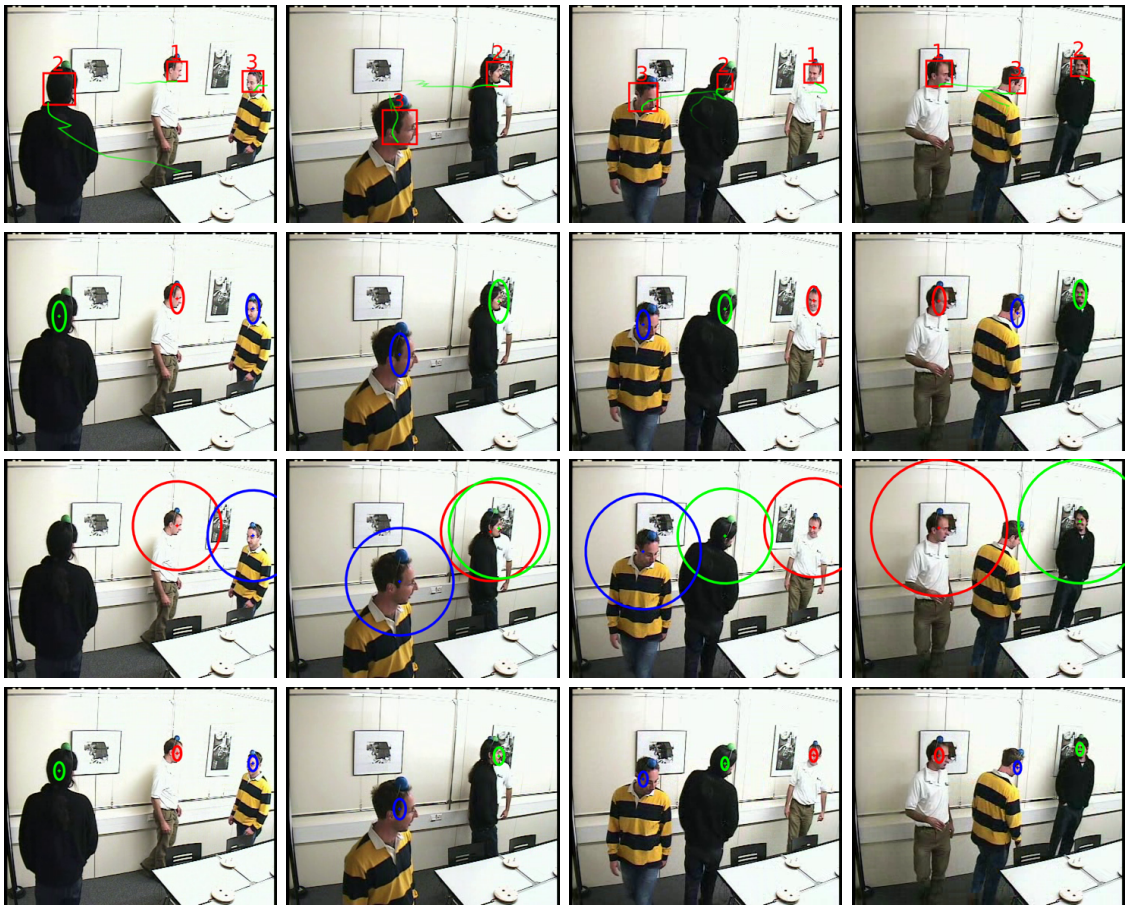
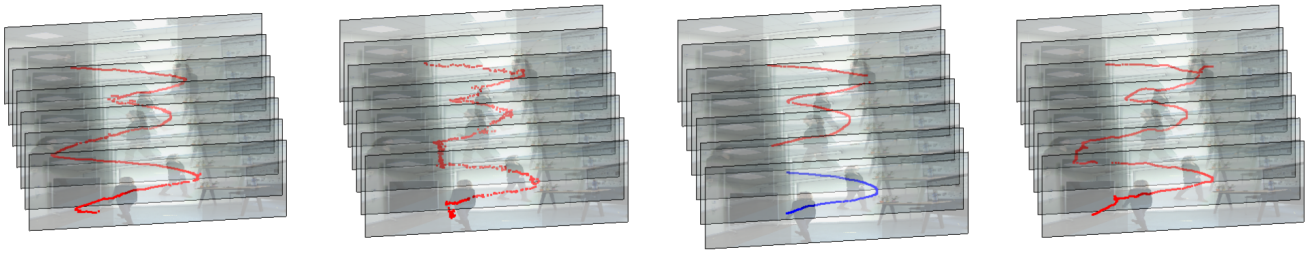


Figure V.4: *Four frames sampled from seq45-3p-1111 of AV16.3. In this dataset, the participants speak simultaneously and continuously.*



(a) Ground-truth trajectory (b) AS-VA-PF [121] (c) OBVT [125] (d) VAVIT (proposed)

Figure V.5: *Trajectories associated with a tracked person under the PFOV configuration (sequence Seq22-1P-S0M1 recorded in meeting room). The ground-truth trajectory (a) corresponds to the center of the bounding-box of the head. The trajectory (b) obtained with [121] is non-smooth. Both [121] and [125] fail to track outside the camera field of view. In the case of the OBVT trajectory (c), there is an identity switch, from "red" (before the person leaves the visual field of view) to "blue" (after the person re-enters in the visual field of view).*

larger than the visual ones since audio localization is less accurate than visual localization. The fourth rows shows the contribution of the dynamic model to the uncertainty, i.e. the inverse of the precision (#3) in eq. (V.23). Notice that these "dynamic" covariances are small, in comparison with the "observation" covariances. This ensures tracking continuity (smooth tracjectories) when audio or visual observations are either weak or totally absent. Fig. V.3 shows a tracking example with a partial camera field of view (PFOV) configuration. In this case, audio and visual observations are barely available simultaneously. The independence of the visual and audio observation models and their fusion within the same dynamic model guarantees robust tracking in this case.

Fig. V.5 shows the ground-truth trajectory of a person and the trajectories estimated with the audio-visual tracker [121], with the visual tracker [125], and with the proposed method. The ground-truth trajectory corresponds to a sequence

Table V.6: *DER (diarization error rate) scores obtained with the AVDIAR dataset.*

Sequence	DiarTK [138]	[136]	Proposed (FFOV)	Proposed (PFOV)
Seq01-1P-S0M1	43.19	3.32	1.64	1.86
Seq02-1P-S0M1	49.9	-	2.38	2.09
Seq03-1P-S0M1	47.25	-	6.59	14.65
Seq04-1P-S0M1	32.62	9.44	4.96	10.45
Seq05-2P-S1M0	37.76	-	29.76	30.78
Seq06-2P-S1M0	56.12	-	14.72	15.83
Seq07-2P-S1M0	41.43	-	42.36	37.56
Seq08-3P-S1M1	31.5	-	38.4	48.86
Seq09-3P-S1M1	52.74	-	38.26	68.81
Seq10-3P-S1M1	56.95	-	54.26	54.04
Seq12-3P-S1M1	63.67	17.32	44.67	47.25
Seq13-4P-S2M1	47.56	29.62	43.45	43.17
Seq15-4P-S2M1	62.53	-	41.49	64.38
Seq17-2P-S1M1	17.24	-	16.53	15.63
Seq18-2P-S1M1	35.05	-	19.55	20.58
Seq19-2P-S1M1	38.96	-	26.47	27.84
Seq20-2P-S1M1	43.58	35.46	38.24	44.3
Seq21-2P-S1M1	32.22	20.93	25.87	25.9
Seq22-1P-S0M1	23.53	4.93	2.79	3.32
Seq27-3P-S1M1	46.05	18.72	47.07	54.75
Seq28-3P-S1M1	30.68	-	23.54	31.77
Seq29-3P-S1M0	38.68	-	30.74	35.92
Seq30-3P-S1M1	51.15	-	49.71	57.94
Seq32-4P-S1M1	41.51	30.20	46.25	43.03
Overall	42.58	**18.88**	28.73	33.36

of bounding-box centers. Both [121] and [125] failed to estimate a correct trajectory. Indeed, [121] requires simultaneous availability of audio-visual data while [125] cannot track outside the visual field of view. Notice the non-smooth trajectory obtained with [121] in comparison with the smooth trajectories obtained with variational inference, i.e. [125] and proposed.

Speaker Diarization Results As already mentioned in Section V.6.3, speaker diarization information can be extracted from the output of the proposed VAVIT algorithm. Notice that, while audio diarization is an extremely well investigated topic, audio-visual diarization has received much less attention. In [147] it is proposed an audio-visual diarization method based on a dynamic Bayesian network that is applied to video conferencing. Their method assumes that participants take speech turns with a small silent interval between turns, which is an unrealistic hypothesis in the general case. The diarization method of [152] requires audio, depth and RGB data. More recently, [136] proposed a Bayesian dynamic model for audio-visual diarization that takes as input fused audio-visual information. Since diarization is not our main objective, we only compared our diarization results with [136], which achieves state of the art results, and with the diarization toolkit of [138] which only considers audio information.

The diarization error rate (DER) is generally used as a quantitative measure. As is the case with MOT, DER combines false positives (FP), false negatives (FN) and identity swithches (IDs) scores within a single metric. The NIST-RT evaluation toolbox[V.9] is used. The results obtained with [136], [138] and with the proposed method are reported in Table V.6, for both the full field-of-view and partial field-of-view configurations (FFOV and PFOV). The proposed method performs better than the audio-only baseline method [138]. In comparison with [136], the proposed method performs slightly less well despite the lack of a special-purpose diarization model. Indeed, [136] implements diarization within a hidden Markov model (HMM) that takes into account both diarization dynamics and the audio activity observed at each time step, whereas our method is only based on observing the audio activity over time.

The ability of the proposed audio-visual tracker to perform diarization is illustrated in Fig. V.6 and in Fig. V.7 with a FFOV sequence (Seq13-4P-S2-M1, Fig. V.2) and with a PFOV sequence (Seq19-2P-S1M1, Fig. V.3), respectively.

[V.9] `https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation`

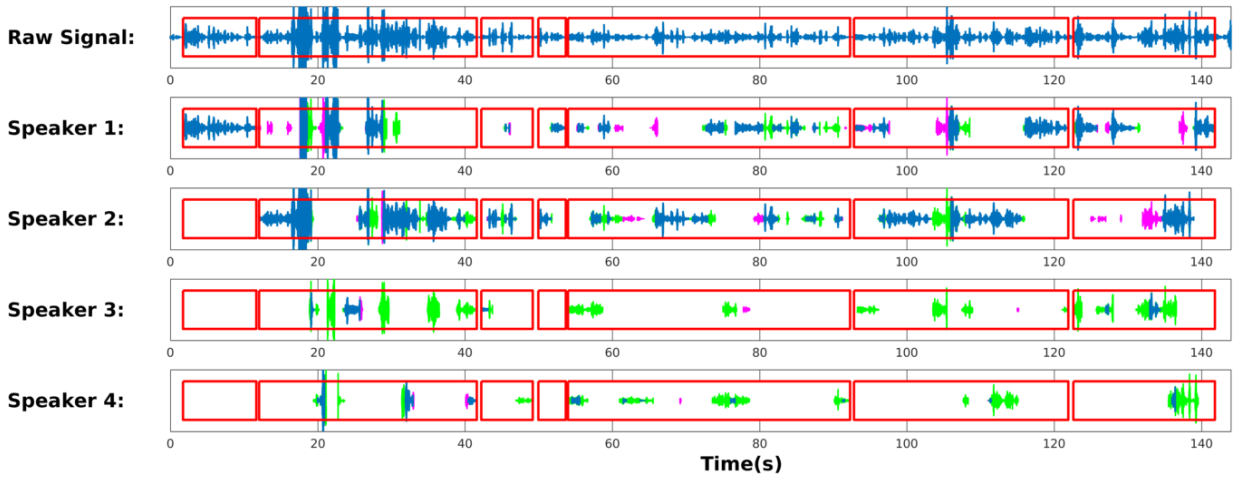


Figure V.6: *Diarization results obtained with Seq13-4P-S2M1 (FFOV). The first row shows the audio signal recorded with one of the microphones. The red boxes show the result of the voice activity detector which is applied to all the microphone signals prior to tracking. For each speaker, correct detections are shown in blue, missed detections are shown in green, and false positives are shown in magenta*

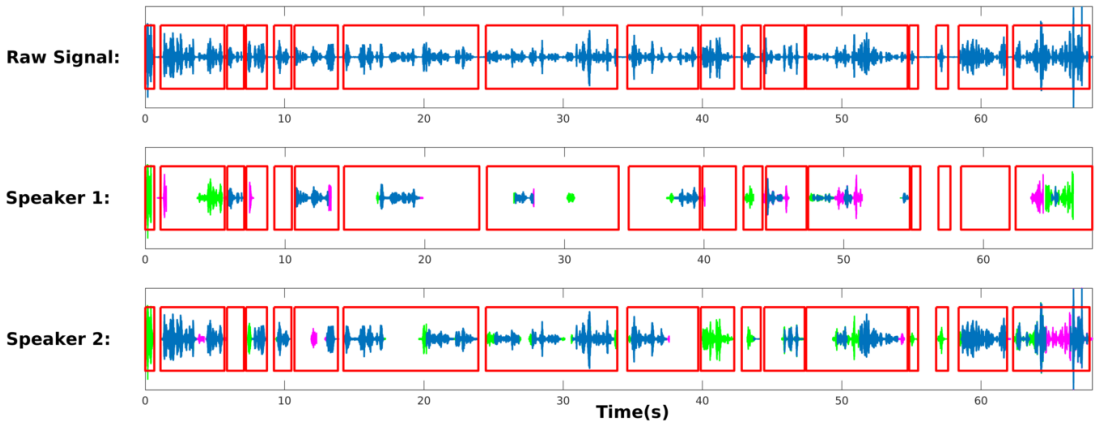


Figure V.7: *Diarization results obtained with Seq19-2P-S1M1 (PFOV).*

V.8 Conclusions

We addressed the problem of tracking multiple speakers using audio and visual data. It is well known that the generalization of single-person tracking to multiple-person tracking is computationally intractable and a number of methods were proposed in the past. Among these methods, sampling methods based on particle filtering (PF) or on PHD filters have recently achieved the best tracking results. However, these methods have several drawbacks: (i) the quality of the approximation of the filtering distribution increases with the number of particles, which also increases the computational burden, (ii) the observation-to-person association problem is not explicitly modeled and a post-processing association mechanism must be invoked, and (iii) audio and visual observations must be available simultaneously and continuously. Some of these limitations were recently addressed both in [121] and in [124], where audio observations were used to compensate the temporal absence of visual observations. Nevertheless, people speak with pauses and hence audio observations are rarely continuously available.

In contrast, we proposed a variational approximation of the filtering distribution and we derived a closed-form variational expectation-maximization algorithm. The observation-to-person association problem is fully integrated in our model, rather than as a post-processing stage. The proposed VAVIT algorithm is able to deal with intermittent audio or visual observations, such that one modality can compensate the other modality, whenever one of them is noisy, too weak or totally missing. Using the OSPA-T and MOT scores we showed that the proposed method outperforms the PF-based method [121].

V.9 Appendix: Learning the parameters of the linear transformations

In this appendix we describe the audio observation model we use. More precisely, we make explicit the generative model introduced in (V.13). For that purpose we consider a training set of audio features, or inter-channel spectral features (which in practice correspond to the real and imaginary parts of complex-valued Fourier coefficients) and their associated source locations, $\mathcal{T} = \{(\boldsymbol{g}_i, \boldsymbol{x}_i)\}_{i=1}^{I}$ and let $(\boldsymbol{g}, \boldsymbol{x}) \in \mathcal{T}$. The vector $\boldsymbol{g}$ is the concatenation of K vectors $\boldsymbol{g} = [\boldsymbol{g}_1 | \dots \boldsymbol{g}_k | \dots \boldsymbol{g}_K]$ where $[\cdot|\cdot]$ denotes vertical vector concatenation. We recall that for all sub-bands $k; 1 \leq k \leq K$, $\boldsymbol{g}_k \in \mathbb{R}^{2J}$ where J is the number of frequencies in each sub-band. Without loss of generality we consider the sub-band k. The joint probability of $(\boldsymbol{g}_k, \boldsymbol{x})$ can be marginalized as:

$$p(\boldsymbol{g}_k, \boldsymbol{x}) = \sum_{r=1}^{R} p(\boldsymbol{g}_k | \boldsymbol{x}, C_k = r) p(\boldsymbol{x}_k | C_k = r) p(C_k = r). \tag{V.36}$$

Assuming Gaussian variables, we have $p(\boldsymbol{g}_k | \boldsymbol{x}, C_k = r) = \mathcal{N}(\boldsymbol{g}_k | h_{kr}(\boldsymbol{x}), \boldsymbol{\Sigma}_{kr})$, $p(\boldsymbol{x} | C_k = r) = \mathcal{N}(\boldsymbol{x} | \boldsymbol{\nu}_{kr}, \boldsymbol{\Omega}_{kr})$, and $p(C_k = r) = \pi_{kr}$, where $h_{kr}(\boldsymbol{x}) = \mathbf{L}_{kr}\boldsymbol{x} + \boldsymbol{l}_{kr}$ with $\mathbf{L}_{kr} \in \mathbb{R}^{2J \times 2}$ and $\boldsymbol{l}_{kr} \in \mathbb{R}^{2J}$, $\boldsymbol{\Sigma}_r \in \mathbb{R}^{2J \times 2J}$ is the associated covariance matrix, and $\boldsymbol{x}$ is drawn from a Gaussian mixture model with R components, each component r being characterized by a prior π_{kr}, a mean $\boldsymbol{\nu}_{kr} \in \mathbb{R}^2$ and a covariance $\boldsymbol{\Omega}_{kr} \in \mathbb{R}^{2 \times 2}$. The parameter set of this model for sub-band k is:

$$\boldsymbol{\Theta}_k = \{\mathbf{L}_{kr}, \boldsymbol{l}_{kr}, \boldsymbol{\Sigma}_{kr}, \boldsymbol{\nu}_{kr}, \boldsymbol{\Omega}_{kr}, \pi_{kr}\}_{r=1}^{r=R}. \tag{V.37}$$

These parameters can be estimated via a closed-form EM procedure from a training dataset, e.g. $\mathcal{T}$ (please consult [133]).

One should notice that there is a parameter set for each sub-band k, $1 \leq k \leq K$, hence there are K models that need be trained in our case. It follows that (12) writes:

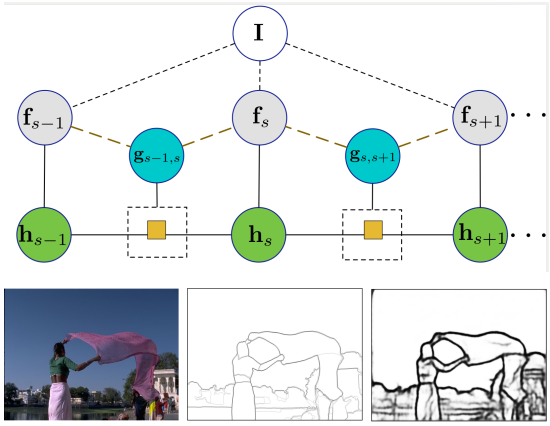
$$p(\boldsymbol{g}_{tk} | \boldsymbol{x}_{tn}, B_{tk} = n, C_{tk} = r) = \tag{V.38}$$
$$\begin{cases} \mathcal{N}(\boldsymbol{g}_{tk}; \mathbf{L}_{kr}\boldsymbol{x}_{tn} + \boldsymbol{l}_{kr}, \boldsymbol{\Sigma}_{kr}) & \text{if } 1 \leq n \leq N \\ \mathcal{U}(\boldsymbol{g}_{tk}; \text{vol}(\mathcal{G})) & \text{if } n = 0. \end{cases}$$

The right-hand side of (7) can now be written as:

$$p(C_{tk} = r | \boldsymbol{x}_{tn}, B_{tk} = n) = \frac{\pi_r \mathcal{N}(\boldsymbol{x}_{tn}; \boldsymbol{\nu}_r, \boldsymbol{\Omega}_r)}{\sum_{i=1}^{R} \pi_i \mathcal{N}(\boldsymbol{x}_{tn}; \boldsymbol{\nu}_i, \boldsymbol{\Omega}_i)}. \tag{V.39}$$

Chapter *VI*

Conditional Random Fields for Deep Pixel-Level Inference



Abstract *Recent works have shown that exploiting multi-scale representations deeply learned via convolutional neural networks (CNN) is of tremendous importance for accurate contour detection. This chapter presents a novel approach for predicting contours which advances the state of the art in two fundamental aspects, i.e. multi-scale feature generation and fusion. Different from previous works directly considering multi-scale feature maps obtained from the inner layers of a primary CNN architecture, we introduce a hierarchical deep model which produces more rich and complementary representations. Furthermore, to refine and robustly fuse the representations learned at different scales, the novel Attention-Gated Conditional Random Fields (AG-CRFs) are proposed. The experiments ran on two publicly available datasets (BSDS500 and NYUDv2) demonstrate the effectiveness of the latent AG-CRF model and of the overall hierarchical framework.*

Chapter Pitch

Methodological contribution A probabilistic model based on conditional random fields, that incorporates attention under the form of binary gates. The posterior probability of these gates is computed using the sigmoid function σ on the a posteriori expectation of the binary potential.

$$E = \sum_s \sum_i \phi_h(\mathbf{h}_s^i, \mathbf{f}_s^i) + \underbrace{\sum_{s',s} \sum_{i,j} g_{s',s}^i \psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j)}_{\text{Gated pairwise potential}} \quad \Rightarrow \quad \mathbb{E}\{g_{s',s}^i\} = \sigma\left(-\mathbb{E}_{q(\mathbf{h}_s^i)}\left\{\sum_j \mathbb{E}_{q(\mathbf{h}_{s'}^j)}\left\{\psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j)\right\}\right\}\right).$$

Applicative task Pixel-level inference tasks, in particular contour detection.

Interesting insight The gating formalism allows to model when the information from a certain random variable has to be used when inferring a neighboring variable, which conceptually corresponds to the attention mechanism in deep networks. The probabilistic formulation allows to formally interpret the a posteriori distribution of the gating variables as attention tensors and implement the whole inference procedure within the deep neural network framework.

Dissemination The attention-gated CRF was published in Advances in Neural Information Processing Systems [153].

VI.1 Introduction

Considered as one of the fundamental tasks in low-level vision, contour detection has been deeply studied in the past decades. While early works mostly focused on low-level cues (*e.g.* colors, gradients, textures) and hand-crafted features [154]–[156], more recent methods benefit from the representational power of deep learning models [157]–[161]. The ability to effectively exploit multi-scale feature representations is considered a crucial factor for achieving accurate predictions of contours in both traditional [162] and CNN-based [159]–[161] approaches. Restricting the attention on deep learning-based solutions, existing methods [159], [161] typically derive multi-scale representations by adopting standard CNN architectures and considering directly the feature maps associated to different inner layers. These maps are highly complementary: while the features from the first layers are responsible for predicting fine details, the ones from the higher layers are devoted to encode the basic structure of the objects. Traditionally, concatenation and weighted averaging are very popular strategies to combine multi-scale representations (see Fig. VI.1.a). While these strategies typically lead to an increased detection accuracy with respect to single-scale models, they severly simplify the complex relationship between multi-scale feature maps.

The motivational cornerstone of this study is the following research question: is it worth modeling and exploiting complex relationships between multiple scales of a deep representation for contour detection? In order to provide an answer and inspired by recent works exploiting graphical models within deep learning architectures [163], [164], we introduce *Attention-Gated Conditional Random Fields* (AG-CRFs), which allow to learn robust feature map representations at each scale by exploiting the information available from other scales. This is achieved by incorporating an attention mechanism [165] seamlessly integrated into the multi-scale learning process under the form of gates [166]. Intuitively, the attention mechanism will further enhance the quality of the learned multi-scale representation, thus improving the overall performance of the model.

We integrated the proposed AG-CRFs into a two-level hierarchical CNN model, defining a novel Attention-guided Multi-scale Hierarchical deepNet (AMH-Net) for contour detection. The hierarchical network is able to learn richer multi-scale features than conventional CNNs, the representational power of which is further enhanced by the proposed AG-CRF model. We evaluate the effectiveness of the overall model on two publicly available datasets for the contour detection task, *i.e.* BSDS500 [167] and NYU Depth v2 [168]. The results demonstrate that our approach is able to learn rich and complementary features, thus outperforming state-of-the-art contour detection methods.

VI.2 Related work

In the last few years several deep learning models have been proposed for detecting contours [157]–[159], [161], [169], [170]. Among these, some works explicitly focused on devising multi-scale CNN models in order to boost performance. For instance, the Holistically-Nested Edge Detection method [159] employed multiple side outputs derived from the inner layers of a primary CNN and combine them for the final prediction. Liu *et. al.* [170] introduced a framework to learn rich deep representations by concatenating features derived from all convolutional layers of VGG16. Bertasius *et. al.* [158] considered skip-layer CNNs to jointly combine feature maps from multiple layers. Maninis *et. al.* [161] proposed Convolutional Oriented Boundaries (COB), where features from different layers are fused to compute oriented contours and region hierarchies. However, these works combine the multi-scale representations from different layers adopting concatenation and weighted averaging schemes while not considering the dependency between the features. Furthermore, these works do not focus on generating more rich and diverse representations at each CNN layer.

The combination of multi-scale representations has been also widely investigated for other pixel-level prediction tasks, such as semantic segmentation [171], visual saliency detection [172] and monocular depth estimation [164], pedestrian detection [153] and different deep architectures have been designed. For instance, to effectively aggregate the multi-scale information, Yu *et. al.* [171] introduced dilated convolutions. Yang *et. al.* [173] proposed DAG-CNNs where multi-scale feature outputs from different ReLU layers are combined through element-wise addition operator. However, none of these works incorporate an attention mechanism into a multi-scale structured feature learning framework.

Attention models have been successfully exploited in deep learning for various tasks such as image classification [174], speech recognition [175] and image caption generation [176]. However, to our knowledge, this work is the first to introduce an attention model for estimating contours. Furthermore, we are not aware of previous studies integrating the attention mechanism into a probabilistic (CRF) framework to control the message passing between hidden variables. We model the attention as *gates* [166], which have been used in previous deep models such as restricted Boltzman machine for unsupervised feature learning [177], LSTM for sequence learning [178], [179] and CNN for image classification [180]. However, none of these works explore the possibility of jointly learning multi-scale deep representations and an attention model within a unified probabilistic graphical model.

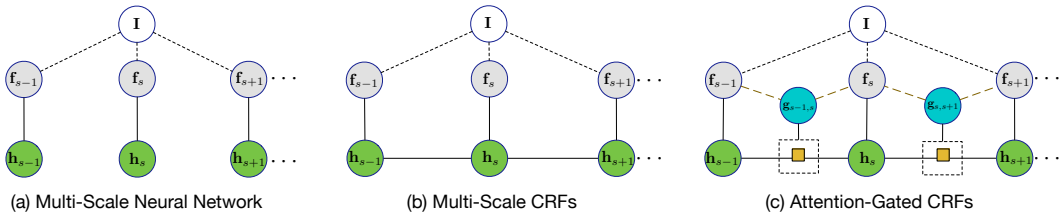


Figure VI.1: *An illustration of different schemes for multi-scale deep feature learning and fusion. (a) the traditional approach (e.g. concatenation, weighted average), (b) CRF implementing multi-scale feature fusion (c) the proposed AG-CRF-based approach.*

VI.3 Attention-Gated CRFs for Deep Structured Multi-Scale Feature Learning

VI.3.1 Problem Definition and Notation

Given an input image $\mathbf{I}$ and a generic front-end CNN model with parameters $\mathbf{W}_c$, we consider a set of S multi-scale feature maps $\mathbf{F} = \{\mathbf{f}_s\}_{s=1}^{S}$. Being a generic framework, these feature maps can be the output of S intermediate CNN layers or of another representation, thus s is a *virtual* scale. The feature map at scale s, $\mathbf{f}_s$ can be interpreted as a set of feature vectors, $\mathbf{f}_s = \{\mathbf{f}_s^i\}_{i=1}^{N}$, where N is the number of pixels. Opposite to previous works adopting simple concatenation or weighted averaging schemes [159], [181], we propose to combine the multi-scale feature maps by learning a set of latent feature maps $\mathbf{h}_s = \{\mathbf{h}_s^i\}_{i=1}^{N}$ with a novel *Attention-Gated CRF* model sketched in Fig.VI.1. Intuitively, this allows a joint refinement of the features by flowing information between different scales. Moreover, since the information from one scale may or may not be relevant for the pixels at another scale, we utilise the concept of *gate*, previously introduced in the literature in the case of graphical models [182], in our CRF formulation. These gates are binary random hidden variables that permit or block the flow of information between scales at every pixel. Formally, $g_{s_e,s_r}^i \in \{0,1\}$ is the gate at pixel i of scale s_r (receiver) from scale s_e (emitter), and we also write $\mathbf{g}_{s_e,s_r} = \{g_{s_e,s_r}^i\}_{i=1}^{N}$. Precisely, when $g_{s_e,s_r}^i = 1$ then the hidden variable $\mathbf{h}_{s_r}^i$ is updated taking (also) into account the information from the s_e-th layer, *i.e.* $\mathbf{h}_{s_e}$. As shown in the following, the joint inference of the hidden features and the gates leads to estimating the optimal features as well as the corresponding attention model, hence the name Attention-Gated CRFs.

VI.3.2 Attention-Gated CRFs

Given the observed multi-scale feature maps $\mathbf{F}$ of image $\mathbf{I}$, the objective is to estimate the hidden multi-scale representation $\mathbf{H} = \{\mathbf{h}_s\}_{s=1}^{S}$ and, accessorily the attention gate variables $\mathbf{G} = \{\mathbf{g}_{s_e,s_r}\}_{s_e,s_r=1}^{S}$. To do that, we formalize the problem within a conditional random field framework and write the Gibbs distribution as $P(\mathbf{H},\mathbf{G}|\mathbf{I},\Theta) = \exp\left(-E(\mathbf{H},\mathbf{G},\mathbf{I},\Theta)\right)/Z\left(\mathbf{I},\Theta\right)$, where Θ is the set of parameters and E is the energy function. As usual, we exploit both unary and binary potentials to couple the hidden variables between them and to the observations. Importantly, the proposed binary potential is gated, and thus only active when the gate is open. More formally the general form[VI.1] of the energy function writes:

$$E(\mathbf{H},\mathbf{G},\mathbf{I},\Theta) = \underbrace{\sum_s \sum_i \phi_h(\mathbf{h}_s^i, \mathbf{f}_s^i)}_{\text{Unary potential}} + \underbrace{\sum_{s_e,s_r} \sum_{i,j} g_{s_e,s_r}^i \psi_h(\mathbf{h}_{s_r}^i, \mathbf{h}_{s_e}^j)}_{\text{Gated pairwise potential}}. \tag{VI.1}$$

The first term of the energy function is a classical unary term that relates the hidden features to the observed multi-scale CNN representations. The second term synthesizes the theoretical contribution of the present study because it conditions the effect of the pair-wise potential $\psi_h(\mathbf{h}_{s_e}^i, \mathbf{h}_{s_r}^j)$ upon the gate hidden variable g_{s_e,s_r}^i. Fig. VI.1c depicts the model formulated in Equ.(VI.1). If we remove the attention gate variables, it becomes a general multi-scale CRFs as shown in Fig. VI.1b.

Given that formulation, and as it is typically the case in conditional random fields, we exploit the mean-field approximation in order to derive a tractable inference procedure. Under this generic form, the mean-field inference procedure

[VI.1] One could certainly include a unary potential for the gate variables as well. However this would imply that there is a way to set/learn the a priori distribution of opening/closing a gate. In practice we did not observe any notable difference between using or skipping the unary potential on g.

writes:

$$q(\mathbf{h}_s^i) \propto \exp\Big(\phi_h(\mathbf{h}_s^i, \mathbf{f}_s^i) + \sum_{s' \neq s} \sum_j \mathbb{E}_{q(g^i_{s',s})}\{g^i_{s',s}\} \mathbb{E}_{q(\mathbf{h}^j_{s'})}\{\psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j)\}\Big), \tag{VI.2}$$

$$q(g^i_{s',s}) \propto \exp\Big(g^i_{s',s} \mathbb{E}_{q(\mathbf{h}^i_s)}\Big\{ \sum_j \mathbb{E}_{q(\mathbf{h}^j_{s'})}\Big\{\psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j)\Big\}\Big\}\Big), \tag{VI.3}$$

where $\mathbb{E}_q$ stands for the expectation with respect to the distribution q.

Before deriving these formulae for our precise choice of potentials, we remark that, since the gate is a binary variable, the expectation of its value is the same as $q(g^i_{s',s} = 1)$. By defining: $\mathcal{M}^i_{s',s} = \mathbb{E}_{q(\mathbf{h}^i_s)}\Big\{\sum_j \mathbb{E}_{q(\mathbf{h}^j_{s'})}\Big\{\psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j)\Big\}\Big\}$, the expected value of the gate writes:

$$\alpha^i_{s,s'} = \mathbb{E}_{q(g^i_{s',s})}\{g^i_{s',s}\} = \frac{q(g^i_{s',s} = 1)}{q(g^i_{s',s} = 0) + q(g^i_{s',s} = 1)} = \sigma\left(-\mathcal{M}^i_{s',s}\right), \tag{VI.4}$$

where $\sigma()$ denotes the sigmoid function. This finding is specially relevant in the framework of CNN since many of the attention models are typically obtained after applying the sigmoid function to the features derived from a feed-forward network. Importantly, since the quantity $\mathcal{M}^i_{s',s}$ depends on the expected values of the hidden features $\mathbf{h}^i_s$, the AG-CRF framework extends the unidirectional connection from the features to the attention model, to a bidirectional connection in which the expected value of the gate allows to refine the distribution of the hidden features as well.

VI.3.3 AG-CRF Inference

In order to construct an operative model we need to define the unary and gated potentials ϕ_h and ψ_h. In our case, the unary potential corresponds to an isotropic Gaussian:

$$\phi_h(\mathbf{h}_s^i, \mathbf{f}_s^i) = -\frac{a^i_s}{2}\|\mathbf{h}^i_s - \mathbf{f}^i_s\|^2, \tag{VI.5}$$

where $a^i_s > 0$ is a weighting factor.

The gated binary potential is specifically designed for a two-fold objective. On the one hand, we would like to learn and further exploit the relationships between hidden vectors at the same, as well as at different scales. On the other hand, we would like to exploit previous knowledge on attention models and include linear terms in the potential. Indeed, this would implicitly shape the gate variable to include a linear operator on the features. Therefore, we chose a bilinear potential:

$$\psi_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j) = \tilde{\mathbf{h}}^i_s \mathbf{K}^{i,j}_{s,s'} \tilde{\mathbf{h}}^j_{s'}, \tag{VI.6}$$

where $\tilde{\mathbf{h}}^i_s = (\mathbf{h}^{i\top}_s, 1)^\top$ and $\mathbf{K}^{i,j}_{s,s'} \in \mathbb{R}^{(C_s+1)\times(C_{s'}+1)}$ being C_s the size, i.e. the number of channels, of the representation at scale s. If we write this matrix as $\mathbf{K}^{i,j}_{s,s'} = (\mathbf{L}^{i,j}_{s,s'}, \mathbf{l}^{i,j}_{s,s'}; \mathbf{l}^{j,i\top}_{s',s}, 1)$, then $\mathbf{L}^{i,j}_{s,s'}$ exploits the relationships between hidden variables, while $\mathbf{l}^{i,j}_{s,s'}$ and $\mathbf{l}^{j,i}_{s',s}$ implement the classically used linear relationships of the attention models. In order words, ψ_h models the pair-wise relationships between features with the upper-left block of the matrix. Furthemore, ψ_h takes into account the linear relationships by completing the hidden vectors with the unity. In all, the energy function writes:

$$E(\mathbf{H}, \mathbf{G}, \mathbf{I}, \Theta) = -\sum_s \sum_i \frac{a^i_s}{2}\|\mathbf{h}^i_s - \mathbf{f}^i_s\|^2 + \sum_{s_e, s_r} \sum_{i,j} g^i_{s_e,s_r} \tilde{\mathbf{h}}^i_{s_r} \mathbf{K}^{i,j}_{s_r,s_e} \tilde{\mathbf{h}}^j_{s_e}. \tag{VI.7}$$

Under these potentials, we can consequently update the mean-field inference equations to:

$$q(\mathbf{h}^i_s) \propto \exp\Big(-\frac{a^i_s}{2}(\|\mathbf{h}^i_s\| - 2\mathbf{h}^{i\top}_s \mathbf{f}^i_s) + \sum_{s' \neq s} \alpha^i_{s,s'} \mathbf{h}^{i\top}_s \sum_j (\mathbf{L}^{i,j}_{s,s'} \bar{\mathbf{h}}^j_{s'} + \mathbf{l}^{i,j}_{s,s'})\Big), \tag{VI.8}$$

where $\bar{\mathbf{h}}^j_{s'}$ is the expected a posteriori value of $\mathbf{h}^j_{s'}$.

The previous expression implies that the a posteriori distribution for $\mathbf{h}^i_s$ is a Gaussian. The mean vector of the Gaussian and the function $\mathcal{M}$ write:

$$\bar{\mathbf{h}}^i_s = \frac{1}{a^i_s}\Big(a^i_s \mathbf{f}^i_s + \sum_{s' \neq s} \alpha^i_{s,s'} \sum_j (\mathbf{L}^{i,j}_{s,s'} \bar{\mathbf{h}}^j_{s'} + \mathbf{l}^{i,j}_{s,s'})\Big) \quad \mathcal{M}^i_{s',s} = \sum_j \Big(\bar{\mathbf{h}}^i_s \mathbf{L}^{i,j}_{s,s'} \bar{\mathbf{h}}^j_{s'} + \bar{\mathbf{h}}^{i\top}_s \mathbf{l}^{i,j}_{s,s'} + \bar{\mathbf{h}}^{j\top}_{s'} \mathbf{l}^{j,i}_{s',s}\Big)$$

which concludes the inference procedure. Furthermore, the proposed framework can be simplified to obtain the traditional attention models. In most of the previous studies, the attention variables are computed directly from the multi-scale features instead of computing them from the hidden variables. Indeed, since many of these studies do not

propose a probabilistic formulation, there are no hidden variables and the attention is computed sequentially through the scales. We can emulate the same behavior within the AG-CRF framework by modifying the gated potential as follows:

$$\tilde{\psi}_h(\mathbf{h}_s^i, \mathbf{h}_{s'}^j, \mathbf{f}_s^i, \mathbf{f}_{s'}^j) = \mathbf{h}_s^i \mathbf{L}_{s,s'}^{i,j} \mathbf{h}_{s'}^j + \mathbf{f}_s^{i\top} \mathbf{l}_{s,s'}^{i,j} + \mathbf{f}_{s'}^{j\top} \mathbf{l}_{s',s}^{j,i}. \tag{VI.9}$$

This means that we keep the pair-wise relationships between hidden variables (as in any CRF) and let the attention model be generated by a linear combination of the observed features from the CNN, as it is traditionally done. The changes in the inference procedure are straightforward and reported in the supplementary material due to space constraints. We refer to this model as partially-latent AG-CRFs (PLAG-CRFs), whereas the more general one is denoted as fully-latent AG-CRFs (FLAG-CRFs).

VI.3.4 Implementation with neural network for joint learning

In order to infer the hidden variables and learn the parameters of the AG-CRFs together with those of the front-end CNN, we implement the AG-CRFs updates in neural network with several steps: (i) message passing from the s_e-th scale to the current s_r-th scale is performed with $\mathbf{h}_{s_e \to s_r} \leftarrow \mathbf{L}_{s_e \to s_r} \otimes \mathbf{h}_{s_e}$, where $\otimes$ denotes the convolutional operation and $\mathbf{L}_{s_e \to s_r}$ denotes the corresponding convolution kernel, (ii) attention map estimation $q(\mathbf{g}_{s_e,s_r} = \mathbf{1}) \leftarrow \sigma(\mathbf{h}_{s_r} \odot (\mathbf{L}_{s_e \to s_r} \otimes \mathbf{h}_{s_e}) + \mathbf{l}_{s_e \to s_r} \otimes \mathbf{h}_{s_e} + \mathbf{l}_{s_r \to s_e} \otimes \mathbf{h}_{s_r})$, where $\mathbf{L}_{s_e \to s_r}$, $\mathbf{l}_{s_e \to s_r}$ and $\mathbf{l}_{s_r \to s_e}$ are convolution kernels and $\odot$ represents element-wise product operation, and (iii) attention-gated message passing from other scales and adding unary term: $\bar{\mathbf{h}}_{s_r} = \mathbf{f}_{s_r} \oplus a_{s_r} \sum_{s_e \neq s_r} (q(\mathbf{g}_{s_e,s_r} = \mathbf{1}) \odot \mathbf{h}_{s_e \to s_r})$, where a_{s_r} encodes the effect of the $a_{s_r}^i$ for weighting the message and can be implemented as a 1×1 convolution. The symbol $\oplus$ denotes element-wise addition. In order to simplify the overall inference procedure, and because the magnitude of the linear term of ψ_h is in practice negligible compared to the quadratic term, we discard the message associated to the linear term. When the inference is complete, the final estimate is obtained by convolving all the scales.

VI.4 Exploiting AG-CRFs with a Multi-scale Hierarchical Network

VI.4.1 AMH-Net Architecture.

The proposed Attention-guided Multi-scale Hierarchical Network (AMH-Net), as sketched in Figure VI.2, consists of a multi-scale hierarchical network (MH-Net) together with the AG-CRF model described above. The MH-Net is constructed from a front-end CNN architecture such as the widely used AlexNet [81], VGG [183] and ResNet [184]. One prominent feature of MH-Net is its ability to generate richer multi-scale representations. In order to do that, we perform distinct non-linear mappings (deconvolution $\mathbf{D}$, convolution $\mathbf{C}$ and max-pooling $\mathbf{M}$) upon $\mathbf{f}_l$, the CNN feature representation from an intermediate layer l of the front-end CNN. This leads to a three-way representation: $\mathbf{f}_l^{\mathbf{D}}$, $\mathbf{f}_l^{\mathbf{C}}$ and $\mathbf{f}_l^{\mathbf{M}}$. Remarkably, while $\mathbf{D}$ upsamples the feature map, $\mathbf{C}$ maintains its original size and $\mathbf{M}$ reduces it, and different kernel size is utilized for them to have different receptive fields, then naturally obtaining complementary inter- and multi-scale representations. The $\mathbf{f}_l^{\mathbf{C}}$ and $\mathbf{f}_l^{\mathbf{M}}$ are further aligned to the dimensions of the feature map $\mathbf{f}_l^{\mathbf{D}}$ by the deconvolutional operation. The hierarchy is implemented in two levels. The first level uses an AG-CRF model to fuse the three representations of each layer l, thus refining the CNN features within the same scale. The second level of the hierarchy uses an AG-CRF model to fuse the information coming from multiple CNN layers. The proposed hierarchical multi-scale structure is general purpose and able to involve an arbitrary number of layers and of diverse intra-layer representations.

VI.4.2 End-to-End Network Optimization.

The parameters of the model consist of the front-end CNN parameters, $\mathbf{W}_c$, the parameters to produce the richer decomposition from each layer l, $\mathbf{W}_l$, the parameters of the AG-CRFs of the first level of the hierarchy, $\{\mathbf{W}_l^{\mathrm{I}}\}_{l=1}^{L}$, and the parameters of the AG-CRFs of the second level of the hierarchy, $\mathbf{W}^{\mathrm{II}}$. L is the number of intermediate layers used from the front-end CNN. In order to jointly optimize all these parameters we adopt deep supervision [159] and we add an optimization loss associated to each AG-CRF module. In addition, since the contour detection problem is highly unbalanced, *i.e.* contour pixels are significantly less than non-contour pixels, we employ the modified cross-entropy loss function of [159]. Given a training data set $\mathcal{D} = \{(\mathbf{I}_p, \mathbf{E}_p)\}_{p=1}^{P}$ consisting of P RGB-contour groundtruth pairs, the loss function ℓ writes:

$$\ell(\mathbf{W}) = \sum_p \beta \sum_{e_p^k \in \mathbf{E}_p^+} \log \mathrm{P}(e_p^k = 1 | \mathbf{I}_p; \mathbf{W}) + (1-\beta) \sum_{e_p^k \in \mathbf{E}_p^-} \log \mathrm{P}(e_p^k = 0 | \mathbf{I}_p; \mathbf{W}), \tag{VI.10}$$

where $\beta = |\mathbf{E}_p^+| / (|\mathbf{E}_p^+| + |\mathbf{E}_p^-|)$, $\mathbf{E}_p^+$ is the set of contour pixels of image p and $\mathbf{W}$ is the set of all parameters. The optimization is performed via the back-propagation algorithm with standard stochastic gradient descent.

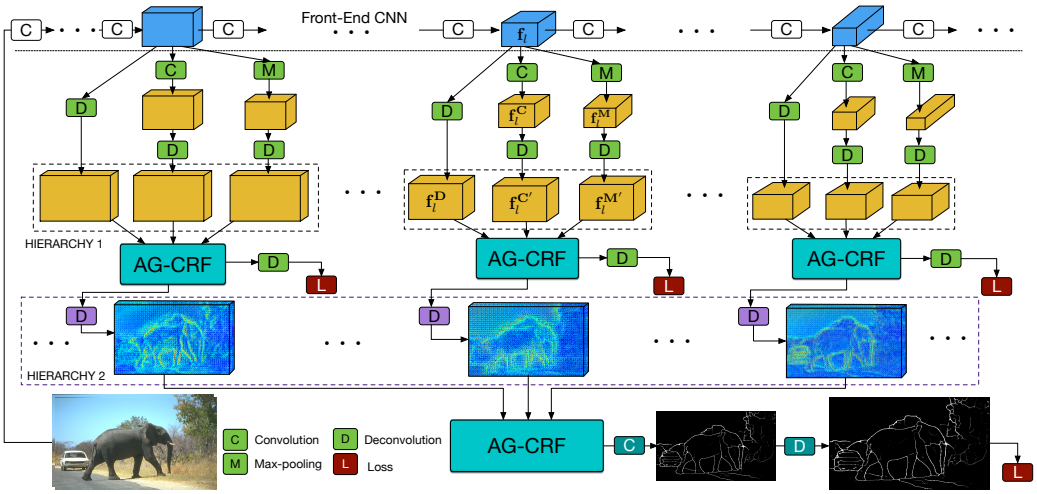


Figure VI.2: *An overview of the proposed AMH-Net for contour detection.*

VI.4.3 AMH-Net for contour detection.

After training of the whole AMH-Net, the optimized network parameters $\mathbf{W}$ are used for the contour detection task. Given a new test image $\mathbf{I}$, the $L+1$ classifiers produce a set of contour prediction maps $\{\hat{\mathbf{E}}_l\}_{l=1}^{L+1} = \text{AMH-Net}(\mathbf{I}; \mathbf{W})$. The $\hat{\mathbf{E}}_l$ are obtained from the AG-CRFs with elementary operations as detailed in the supplementary material. We inspire from [159] to fuse the multiple scale predictions thus obtaining an average prediction $\hat{\mathbf{E}} = \sum_l \hat{\mathbf{E}}_l/(L+1)$.

VI.5 Experiments

VI.5.1 Experimental Setup

Datasets To evaluate the proposed approach we employ two different benchmarks: the BSDS500 and the NYUDv2 datasets. The BSDS500 dataset is an extended dataset based on BSDS300 [167]. It consists of 200 training, 100 validation and 200 testing images. The groundtruth pixel-level labels for each sample are derived considering multiple annotators. Following [159], [169], we use all the training and validation images for learning the proposed model and perform data augmentation as described in [159]. The NYUDv2 [168] contains 1449 RGB-D images and it is split into three subsets, comprising 381 training, 414 validation and 654 testing images. Following [159] in our experiments we employ images at full resolution (*i.e.* 560×425 pixels) both in the training and in the testing phases.

Evaluation Metrics During the test phase standard non-maximum suppression (NMS) [185] is first applied to produce thinned contour maps. We then evaluate the detection performance of our approach according to different metrics, including the F-measure at Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS) and the Average Precision (AP). The maximum tolerance allowed for correct matches of edge predictions to the ground truth is set to 0.0075 for the BSDS500 dataset, and to .011 for the NYUDv2 dataset as in previous works [159], [185], [186].

Implementation Details The proposed AMH-Net is implemented under the deep learning framework *Caffe* [187]. The implementation code is available on Github[VI.2]. The training and testing phase are carried out on an Nvidia Titan X GPU with 12GB memory. The ResNet50 network pretrained on ImageNet [188] is used to initialize the front-end CNN of AMH-Net. Due to memory constraints, our implementation only considers three scales, *i.e.* we generate multi-scale features from three different layers of the front-end CNN (*i.e.* *res3d*, *res4f*, *res5c*). In our CRF model we consider dependencies between all scales. Within the AG-CRFs, the kernel size for all convolutional operations is set to 3×3 with stride 1 and padding 1. To simplify the model optimization, the parameters $a^i_{s_r}$ are set as 0.1 for all scales during training. We choose this value as it corresponds to the best performance after cross-validation in the range $[0, 1]$. The initial learning rate is set to 1e-7 in all our experiments, and decreases 10 times after every 10k iterations. The total number of iterations for BSDS500 and NYUD v2 is 40k and 30k, respectively. The momentum and weight decay parameters are set to 0.9 and 0.0002, as in [159]. As the training images have different resolution, we need to set the batch size to 1, and for the sake of smooth convergence we updated the parameters only every 10 iterations.

[VI.2] https://github.com/danxuhk/AttentionGatedMulti-ScaleFeatureLearning

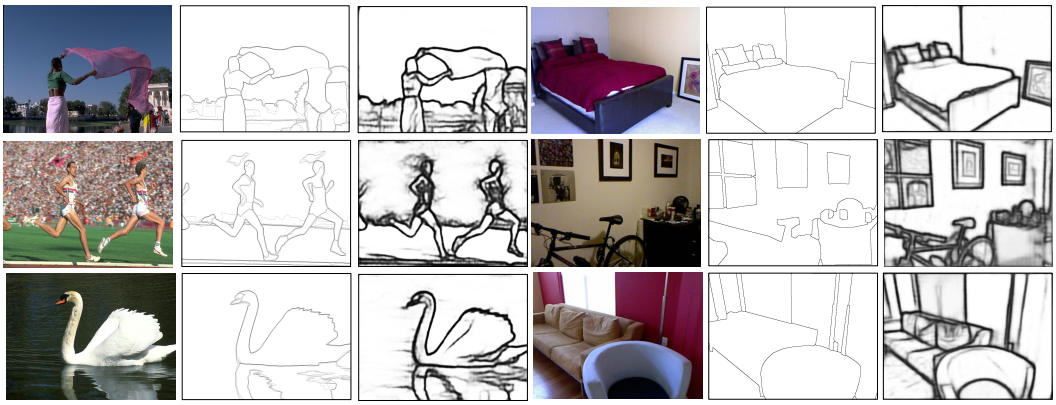


Figure VI.3: *Qualitative results on the BSDS500 (left) and the NYUDv2 (right) test samples. The 2nd (4th) and 3rd (6th) columns are the ground-truth and estimated contour maps respectively.*

Table VI.1: *BSDS500 dataset: quantitative results.*

Method	ODS	OIS	AP
Human	.800	.800	-
Felz-Hutt[189]	.610	.640	.560
Mean Shift[190]	.640	.680	.560
Normalized Cuts[49]	.641	.674	.447
ISCRA[191]	.724	.752	.783
gPb-ucm[167]	.726	.760	.727
Sketch Tokens[156]	.727	.746	.780
MCG[192]	.747	.779	.759
DeepEdge[158]	.753	.772	.807
DeepContour[157]	.756	.773	.797
LEP[193]	.757	.793	.828
HED[159]	.788	.808	.840
CEDN[169]	.788	.804	.834
COB [161]	.793	.820	.859
RCF [170] (not comp.)	.811	.830	–
AMH-Net (fusion)	**.798**	**.829**	**.869**

Table VI.2: *NYUDv2 dataset: quantitative results.*

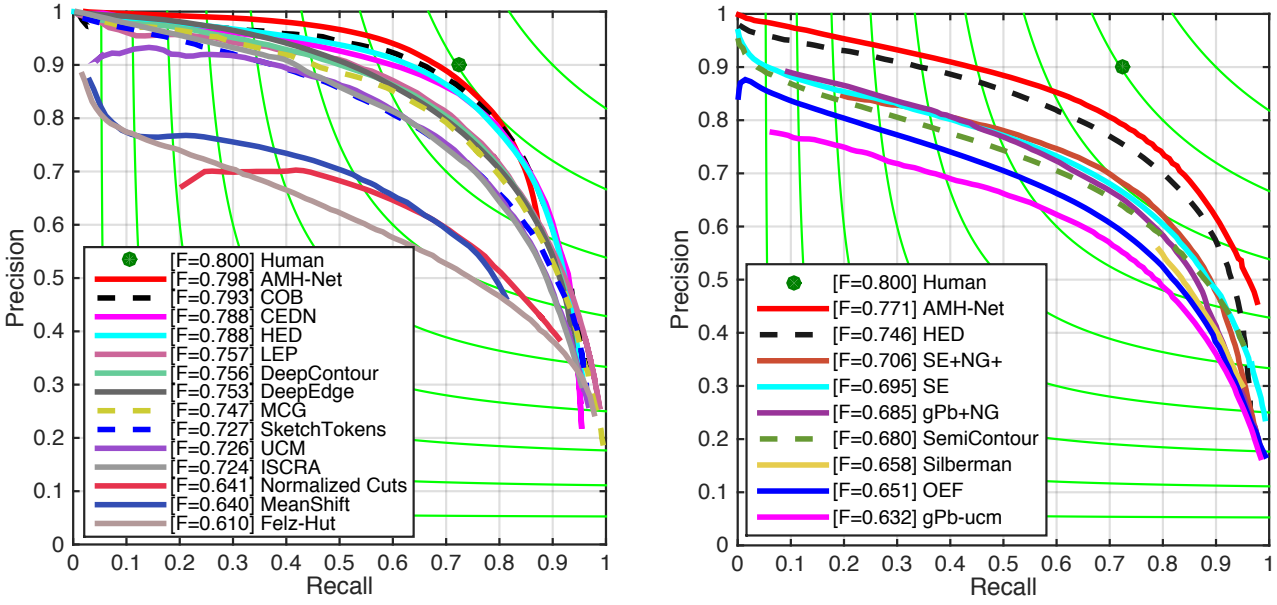
Method	ODS	OIS	AP
gPb-ucm [167]	.632	.661	.562
OEF [194]	.651	.667	–
Silberman *et. al.* [168]	.658	.661	–
SemiContour [195]	.680	.700	.690
SE [196]	.685	.699	.679
gPb+NG [197]	.687	.716	.629
SE+NG+ [186]	.710	.723	.738
HED (RGB) [159]	.720	.734	.734
HED (HHA) [159]	.682	.695	.702
HED (RGB + HHA) [159]	.746	.761	.786
RCF (RGB) + HHA) [170]	.757	.771	–
AMH-Net (RGB)	.744	.758	.765
AMH-Net (HHA)	.716	.729	.734
AMH-Net (RGB+HHA)	**.771**	**.786**	**.802**

VI.5.2 Experimental Results

In this section, we present the results of our evaluation, comparing our approach with several state of the art methods. We further conduct an in-depth analysis of our method, to show the impact of different components on the detection performance.

Comparison with state of the art methods. We first consider the BSDS500 dataset and compare the performance of our approach with several traditional contour detection methods, including Felz-Hut [189], MeanShift [190], Normalized Cuts [49], ISCRA [191], gPb-ucm [167], SketchTokens [156], MCG [192], LEP [193], and more recent CNN-based methods, including DeepEdge [158], DeepContour [157], HED [159], CEDN [169], COB [161]. We also report results of the RCF method [170], although they are not comparable because in [170] an extra dataset (Pascal Context) was used during RCF training to improve the results on BSDS500. In this series of experiments we consider AMH-Net with FLAG-CRFs. The results of this comparison are shown in Table VI.1 and Fig. VI.4a. AMH-Net obtains an F-measure (ODS) of 0.798, thus outperforms all previous methods. The improvement over the second and third best approaches, *i.e.* COB and HED, is 0.5% and 1.0%, respectively, which is not trivial to achieve on this challenging dataset. Furthermore, when considering the OIS and AP metrics, our approach is also better, with a clear performance gap.

To perform experiments on NYUDv2, following previous works [159] we consider three different types of input representations, *i.e.* RGB, HHA [186] and RGB-HHA data. The results corresponding to the use of both RGB and HHA data (*i.e.* RGB+HHA) are obtained by performing a weighted average of the estimates obtained from two AMH-Net models trained separately on RGB and HHA representations. As baselines we consider gPb-ucm [167], OEF [194], the



(a) *BSDS500 Data Set* **(b)** *NYUDv2 Data Set*

Figure VI.4: *Precision-Recall Curves of different methods on two data sets.*

Table VI.3: *Performance analysis on NYUDv2 RGB data.*

Method	ODS	OIS	AP
Hypercolumn [181]	.718	.729	.731
HED [159]	.720	.734	.734
AMH-Net (baseline)	.711	.720	.724
AMH-Net (w/o AG-CRFs)	.722	.732	.739
AMH-Net (w/ CRFs)	.732	.742	.750
AMH-Net (w/o deep supervision)	.725	.738	.747
AMH-Net (w/ PLAG-CRFs)	.737	.749	.746
AMH-Net (w/ FLAG-CRFs)	**.744**	**.758**	**.765**

method in [168], SemiContour [195], SE [196], gPb+NG [197], SE+NG+ [186], HED [159] and RCF [170]. In this case the results are comparable to the RCF [170] since the experimental protocol is exactly the same. All of them are reported in Table VI.2 and Fig. VI.4b. Again, our approach outperforms all previous methods. In particular, the increased performance with respect to HED [159] and RCF [170] confirms the benefit of the proposed multi-scale feature learning and fusion scheme. Examples of qualitative results on the BSDS500 and the NYUDv2 datasets are shown in Fig. VI.3. We show more examples of predictions from different multi-scale features on the BSDS500 dataset VI.5.

Ablation Study. To further demonstrate the effectiveness of the proposed model and analyze the impact of the different components of AMH-Net on the countour detection task, we conduct an ablation study considering the NYUDv2 dataset (RGB data). We tested the following models: (i) AMH-Net (baseline), which removes the first-level hierarchy and directly concatenates the feature maps for prediction, (ii) AMH-Net (w/o AG-CRFs), which employs the proposed multi-scale hierarchical structure but discards the AG-CRFs, (iii) AMH-Net (w/ CRFs), obtained by replacing our AG-CRFs with a multi-scale CRF model without attention gating, (iv) AMH-Net (w/o deep supervision) obtained removing intermediate loss functions in AMH-Net and (v) AMH-Net with the proposed two versions of the AG-CRFs model, *i.e.* PLAG-CRFs and FLAG-CRFs. The results of our comparison are shown in Table VI.3, where we also consider as reference traditional multi-scale deep learning models employing multi-scale representations, *i.e.* Hypercolumn [181] and HED [159].

These results clearly show the advantages of our contributions. The ODS F-measure of AMH-Net (w/o AG-CRFs) is 1.1% higher than AMH-Net (baseline), clearly demonstrating the effectiveness of the proposed hierarchical network and confirming our intuition that exploiting more richer and diverse multi-scale representations is beneficial. Table VI.3 also shows that our AG-CRFs plays a fundamental role for accurate detection, as AMH-Net (w/ FLAG-CRFs) leads to an improvement of 1.9% over AMH-Net (w/o AG-CRFs) in terms of OSD. Finally, AMH-Net (w/ FLAG-CRFs) is 1.2% and 1.5% better than AMH-Net (w/ CRFs) in ODS and AP metrics respectively, confirming the effectiveness of embedding

Figure VI.5: *Examples of predictions from different multi-scale features on BSDS500. The first column is the input test images. The 2nd to the 5nd columns show the predictions from different multi-scale features. The last column shows the final contour map after standard NMS.*

an attention mechanism in the multi-scale CRF model. AMH-Net (w/o deep supervision) decreases the overall performance of our method by 1.9% in ODS, showing the crucial importance of deep supervision for better optimization of the whole AMH-Net. Comparing the performance of the proposed two versions of the AG-CRF model, *i.e.* PLAG-CRFs and FLAG-CRFs, we can see that AMH-Net (FLAG-CRFs) slightly outperforms AMH-Net (PLAG-CRFs) in both ODS and OIS, while bringing a significant improvement (around 2%) in AP. Finally, considering HED [159] and Hypercolumn [181], it is clear that our AMH-Net (FLAG-CRFs) is significantly better than these methods. Importantly, our approach utilizes only three scales while for HED [159] and Hypercolumn [181] we consider five scales. We believe that our accuracy could be further boosted by involving more scales.

VI.6 Conclusions

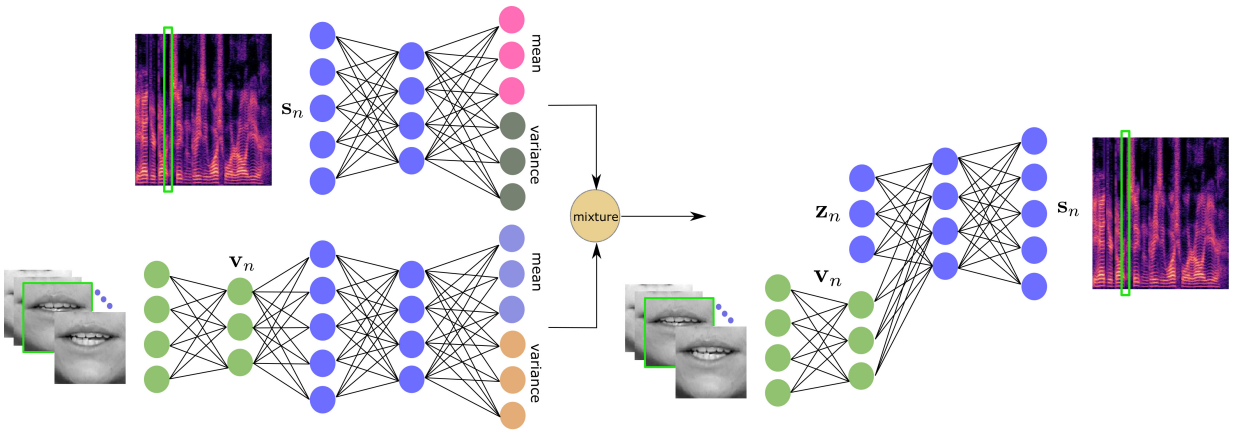
We presented a novel multi-scale hierarchical convolutional neural network for contour detection. The proposed model introduces two main components, *i.e.* a hierarchical architecture for generating more rich and complementary multi-scale feature representations, and an Attention-Gated CRF model for robust feature refinement and fusion. The effectiveness of our approach is demonstrated through extensive experiments on two public available datasets and state of the art detection performance is achieved. The proposed approach addresses a general problem, *i.e.*

how to generate rich multi-scale representations and optimally fuse them. Therefore, we believe it may be also useful for other pixel-level tasks.

Chapter *VII*

Variational Auto-Encoders for Audio-Visual Speech Enhancement

Abstract *We are interested in unsupervised (unknown noise) speech enhancement using latent variable generative models. We propose to learn generative models for clean speech spectrogram based on a variational autoencoder (VAE) where both auditory and visual information are used to infer the posterior of the latent variables. This is motivated by the fact that visual data, i.e. lips images of the speaker, provide helpful and complementary information about speech. As such, they can help train a richer inference network, where the audio and visual information are fused. We propose two different strategies, one systematically using audio and video, and a second one inferring the best combination of audio and video. Moreover, during speech enhancement, visual data are used to initialize the latent variables, thus providing a more robust initialization than using the noisy speech spectrogram. A variational inference approach is derived to train the proposed VAE. Thanks to the novel inference procedure and the robust initialization, the proposed audio-visual VAEs exhibit superior performance on speech enhancement than using the standard audio-only counterpart.*



Chapter Pitch

Methodological contribution Two audio-visual speech generative models based on variational autoencoders. One systematically employing both auditory and visual data, and a second one automatically infering the optimal balance between the two modalities.

Applicative task Audio-visual speech enhancement.

Interesting insight The non-linearity associated to the variational autoencoders is attractive because it leads to high representation power, but one must be careful about the additional computational burden. When dealing with additional hidden variables (*i.e.* mixing variables), one must rederive appropriate learning and inference algorithms, since the standard procedures for learning VAE are not sufficient anymore.

Dissemination The first method was published in IEEE Transactions on Audio, Speech and Language Processing [198] and the second is Submitted to IEEE Transactions on Signal Processing [199].

VII.1 Introduction

Speech enhancement, or removing background noise from noisy speech [200], [201], is a classic yet very important problem in signal processing and machine learning. Traditional solutions to this problem are based on spectral subtraction [202] and Wiener filtering [203], targeting noise and/or speech power spectral density (PSD) estimation in the short-time Fourier transform (STFT) domain. The recent impressive performance of deep neural networks (DNNs) in computer vision and machine learning has paved the way to revisit the speech enhancement problem. DNNs have been widely utilized in this regard, where a neural network is trained to map a noisy speech spectrogram to its clean version, or to a time frequency (TF) mask [204]–[206]. This is usually done in a supervised way, using a huge dataset of noise and clean speech signals for training. As such, the performance of a supervised speech enhancement technique often degrades when dealing with an unknown type of noise.

Unsupervised techniques offer another procedure for speech enhancement that does not use noise signals for training. A popular unsupervised method is based on nonnegative matrix factorization (NMF) [207]–[209] for modeling the PSD of speech signals [210], which decomposes PSD as a product of two non-negative low-rank matrices (a dictionary of basis spectra and the corresponding activations). An NMF-based speech enhancement method consists of first learning a set of basis spectra for clean speech spectrograms at training phase, prior to speech enhancement [208], [211], [212]. Then, by decomposing the noisy spectrogram as the sum of clean speech and noise spectrograms, the corresponding clean speech activations as well as the NMF parameters of noise are estimated. While being computationally efficient, this modeling and enhancement framework cannot properly explain complicated structure of speech spectrogram due to the limited representational power dictated by the two low-rank matrices. A deep autoencoder (DAE) has been employed in [213] to model clean speech and noise spectrograms. A DAE is pre-trained for clean speech spectrograms, while an extra DAE for noise spectrogram is trained at the enhancement stage using the noisy spectrogram. The corresponding inference problem is under-determined, and the authors proposed to constrain the unknown speech using a pre-trained NMF model. As such, this DAE-based method might encounter the same shortcomings as those of the NMF-based speech enhancement [214].

Deep latent variable models offer a more sophisticated and efficient modeling framework than NMF and DAE, gaining much interest over the past few years [214]–[219]. The first and main step is to train a generative model for clean speech spectrogram using a variational auto-encoder (VAE) [9], [220]. VAE provides an efficient way to estimate the parameters of a non-linear generative model, also called the decoder. This is done by approximating the intractable posterior distribution of the latent variables using a Gaussian distribution parametrized by a neural network, called the inference (encoder) network. The encoder and decoder are jointly trained to maximize a variational lower bound on the marginal data log-likelihood. At test time, the trained generative model is combined with a noise model, e.g. NMF. The unknown noise parameters and clean speech are then estimated from the observed noisy speech. Being independent of the noise type at training, these methods show better generalization than the supervised approaches [214], [215].

Although it has been shown that the fusion of visual and audio information is beneficial for various speech perception tasks, e.g. [221]–[223], audio-visual speech enhancement (AVSE) has been far less investigated than audio speech enhancement (ASE). AVSE methods can be traced back to [224] and subsequent work, e.g. [225]–[230]. Not surprisingly, AVSE has been recently addressed in the framework of deep neural networks (DNNs) and a number of interesting architectures and well-performing algorithms were developed, e.g. [231]–[235]. In this chapter, we propose to fuse single-channel audio and single-camera visual information for speech enhancement in the framework of VAEs. This may well be viewed as a multimodal extension of VAE-based methods of [214]–[218], [236] which, up to our knowledge, yield state-of-the-art ASE performance in an unsupervised learning setting. In order to incorporate visual observations into the VAE speech enhancement framework, we propose to investigate two strategies. First, to systematically combine both streams, formulating the problem via a conditional VAE – we name this strategy AV-VAE. Second, to automatically estimate the optimal balance between auditory and visual information, formulating the problem via a mixture of interence networks – we name this strategy MIN-VAE.

For both strategies, as in [215] we proceed in three steps. First, the parameters of the generative architectures are learned using synchronized clean audio-speech and visual-speech data. This yields an audio-visual speech prior model. The training is totally unsupervised, in the sense that speech signals mixed with various types of noise signal are not required. This stays in contrast with supervised DNN methods that need to be trained in the presence of many noise types and noise levels in order to ensure generalization and good performance, e.g. [231]–[233]. Second, the learned speech prior is used in conjunction with a mixture model and with a NMF noise variance model, to infer the noise NMF parameters. Third, the clean speech is reconstructed using the speech prior (VAE parameters) as well as the estimated noise variance model. The latter may well be viewed as a probabilistic Wiener filter. The learned VAE architecture and the proposed speech reconstruction methods are thoroughly tested and compared with the state-of-the-art method, using the NTCD-TIMIT dataset [237].

The rest of the chapter is structured as follows. Section VII.2 discusses the related work. Sections VII.3 and VII.4 discuss the audio-only VAE and the visual-only VAE, respectively. The AV-VAE is then discussed in Section VII.5,

and the associated speech enhancement is presented in Section VII.6. We then introduce the MIN-VAE model in Section VII.7, and the associated speech enhancement procedure in Section VII.8. The conducted experiments are presented in Section VII.9, and Section VII.10 concludes the chapter.

VII.2 Related Work

Speech enhancement has been an extremely investigated topic for the last decades and a complete state-of-the-art is beyond the scope of this chapter. We briefly review the literature on single-channel speech enhancement (SE) and then we discuss the most significant work in AVSE. Classical methods use spectral subtraction [202] and Wiener filtering [203] based on noise and/or speech PSD estimation in the STFT domain. Another popular family of methods is the short-term spectral amplitude estimator [238], initially based on a local complex-valued Gaussian model of the speech STFT coefficients and then extended to other density models [239], [240], and to a log-spectral amplitude estimator [241], [242]. A popular technique for modeling the PSD of speech signals [210] is NMF, e.g. [207], [208], [243].

More recently, SE has been addressed in the framework of DNNs [204]. Supervised methods learn mappings between noisy-speech and clean-speech spectrograms, which are then used to reconstruct a speech waveform [205], [244], [245]. Alternatively, the noisy input is mapped onto a TF mask, which is then applied to the input to remove noise and to preserve speech information as much as possible [246]–[248]. In order for these supervised learning methods to generalize well and to yield state-of-the-art results, the training data must contain a large variability in terms of speakers and, even more critically, in terms of noise types and noise levels [205], [246]; in practice this leads to cumbersome learning processes.

Alternatively, generative (or unsupervised) DNNs do not use any kind of noise information for training, and for this reason they are very interesting because they have very good generalization capabilities. An interesting generative formulation is provided by VAEs [8]. Combined with NMF, VAE-based methods yield state-of-the-art SE performance [214]–[218], [236] for an unsupervised learning setting. VAEs conditioned on the speaker identity have also been used for speaker-dependent multi-microphone speech separation [219], [249] and dereverberation [250].

The use of visual cues to complement audio, whenever the latter is noisy, ambiguous or incomplete, has been thoroughly studied in psychophysics [221]–[223]. Indeed, speech production implies simultaneous air circulation through the vocal tract and tongue and lip movements, and hence speech perception is multimodal. Several computational models were proposed to exploit the correlation between audio and visual information for the perception of speech, e.g. [226], [229]. A multi-layer perceptron architecture was proposed in [225] to map noisy-speech linear prediction features concatenated with visual features onto clean-speech linear prediction features. Then Wiener filters were built for denoising. Audio-visual Wiener filtering was later extended using phoneme-specific Gaussian mixture regression and filterbank audio features [251]. Other AVSE methods exploit noise-free visual information [227], [228] or make use of twin hidden Markov models (HMMs) [230].

State-of-the-art supervised AVSE methods are based on DNNs. The rationale of [231], [233] is to use visual information to predict a TF soft mask in the STFT domain and to apply this mask to the audio input in order to remove noise. In [233] a video-to-speech architecture is trained for each speaker in the dataset, which yields a speaker-dependent AVSE method. The architecture of [231] is composed of a magnitude subnetwork that takes both visual and audio data as inputs, and a phase subnetwork that only takes audio as input. Both subnetworks are trained using ground-truth clean speech. Then, the magnitude subnetwork predicts a binary mask which is then applied to both the magnitude and phase spectrograms of the input signal, thus predicting a filtered speech spectrogram. The architectures of [234] and [232] are quite similar: they are composed of two subnetworks, one for processing noisy speech and one for processing visual speech. The two encodings are then concatenated and processed to eventually obtain an enhanced speech spectrogram. The main difference between [234] and [232] is that the former predicts both enhanced visual and audio speech, while the latter predicts only audio speech. The idea of obtaining a binary mask for separating speech of an unseen speaker from an unknown noise was exploited in [235]: a hybrid DNN model integrates a stacked long short-term memory (LSTM) and convolutional LSTM for audio-visual (AV) mask estimation.

In the supervised deep learning methods just mentioned, generalization to unseen data is a critical issue. The major issues are noise and speaker variability. Therefore, training these methods requires noisy mixtures with a large number of noise types and speakers, in order to guarantee generalization. In comparison, the proposed method is totally unsupervised: its training is based on VAEs and it only requires clean audio speech and visual speech. The gain and the noise variance are estimated at test time using a Monte Carlo expectation-maximization (MCEM) algorithm [252]. The clean speech is then reconstructed from the audio and visual inputs using the learned parameters. The latter may well be viewed as a probabilistic Wiener filter. This stays in contrast with the vast majority of supervised DNN-based AVSE methods that predict a TF mask which is applied to the noisy input.

VII.3 Audio VAE

In this section, we briefly review the deep generative speech model that was first proposed in [214] along with its parameters estimation procedure using VAEs [8]. Let s_{fn} denote the complex-valued speech STFT coefficient at frequency index $f \in \{0, ..., F-1\}$ and at frame index n. At each TF bin, we have the following model which will be referred to as audio VAE (A-VAE):

$$s_{fn}|\mathbf{z}_n \sim \mathcal{N}_c(0, \sigma_f(\mathbf{z}_n)), \tag{VII.1}$$

$$\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{VII.2}$$

where $\mathbf{z}_n \in \mathbb{R}^L$, with $L \ll F$, is a latent random variable describing a speech generative process, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is a zero-mean multivariate Gaussian distribution with identity covariance matrix, and $\mathcal{N}_c(0, \sigma)$ is a univariate complex proper Gaussian distribution with zero mean and variance σ. Let $\mathbf{s}_n \in \mathbb{C}^F$ be the vector whose components are the speech STFT coefficients at frame n. The set of non-linear functions $\{\sigma_f : \mathbb{R}^L \mapsto \mathbb{R}_+\}_{f=0}^{F-1}$ are modeled as neural networks sharing the input $\mathbf{z}_n \in \mathbb{R}^L$. The parameters of these neural networks are collectively denoted by $\boldsymbol{\theta}$. This variance can be interpreted as a model for the short-term PSD of the speech signal.

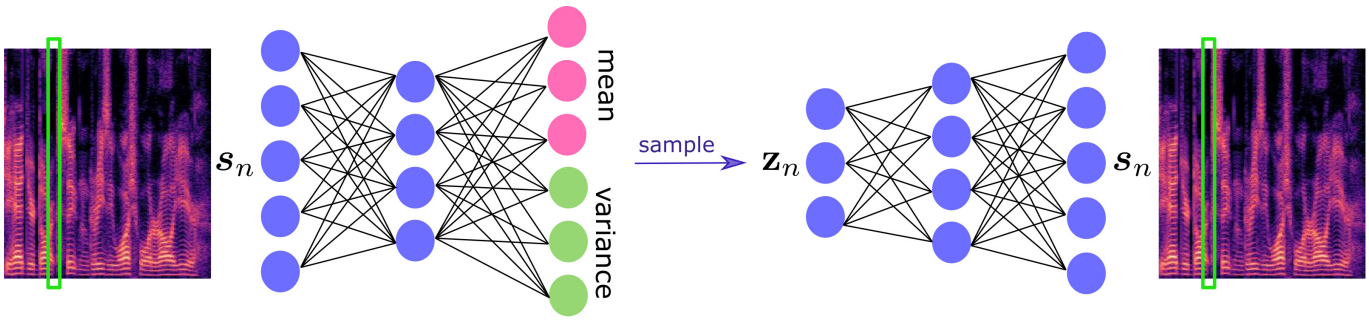


Figure VII.1: *The A-VAE network used for learning a speech prior using audio data. The encoder network (left) takes as input the squared magnitude vector $\tilde{\mathbf{s}}_n$, associated with the STFT frame $\mathbf{s}_n$ (outlined in green), and outputs the mean and variance of the posterior distribution $q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\psi})$. The decoder network (right) takes $\mathbf{z}_n$ as input (sampled from the posterior distribution) and outputs the variance of $p(\mathbf{s}_n|\mathbf{z}_n; \boldsymbol{\theta})$.*

An important property of VAEs is to provide an efficient way of learning the parameters $\boldsymbol{\theta}$ of such generative models [8], taking ideas from variational inference [253], [254]. Let $\mathbf{s} = \{\mathbf{s}_n \in \mathbb{C}^F\}_{n=0}^{N_{tr}-1}$ be a training dataset of clean-speech STFT frames and let $\mathbf{z} = \{\mathbf{z}_n \in \mathbb{R}^L\}_{n=0}^{N_{tr}-1}$ be the associated latent variables. In the VAE framework, the parameters $\boldsymbol{\theta}$ are estimated by maximizing a lower bound of the log-likelihood, $\ln p(\mathbf{s}; \boldsymbol{\theta})$, called evidence lower bound (ELBO), defined by:

$$\mathcal{L}(\mathbf{s}; \boldsymbol{\theta}, \boldsymbol{\psi}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{s};\boldsymbol{\psi})}\left[\ln p(\mathbf{s}|\mathbf{z}; \boldsymbol{\theta})\right] - D_{\mathsf{KL}}\left(q(\mathbf{z}|\mathbf{s}; \boldsymbol{\psi}) \parallel p(\mathbf{z})\right), \tag{VII.3}$$

where $q(\mathbf{z}|\mathbf{s}; \boldsymbol{\psi})$ denotes an approximation of the intractable true posterior distribution $p(\mathbf{z}|\mathbf{s}; \boldsymbol{\theta})$, $p(\mathbf{z})$ is the prior distribution of $\mathbf{z}$, and $D_{\mathsf{KL}}(q \parallel p) = \mathbb{E}_q[\ln(q/p)]$ is the Kullback-Leibler divergence. Independently, for all $l \in \{0, ..., L-1\}$ and all $n \in \{0, ..., N_{tr}-1\}$, $q(\mathbf{z}|\mathbf{s}; \boldsymbol{\psi})$ is defined by:

$$z_{ln}|\mathbf{s}_n \sim \mathcal{N}\left(\tilde{\mu}_l(\tilde{\mathbf{s}}_n), \tilde{\sigma}_l(\tilde{\mathbf{s}}_n)\right), \tag{VII.4}$$

where $\tilde{\mathbf{s}}_n \triangleq (|s_{0n}|^2 \dots |s_{F-1\,n}|^2)^\top$. The non-linear functions $\{\tilde{\mu}_l : \mathbb{R}_+^F \mapsto \mathbb{R}\}_{l=0}^{L-1}$ and $\{\tilde{\sigma}_l : \mathbb{R}_+^F \mapsto \mathbb{R}_+\}_{l=0}^{L-1}$ are modeled as neural networks, sharing as input the speech power spectrum frame $\tilde{\mathbf{s}}_n$, and collectively parameterized by $\boldsymbol{\psi}$. The parameter set $\boldsymbol{\psi}$ is also estimated by maximizing the *variational lower bound* defined in (VII.3), which is actually equivalent to minimizing the Kullback-Leibler divergence between $q(\mathbf{z}|\mathbf{s}; \boldsymbol{\psi})$ and the intractable true posterior distribution $p(\mathbf{z}|\mathbf{s}; \boldsymbol{\theta})$ [253]. Using (VII.1), (VII.2) and (VII.4) we can develop this objective function as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{s}; \boldsymbol{\theta}, \boldsymbol{\psi}) \stackrel{c}{=} &- \sum_{f=0}^{F-1} \sum_{n=0}^{N_{tr}-1} \mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\psi})}\left[d_{\mathsf{IS}}\left(|s_{fn}|^2; \sigma_f(\mathbf{z}_n)\right)\right] \\ &+ \frac{1}{2} \sum_{l=0}^{L-1} \sum_{n=0}^{N_{tr}-1} \left[\ln \tilde{\sigma}_l(\tilde{\mathbf{s}}_n) - \tilde{\mu}_l^2(\tilde{\mathbf{s}}_n) - \tilde{\sigma}_l(\tilde{\mathbf{s}}_n)\right],\end{aligned} \tag{VII.5}$$

where $d_{\mathsf{IS}}(x; y) = x/y - \ln(x/y) - 1$ is the Itakura-Saito divergence [210]. Finally, using sampling techniques combined with the so-called "reparametrization trick" [8] to approximate the intractable expectation in (VII.5), one obtains an objective function which is differentiable with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ and can be optimized using gradient-ascent algorithms [8]. The encoder-decoder architecture of the A-VAE speech prior is summarized in Figure VII.1.

VII.4 Visual VAE

We now introduce two VAE network variants for learning the speech prior from visual data, that will be referred to as *base* visual VAE (V-VAE) and *augmented* V-VAE, and which are summarized in Figure VII.2. As it can be seen, this architecture is similar to A-VAE, with the notable difference that it takes as input visual observations, namely lip images. In more detail, standard computer vision algorithms are used to extract a fixed-sized bounding-box from the image of a speaking face, with the lips in its center, i.e. a lip region of interest (ROI). This ROI is embedded into a visual feature vector $\mathbf{v}_n \in \mathbb{R}^M$ using a two-layer fully-connected network, referred below as the *base* network, where M is the dimension of the visual embedding. Optionally, one can use an additional pre-trained *front-end* network (dashed box) composed of a 3D convolution layer followed by a ResNet with 34 layers, as part of a network specifically trained for the task of supervised audio-visual speech recognition [255]. This second option is referred to as *augmented* V-VAE.

In variational inference [253], [254], any distribution over the latent variables $\mathbf{z}$ can be considered for approximating the intractable posterior $p(\mathbf{z}|\mathbf{s};\boldsymbol{\theta})$ and for defining the ELBO. For the V-VAE model, we explore the use of an approximate posterior distribution $q(\mathbf{z}|\mathbf{v};\boldsymbol{\gamma})$ defined by:

$$z_{ln}|\mathbf{v}_n \sim \mathcal{N}\left(\bar{\mu}_l(\mathbf{v}_n), \bar{\sigma}_l(\mathbf{v}_n)\right), \tag{VII.6}$$

where $\mathbf{v} = \{\mathbf{v}_n\}_{n=1}^{N_{tr}-1}$ is the training set of visual features, and where the non-linear functions $\{\bar{\mu}_l : \mathbb{R}^M \mapsto \mathbb{R}\}_{l=0}^{L-1}$ and $\{\bar{\sigma}_l : \mathbb{R}^M \mapsto \mathbb{R}_+\}_{l=0}^{L-1}$ are collectively modeled with a neural network parameterized by $\boldsymbol{\gamma}$ which takes $\mathbf{v}_n$ as input. Notice that V-VAE and A-VAE share the same decoder architecture, i.e. (VII.1). Eventually, the objective function of V-VAE has the same structure as (VII.5) and hence one can use the same gradient-ascent algorithm as above to estimate the parameters of the V-VAE network.

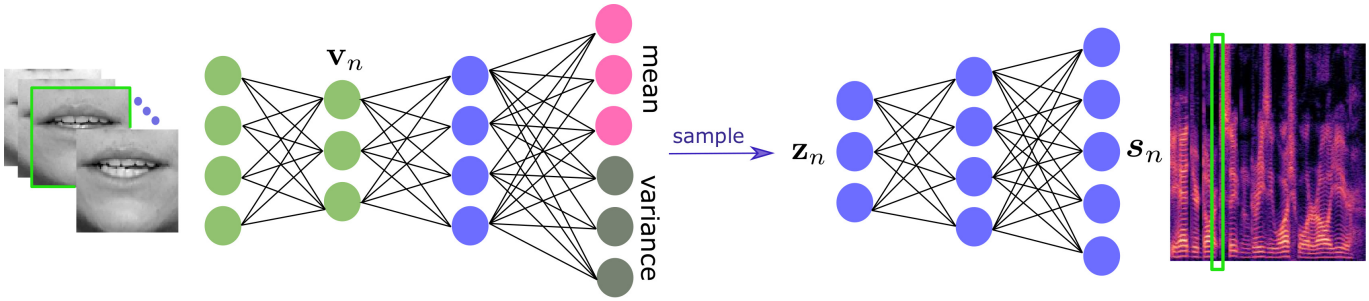


Figure VII.2: *The two V-VAE network variants (base and augmented) for learning speech prior from visual features. A lip ROI is embedded into a visual feature vector, denoted by $\mathbf{v}_n$, which is encoded and decoded using the same architecture and the same learning method as A-VAE. Optionally, one can also use a pre-trained network (dashed box) composed of a 3D convolution layer followed by a ResNet with 34 layers.*

VII.5 Audio-visual VAE

We now investigate an audio-visual VAE model, namely a model that combines audio speech with visual speech. The rationale behind this multimodal approach is that audio data are often corrupted by noise while visual data are not. Without loss of generality, it will be assumed that audio and visual data are synchronized, i.e. there is a video frame associated with each audio frame.

In order to combine the above A-VAE and V-VAE formulations, we consider the conditional variational auto-encoder (CVAE) framework to learn structured-output representations [256]. At training, a CVAE is provided with data as well as with associated class labels, such that the network is able to learn a structured data distribution. At test time, the trained network is provided with a class label to generate samples from the corresponding class. CVAEs have been proven to be very effective for missing-value inference problems, e.g. computer vision problems with partially available input-output pairs [256].

VII.5.1 The Generative Model

In the case of AV speech enhancement we consider a training set of N_{tr} synchronized frames of AV features, namely $(\mathbf{s}, \mathbf{v}) = \{\mathbf{s}_n, \mathbf{v}_n\}_{n=1}^{N_{tr}-1}$ where, as above, $\mathbf{v}_n \in \mathbb{R}^M$ is a lip ROI embedding. The clean audio speech, which is only available at training, is conditioned on the observed visual speech. The visual information is however available both at training and at testing, therefore it serves as a deterministic prior on the desired clean audio speech. Interestingly, it

also affects the prior distribution of $\mathbf{z}_n$. To summarize, the following latent space model is considered, independently for all $l \in \{0, ..., L-1\}$ and all TF bins (f, n):

$$s_{fn}|\mathbf{z}_n, \mathbf{v}_n \sim \mathcal{N}_c(0, \sigma_f(\mathbf{z}_n, \mathbf{v}_n)), \tag{VII.7}$$

$$z_{ln}|\mathbf{v}_n \sim \mathcal{N}\left(\bar{\mu}_l(\mathbf{v}_n), \bar{\sigma}_l(\mathbf{v}_n)\right), \tag{VII.8}$$

where the non-linear functions $\{\sigma_f : \mathbb{R}^L \times \mathbb{R}^M \mapsto \mathbb{R}_+\}_{f=0}^{F-1}$ are modeled as a neural network parameterized by $\boldsymbol{\theta}$ and taking $\mathbf{z}_n$ and $\mathbf{v}_n$ as input, and where (VII.8) is identical to (VII.6) but the corresponding parameter set $\boldsymbol{\gamma}$ will have different estimates, as explained below. Also, notice that σ_f in (VII.1) and in (VII.7) are different, but they both correspond to the PSD of the generative speech model. This motivates the abuse of notation that holds through the chapter. The proposed architecture is referred to as audio-visual VAE (AV-VAE) and is shown in Fig. VII.3. Compared to A-VAE of Section VII.3 and Figure VII.1, and with V-VAE of Section VII.4 and Figure VII.2, the mean and variance of the $\mathbf{z}_n$ prior distribution, are conditioned by visual inputs.

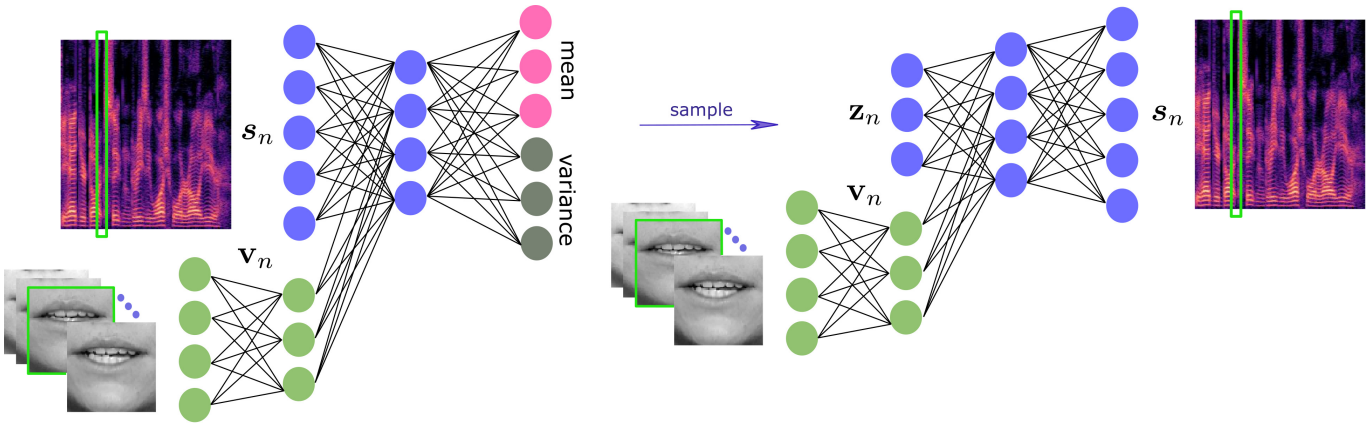


Figure VII.3: *Pipeline of the proposed AV-VAE architecture for learning an audio-visual speech prior for speech enhancement. The encoder takes a single frame of squared magnitude of speech's STFT, denoted by $\tilde{\mathbf{s}}_n$, as well as the corresponding visual feature vector $\mathbf{v}_n$, and outputs the parameters of the posterior distribution $q\left(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n; \boldsymbol{\psi}\right)$. The decoder network takes $\mathbf{z}_n$, sampled from the posterior distribution, together with $\mathbf{v}_n$ as input and outputs the variance of $p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta})$.*

VII.5.2 The Posterior Distribution

We now introduce the distribution $q\left(\mathbf{z}|\mathbf{s}, \mathbf{v}; \boldsymbol{\psi}\right)$, which approximates the intractable posterior distribution $p(\mathbf{z}|\mathbf{s}, \mathbf{v}; \boldsymbol{\theta})$, defined, as above, independently for all $l \in \{0, ..., L-1\}$ and all frames:

$$z_{ln}|\mathbf{s}_n, \mathbf{v}_n \sim \mathcal{N}\left(\tilde{\mu}_l\left(\tilde{\mathbf{s}}_n, \mathbf{v}_n\right), \tilde{\sigma}_l\left(\tilde{\mathbf{s}}_n, \mathbf{v}_n\right)\right), \tag{VII.9}$$

where the non-linear functions $\{\tilde{\mu}_l : \mathbb{R}_+^F \times \mathbb{R}^M \mapsto \mathbb{R}\}_{l=0}^{L-1}$ and $\{\tilde{\sigma}_l : \mathbb{R}_+^F \times \mathbb{R}^M \mapsto \mathbb{R}_+\}_{l=0}^{L-1}$ are collectively modeled as an encoder neural network, parameterized by $\boldsymbol{\psi}$, that takes as input the speech power spectrum and its associated visual feature vector, at each frame. The complete set of model parameters, i.e. $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, can be estimated by maximizing a lower bound of the conditional log-likelihood $\ln p(\mathbf{s}|\mathbf{v}; \boldsymbol{\theta}, \boldsymbol{\gamma})$ over the training dataset, defined by:

$$\mathcal{L}_{\text{av-cvae}}\left(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\gamma}\right) = \mathbb{E}_{q(\mathbf{z}|\mathbf{s},\mathbf{v};\boldsymbol{\psi})}\left[\ln p\left(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta}\right)\right] - D_{KL}\left(q\left(\mathbf{z}|\mathbf{s}, \mathbf{v}; \boldsymbol{\psi}\right) \parallel p(\mathbf{z}|\mathbf{v}; \boldsymbol{\gamma})\right). \tag{VII.10}$$

This network architecture appears to be very effective for the task at hand. In fact, if one looks at the cost function in (VII.10), it can be seen that the KL term achieves its optimal value for $q\left(\mathbf{z}|\mathbf{s}, \mathbf{v}; \boldsymbol{\psi}\right) = p(\mathbf{z}|\mathbf{v}; \boldsymbol{\gamma})$. By looking at the encoder of Fig. VII.3, this can happen by ignoring the contribution of the audio input. Moreover, the first term in the cost function (VII.10) attempts to reconstruct as well as possible the audio speech vector at the output of the decoder. This can be done by using the audio vector in the input of the encoder as much as possible. This stays in contrast with the optimal behavior of the second term which tries to ignore the audio input. By minimizing the overall cost, the visual and audio information can be fused in the encoder.

VII.5.3 Training of the AV-VAE

During the training of AV-VAE, the variable $\mathbf{z}_n$ is sampled from the approximate posterior modeled by the encoder, and it is then passed to the decoder. However, at test time only the decoder and prior networks are used while the encoder is discarded. Hence, $\mathbf{z}_n$ is sampled from the prior network, which is basically different from the encoder

network. The KL-divergence term in the cost function (VII.10) is responsible for reducing as much as possible the discrepancy between the recognition and prior networks. One can even control this by weighting the KL-divergence term with $\beta > 1$:

$$\mathcal{L}_{\beta\text{-av-cvae}}\left(\mathbf{s},\mathbf{v};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\gamma}\right) = \mathbb{E}_{q(\mathbf{z}|\mathbf{s},\mathbf{v};\boldsymbol{\psi})}\left[\ln p\left(\mathbf{s}|\mathbf{z},\mathbf{v};\boldsymbol{\theta}\right)\right] - \beta D_{\mathsf{KL}}\left(q\left(\mathbf{z}|\mathbf{s},\mathbf{v};\boldsymbol{\psi}\right) \parallel p(\mathbf{z}|\mathbf{v};\boldsymbol{\gamma})\right). \tag{VII.11}$$

This was introduced in [257], namely β-VAE, and was shown to facilitate the automated discovery of interpretable factorized latent representations. However, in the case of the proposed AV-VAE architecture, we follow a different stragety, proposed in [256], in order to decrease the gap between the recognition and prior networks. As a consequence, the ELBO defined in (VII.10) is modified as follows:

$$\tilde{\mathcal{L}}_{\text{av-cvae}}\left(\mathbf{s},\mathbf{v};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\gamma}\right) = \alpha\mathcal{L}_{\text{av-cvae}}\left(\mathbf{s},\mathbf{v};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\gamma}\right) + (1-\alpha)\mathbb{E}_{p(\mathbf{z}|\mathbf{v};\boldsymbol{\gamma})}\left[\ln p\left(\mathbf{s}|\mathbf{z},\mathbf{v};\boldsymbol{\theta}\right)\right], \tag{VII.12}$$

where $0 \leq \alpha \leq 1$ is a trade-off parameter. Note that the original ELBO is obtained by setting $\alpha = 1$. The new term in the right-hand side of the above cost function is actually the original reconstruction cost in (VII.10) but with each $\mathbf{z}_n$ being sampled from the prior distribution, i.e. $p\left(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\gamma}\right)$. In this way the prior network is forced to learn latent vectors that are suitable for reconstructing the corresponding speech frames. As it will be shown below, this method significantly improves the overall speech enhancement performance.

To develop the cost function in (VII.12), we note that the KL-divergence term admits a closed-form solution, because the involved distributions are Gaussian. Furthermore, since the expectations with respect to the approximate posterior and prior of $\mathbf{z}_n$ are not tractable, we approximate them using Monte-Carlo estimations, as usually done in practice. After some mathematical manipulations, one obtains the following cost function:

$$\begin{aligned}\tilde{\mathcal{L}}_{\text{av-cvae}}\big(\mathbf{s},\mathbf{v};\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\gamma}\big) = \frac{1}{R}\sum_{r=1}^{R}\sum_{n=0}^{N_{tr}-1}\Big(\alpha\ln p(\mathbf{s}_n|\mathbf{z}_{n,1}^{(r)},\mathbf{v}_n;\boldsymbol{\theta}) + (1-\alpha)\ln p(\mathbf{s}_n|\mathbf{z}_{n,2}^{(r)},\mathbf{v}_n;\boldsymbol{\theta})\Big) \\ + \frac{\alpha}{2}\sum_{l=0}^{L-1}\sum_{n=0}^{N_{tr}-1}\Big(\ln\frac{\tilde{\sigma}_l\left(\tilde{\mathbf{s}}_n,\mathbf{v}_n\right)}{\bar{\sigma}_l(\mathbf{v}_n)} - \frac{\ln\tilde{\sigma}_l\left(\tilde{\mathbf{s}}_n,\mathbf{v}_n\right) + \left(\tilde{\mu}_l\left(\tilde{\mathbf{s}}_n,\mathbf{v}_n\right) - \bar{\mu}_l(\mathbf{v}_n)\right)^2}{\bar{\sigma}_l(\mathbf{v}_n)}\Big),\end{aligned} \tag{VII.13}$$

where $\mathbf{z}_{n,1}^{(r)} \sim q(\mathbf{z}_n|\mathbf{s}_n,\mathbf{v}_n;\boldsymbol{\psi})$ and $\mathbf{z}_{n,2}^{(r)} \sim p(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\gamma})$. This cost function can be optimized in a similar way as with classical VAEs, namely by using the reparametrization trick together with a stochastic gradient-ascent algorithm. Notice that the reparameterization trick must be used twice, for $\mathbf{z}_{n,1}^{(r)}$ and for $\mathbf{z}_{n,2}^{(r)}$.

VII.6 AV-VAE for Speech Enhancement

This section describes the speech enhancement algorithm based on the proposed AV-VAE speech model. It is very similar to the algorithm that was proposed in [215] for audio-only speech enhancement with VAE. The unsupervised noise model is first presented, followed by the mixture model, and by the proposed algorithm to estimate the parameters of the noise model. Finally, clean-speech inference procedure is described. Through this section, $\mathbf{v} = \{\mathbf{v}_n\}_{n=0}^{N-1}$, $\mathbf{s} = \{\mathbf{s}_n\}_{n=0}^{N-1}$ and $\mathbf{z} = \{\mathbf{z}_n\}_{n=0}^{N-1}$ denote the test sets of visual features, clean-speech STFT features and latent vectors, respectively. These variables are associated with a noisy-speech test sequence of N frames. One should notice that the test data are different than the training data used in the previous sections. The observed microphone (mixture) frames are denoted by $\mathbf{x} = \{\mathbf{x}_n\}_{n=0}^{N-1}$.

VII.6.1 Unsupervised Noise Modeling

As in [214], [215], we use an unsupervised NMF-based Gaussian noise model that assumes independence across TF bins:

$$b_{fn} \sim \mathcal{N}_c\left(0, \left(\mathbf{W}_b\mathbf{H}_b\right)_{fn}\right), \tag{VII.14}$$

where $\mathbf{W}_b \in \mathbb{R}_+^{F\times K}$ is a nonnegative matrix of spectral power patterns and $\mathbf{H}_b \in \mathbb{R}_+^{K\times N}$ is a nonnegative matrix of temporal activations, with K being chosen such that $K(F+N) \ll FN$ [210]. We remind that $\mathbf{W}_b$ and $\mathbf{H}_b$ need to be estimated from the observed microphone signal.

The observed mixture (microphone) signal is modeled as follows:

$$x_{fn} = \sqrt{g_n}s_{fn} + b_{fn}, \tag{VII.15}$$

for all TF bins (f,n), where $g_n \in \mathbb{R}_+$ represents a frame-dependent and frequency-independent gain, as suggested in [215]. This gain provides robustness of the AV-VAE model with respect to the possibly highly varying loudness of the speech signal across frames. Let us denote by $\mathbf{g} = (g_0 \dots g_{N-1})^\top$ the vector of gain parameters that must be estimated. The speech and noise signals are further assumed to be mutually independent, such that by combining (VII.7), (VII.14) and (VII.15), we obtain, for all TF bins (f,n):

$$x_{fn}|\mathbf{z}_n,\mathbf{v}_n \sim \mathcal{N}_c\left(0, g_n\sigma_f(\mathbf{z}_n,\mathbf{v}_n) + \left(\mathbf{W}_b\mathbf{H}_b\right)_{f,n}\right). \tag{VII.16}$$

VII.6.2 Parameter Estimation

Having defined the speech generative model (VII.7), (VII.8), and the observed mixture model (VII.16), the inference process requires to estimate the set of model parameters $\boldsymbol{\phi} = \{\mathbf{W}_b, \mathbf{H}_b, \mathbf{g}\}$ from the set of observed STFT coefficients $\mathbf{x}$ and of observed visual features $\mathbf{v}$. Then, these parameters will be used to estimate the clean-speech STFT coefficients. Since integration with respect to the latent variables is intractable, straightforward maximum likelihood estimation of $\boldsymbol{\phi}$ is not possible. Alternatively, the latent-variable structure of the model can be exploited to derive an expectation-maximization (EM) algorithm [6]. Starting from an initial set of model parameters $\boldsymbol{\phi}^\star$, EM consists of iterating until convergence between:

- E-step: Evaluate $Q(\boldsymbol{\phi}; \boldsymbol{\phi}^\star) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x},\mathbf{v};\boldsymbol{\phi}^\star)}[\ln p(\mathbf{x}, \mathbf{z}, \mathbf{v}; \boldsymbol{\phi})]$;
- M-step: Update $\boldsymbol{\phi}^\star \leftarrow \text{argmax}_{\boldsymbol{\phi}}\ Q(\boldsymbol{\phi}; \boldsymbol{\phi}^\star)$.

E-Step Because of the non-linear relation between the observations and the latent variables in (VII.16), we cannot compute the posterior distribution $p(\mathbf{z}|\mathbf{x}, \mathbf{v}; \boldsymbol{\phi}^\star)$, and hence we cannot evaluate $Q(\boldsymbol{\phi}; \boldsymbol{\phi}^\star)$ analytically. As in [215], we thus rely on the following Monte Carlo approximation:

$$\begin{aligned} Q(\boldsymbol{\phi}; \boldsymbol{\phi}^\star) &\approx \tilde{Q}(\boldsymbol{\phi}; \boldsymbol{\phi}^\star) \\ &\overset{c}{=} -\frac{1}{R}\sum_{r=1}^{R}\sum_{(f,n)}\left(\ln\left(g_n\sigma_f(\mathbf{z}_n^{(r)}, \mathbf{v}_n) + (\mathbf{W}_b\mathbf{H}_b)_{f,n}\right) + \frac{|x_{fn}|^2}{g_n\sigma_f(\mathbf{z}_n^{(r)}, \mathbf{v}_n) + (\mathbf{W}_b\mathbf{H}_b)_{f,n}}\right), \end{aligned} \tag{VII.17}$$

where $\overset{c}{=}$ denotes equality up to additive terms that do not depend on $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^\star$, and where $\{\mathbf{z}_n^{(r)}\}_{r=1}^R$ is a sequence of samples drawn from the posterior $p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{v}_n; \boldsymbol{\phi}^\star)$ using Markov Chain Monte Carlo (MCMC) sampling. In practice we use the Metropolis-Hastings algorithm [258], which forms the basis of the MCEM algorithm [252]. At the m-th iteration of the Metropolis-Hastings algorithm and independently for all $n \in \{0, ..., N-1\}$, a sample $\mathbf{z}_n$ is first drawn from a proposal random walk distribution:

$$\mathbf{z}_n|\mathbf{z}_n^{(m-1)}; \epsilon^2 \sim \mathcal{N}(\mathbf{z}_n^{(m-1)}, \epsilon^2\mathbf{I}). \tag{VII.18}$$

Using the fact that this is a symmetric proposal distribution [258], the acceptance probability η is computed by:

$$\eta = \min\left(1, \frac{p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\phi}^\star)p(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\gamma}^\star)}{p(\mathbf{x}_n|\mathbf{z}_n^{(m-1)}, \mathbf{v}_n; \boldsymbol{\phi}^\star)p(\mathbf{z}_n^{(m-1)}|\mathbf{v}_n; \boldsymbol{\gamma}^\star)}\right),$$

where

$$p\left(\mathbf{x}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\phi}^\star\right) = \prod_{f=0}^{F-1} p(x_{fn}|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}_u^\star), \tag{VII.19}$$

with $p(x_{fn}|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}_u^\star)$ defined in (VII.16) and $p\left(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\gamma}^\star\right)$ defined in (VII.8). Next, u is drawn from a uniform distribution $\mathcal{U}([0, 1])$. If $u < \eta$, the sample is accepted and we set $\mathbf{z}_n^{(m)} = \mathbf{z}_n$, otherwise the sample is rejected and we set $\mathbf{z}_n^{(m)} = \mathbf{z}_n^{(m-1)}$. Only the last R samples are kept for computing $\tilde{Q}(\boldsymbol{\phi}; \boldsymbol{\phi}^\star)$ in (VII.17), i.e. the samples drawn during the so-called burn-in period are discarded.

M-Step $\tilde{Q}(\boldsymbol{\phi}; \boldsymbol{\phi}^\star)$ in (VII.17) is maximized with respect to the new model parameters $\boldsymbol{\phi}$. As usual in the NMF literature [259], we adopt a block-coordinate approach by successively and individually updating $\mathbf{H}_b$, $\mathbf{W}_b$ and $\mathbf{g}$, using the auxiliary function technique as done in [215]. Following the same methodology, we obtain the following formula for updating the NMF model parameters:

$$\mathbf{H}_b \leftarrow \mathbf{H}_b \odot \left(\frac{\mathbf{W}_b^\top\left(|\mathbf{X}|^{\odot 2} \odot \sum_{r=1}^{R}\left(\mathbf{V}_x^{(r)}\right)^{\odot -2}\right)}{\mathbf{W}_b^\top \sum_{r=1}^{R}\left(\mathbf{V}_x^{(r)}\right)^{\odot -1}}\right)^{\odot 1/2}, \tag{VII.20}$$

$$\mathbf{W}_b \leftarrow \mathbf{W}_b \odot \left(\frac{\left(|\mathbf{X}|^{\odot 2} \odot \sum_{r=1}^{R}\left(\mathbf{V}_x^{(r)}\right)^{\odot -2}\right)\mathbf{H}_b^\top}{\sum_{r=1}^{R}\left(\mathbf{V}_x^{(r)}\right)^{\odot -1}\mathbf{H}_b^\top}\right)^{\odot 1/2}, \tag{VII.21}$$

where $(\cdot)^{\odot(\cdot)}$ denotes element-wise exponentiation, $(\cdot) \odot (\cdot)$ denotes element-wise multiplication, and $\frac{(\cdot)}{(\cdot)}$ denotes element-wise division. Moreover, $\mathbf{V}_x^{(r)} \in \mathbb{R}_+^{F \times N}$ is the matrix with entries $g_n \sigma_f(\mathbf{z}_n^{(r)}, \mathbf{v}_n) + (\mathbf{W}_b \mathbf{H}_b)_{f,n}$, and $\mathbf{X} \in \mathbb{C}^{F \times N}$ is the matrix with entries $(\mathbf{X})_{f,n} = x_{fn}$. The gains are updated as follows:

$$\mathbf{g}^\top \leftarrow \mathbf{g}^\top \odot \left(\frac{\mathbf{1}^\top \left(|\mathbf{X}|^{\odot 2} \odot \sum_{r=1}^{R} \left(\mathbf{V}_s^{(r)} \odot \left(\mathbf{V}_x^{(r)} \right)^{\odot -2} \right) \right)}{\mathbf{1}^\top \left[\sum_{r=1}^{R} \left(\mathbf{V}_s^{(r)} \odot \left(\mathbf{V}_x^{(r)} \right)^{\odot -1} \right) \right]} \right)^{\odot 1/2}, \tag{VII.22}$$

where $\mathbf{1}$ is a vector of ones with dimension F and $\mathbf{V}_s^{(r)} \in \mathbb{R}_+^{F \times N}$ is the matrix with entries $\sigma_f(\mathbf{z}_n^{(r)}, \mathbf{v}_n)$. The nonnegative property of $\mathbf{H}_b$, $\mathbf{W}_b$, and of $\mathbf{g}$ is ensured, provided that their entries are initialized with nonnegative values. In practice, only one iteration of updates (VII.20), (VII.21) and (VII.22) is performed at each M-step.

VII.6.3 Speech Reconstruction with AV-VAE

Let $\boldsymbol{\phi}^* = \{\mathbf{W}_b^*, \mathbf{H}_b^*, \mathbf{g}^*\}$ denote the set of parameters estimated by the above MCEM algorithm. Let $\tilde{s}_{fn} = \sqrt{g_n^*} s_{fn}$ be the scaled version of the speech STFT coefficients as introduced in (VII.15), with $g_n^* = (\mathbf{g}^*)_n$. The final step is to estimate these coefficients according to their posterior mean [215]:

$$\begin{aligned} \hat{\tilde{s}}_{fn} &= \mathbb{E}_{p(\tilde{s}_{fn}|x_{fn}, \mathbf{v}_n; \boldsymbol{\phi}^*)}[\tilde{s}_{fn}] = \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{v}_n; \boldsymbol{\phi}^*)} \left[\mathbb{E}_{p(\tilde{s}_{fn}|\mathbf{z}_n, \mathbf{v}_n, \mathbf{x}_n; \boldsymbol{\phi}^*)}[\tilde{s}_{fn}] \right] \\ &= \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{v}_n; \boldsymbol{\phi}^*)} \left[\frac{g_n^* \sigma_f(\mathbf{z}_n, \mathbf{v}_n)}{g_n^* \sigma_f(\mathbf{z}_n, \mathbf{v}_n) + (\mathbf{W}_b^* \mathbf{H}_b^*)_{f,n}} \right] x_{fn}. \end{aligned} \tag{VII.23}$$

This estimation corresponds to a "probabilistic" version of Wiener filtering, with an averaging of the filter over the posterior distribution of the latent variables. As above, this expectation cannot be computed analytically, but instead it can be approximated using the same Metropolis-Hastings algorithm of Section VII.6.2. The time-domain estimate of the speech signal is finally obtained from the inverse STFT with overlap-add.

VII.7 The Mixture of Inference Networks VAE

In this section, we aim to devise a framework able to choose the best combination between the auditory and visual encodings, as opposed to systematically using both encodings like in AV-VAE. To achieve this goal, we propose a probabilistic mixture of the audio and visual encoders, and name it mixture of inference networks VAE (MIN-VAE). In a nutshell, the model learns to select if the posterior of $\mathbf{z}_n$ should be audio- or video-based.. In the following we introduce the mathematical formulation associated with the proposed MIN-VAE. The overall architecture is depicted in the Figure VII.4.

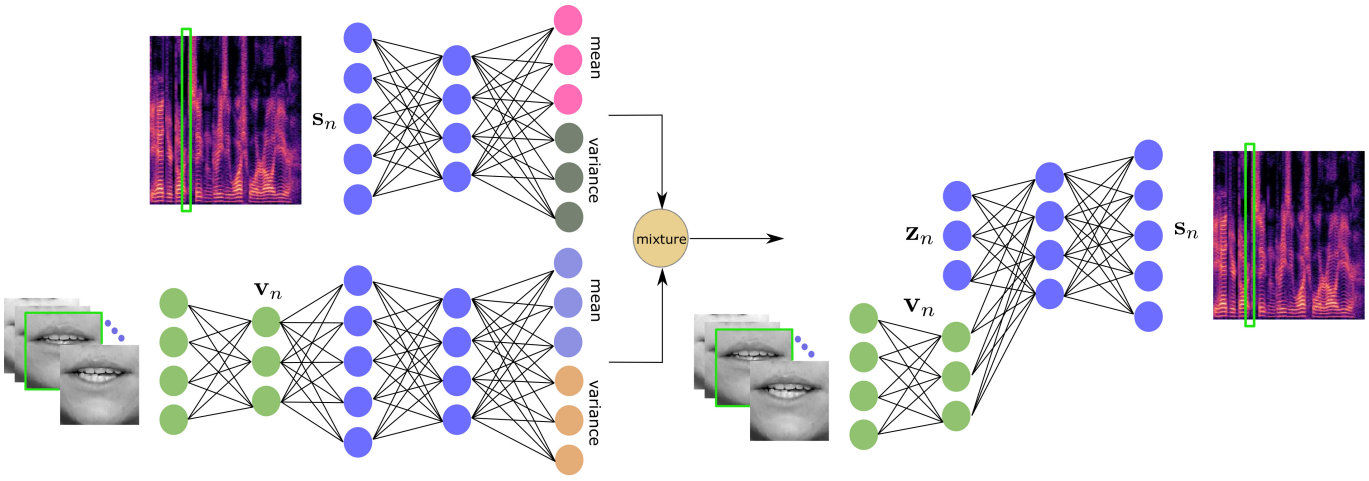


Figure VII.4: *Architecture of the proposed mixture of inference networks VAE (MIN-VAE). A mixture of an audio- and a video-based encoder is used to approximate the intractable posterior distribution of the latent variables.*

VII.7.1 The Generative Model

We assume that each latent code is generated either from an audio or from a video prior. We model this with a mixing variable $\alpha_n \in \{0, 1\}$ describing whether the latent code $\mathbf{z}_n$ corresponds to the audio or to the visual prior. Once the latent code is generated from the corresponding prior, the speech frame $\mathbf{s}_n$ follows a complex Gaussian with the variance computed by the decoder. We recall that the variance is a non-linear transformation of the latent code.

Formally, each STFT time frame $\mathbf{s}_n$ is modeled as:

$$\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}\Big(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n)\Big)\Big), \tag{VII.24}$$

$$\mathbf{z}_n|\alpha_n \sim \Big[\mathcal{N}(\boldsymbol{\mu}_a, \sigma_a\mathbf{I})\Big]^{\alpha_n} \cdot \Big[\mathcal{N}(\boldsymbol{\mu}_v, \sigma_v\mathbf{I})\Big]^{1-\alpha_n}, \tag{VII.25}$$

$$\alpha_n \sim \pi^{\alpha_n} \times (1-\pi)^{1-\alpha_n}, \tag{VII.26}$$

where the audio and video priors are parametrized by $(\boldsymbol{\mu}_a, \sigma_a)$ and $(\boldsymbol{\mu}_v, \sigma_v)$ respectively, and α_n is assumed to follow a Bernoulli distribution with parameter π. We propose two versions of this architecture, namely: MIN-VAE-v1 where the decoder (VII.24) takes the same form as (VII.7) and uses explicitly visual information (see Fig. VII.4), and MIN-VAE-v2 where the decoder (VII.24) takes the same form as (VII.1) and does not use explicitly visual information. In both cases the parameters of the decoder are denoted by $\boldsymbol{\theta}$. The derivations will be done for the general case, that is MIN-VAE-v1.

VII.7.2 The Posterior Distribution

In order to estimate the parameters of the generative model described above, i.e. $\boldsymbol{\psi} = \{\boldsymbol{\mu}_a, \boldsymbol{\mu}_v, \sigma_a, \sigma_v\}$, $\boldsymbol{\theta}$, and π, we follow a maximum likelihood procedure. To derive it, we need to compute the posterior of the latent variables:

$$p(\mathbf{z}_n, \alpha_n|\mathbf{s}_n, \mathbf{v}_n) = p(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n) \cdot p(\alpha_n|\mathbf{s}_n, \mathbf{v}_n). \tag{VII.27}$$

The individual factors in the right-hand side of the above equation cannot be computed in closed-form, due to the non-linear generative model. As similarly done in VAE, we pursue an amortized inference approach to approximate $p(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n)$ with a parametric Gaussian distribution defined as follows:

$$q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) = \begin{cases} q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a) & \alpha_n = 1, \\ q(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\phi}_v) & \alpha_n = 0, \end{cases} \tag{VII.28}$$

in which, $\boldsymbol{\phi} = \{\boldsymbol{\phi}_a, \boldsymbol{\phi}_v\}$, and $\boldsymbol{\phi}_a$ and $\boldsymbol{\phi}_v$ denote the parameters of the associated audio and visual inference neural networks, taking the same architectures as those in (VII.4) and (VII.6), respectively. For the posterior of α_n, i.e. $p(\alpha_n|\mathbf{s}_n, \mathbf{v}_n)$, we resort to a variational approximation, denoted $r(\alpha_n)$. Put it all together, we have the following approximate posterior:

$$q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) \cdot r(\alpha_n) \approx p(\mathbf{z}_n, \alpha_n|\mathbf{s}_n, \mathbf{v}_n). \tag{VII.29}$$

VII.7.3 Training the MIN-VAE

In order to train the MIN-VAE, we devise an optimization procedure alternating between estimating $\boldsymbol{\Theta} = \{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}, \pi\}$ and updating the variational posterior r. We recall the definition $\mathbf{s} = \{\mathbf{s}_n\}_{n=1}^{N_{tr}}$, and $\mathbf{z}$, and define $\boldsymbol{\alpha}$ and $\mathbf{v}$ analogously. The full posterior of the latent variables can be written as:

$$p(\mathbf{z}, \boldsymbol{\alpha}|\mathbf{s}, \mathbf{v}) = \frac{p(\mathbf{s}, \mathbf{v}, \mathbf{z}, \boldsymbol{\alpha})}{p(\mathbf{s}, \mathbf{v})} = \frac{p(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta})}. \tag{VII.30}$$

We then target the KL-divergence between the approximate posterior and the true posterior which reads:

$$\begin{aligned} &\mathcal{D}_{KL}\Big(q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})r(\boldsymbol{\alpha}) \| p(\mathbf{z}, \boldsymbol{\alpha}|\mathbf{s}, \mathbf{v})\Big) = \\ &= \int_{\mathbb{Z},\mathbb{A}} q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})r(\boldsymbol{\alpha}) \log \frac{q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})r(\boldsymbol{\alpha})p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta})}{p(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}\mathrm{d}\boldsymbol{\alpha} \\ &= -\mathcal{L}(\boldsymbol{\Theta}, r) + \log p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta}) \geq 0, \end{aligned} \tag{VII.31}$$

where

$$\mathcal{L}(\boldsymbol{\Theta}, r) = \int_{\mathbb{Z},\mathbb{A}} q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})r(\boldsymbol{\alpha}) \log \frac{p(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})r(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}\mathrm{d}\boldsymbol{\alpha}. \tag{VII.32}$$

From (VII.31) we can see that $\log p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\Theta}, r)$. Therefore, instead of maximizing the intractable data log-likelihood $\log p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta})$, we maximize its lower-bound, i.e. $\mathcal{L}(\boldsymbol{\Theta}, r)$, or equivalently:

$$\boldsymbol{\Theta}^*, r^* = \underset{\boldsymbol{\Theta}, r}{\operatorname{argmin}} -\mathcal{L}(\boldsymbol{\Theta}, r) \tag{VII.33}$$

subject to the constraint that r integrates to one. We solve this problem by alternately optimizing the cost over r and $\boldsymbol{\Theta}$. In the following, the two optimization steps are discussed.

Optimizing w.r.t. $r(\alpha)$ With $\boldsymbol{\Theta}$ being fixed to its current estimate, solving (VII.33) boils down to:

$$\min_{r_n(\alpha_n)} \int_{\mathbb{A}} r_n(\alpha_n) \Big[\log \frac{r_n(\alpha_n)}{p(\alpha_n)} + J_n(\alpha_n) \Big] \mathrm{d}\alpha_n, \forall n, \tag{VII.34}$$

meaning that the optimal r is separable on n, where,

$$\begin{aligned} J_n(\alpha_n) &= \int_{\mathbb{Z}} q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) \log \frac{q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi})}{p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}) p(\mathbf{z}_n|\alpha_n)} \mathrm{d}\mathbf{z}_n \\ &= D_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) \| p(\mathbf{z}_n|\alpha_n) \Big) - \mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi})} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}) \Big]. \end{aligned} \tag{VII.35}$$

Using calculus of variations, we find that $r_n(\alpha_n) \propto p(\alpha_n) \exp\big(-J_n(\alpha_n)\big)$, which is a Bernoulli distribution. To find the associated parameter, we need to compute $J_n(\alpha_n)$. Since the expectation involved in (VII.35) is intractable to compute, we approximate it using a single sample denoted $\mathbf{z}_n^{\alpha_n}$ drawn from $q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi})$, obtaining:

$$\tilde{J}_n(\alpha_n) = D_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) \| p(\mathbf{z}_n|\alpha_n) \Big) - \log p(\mathbf{s}_n|\mathbf{z}_n^{\alpha_n}, \mathbf{v}_n; \boldsymbol{\theta}), \tag{VII.36}$$

The parameter of the Bernoulli distribution then takes the following form:

$$\pi_n = g\Big(\tilde{J}_n(\alpha_n = 0) - \tilde{J}_n(\alpha_n = 1) + \log \frac{\pi}{1-\pi} \Big), \tag{VII.37}$$

where $g(x) = 1/(1+\exp(-x))$ is the sigmoid function. To compute the KL divergence terms, we use the following lemma:

Lemma Let $p_1(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $p_2(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ be two multivariate Gaussian distributions in $\mathbb{R}^n$. Then, the KL divergence between p_1 and p_2 reads:

$$D_{KL}(p_1 \| p_2) = \frac{1}{2} \Big(\log \frac{|\boldsymbol{\Sigma}|_2}{|\boldsymbol{\Sigma}|_1} - n + \text{trace}\,(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_2^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \Big). \tag{VII.38}$$

Utilizing the above lemma, we can write (for $\alpha_n = 1$):

$$D_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a) \parallel p(\mathbf{z}_n|\alpha_n) \Big) = \frac{1}{2} \log \frac{\sigma_a^L}{\Big|\text{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big|} - + \frac{^\top \Big(\text{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big) + \|\boldsymbol{\mu}_z^a(\mathbf{s}_n) - \boldsymbol{\mu}_a\|^2}{2\sigma_a} - \frac{L}{2}, \tag{VII.39}$$

and analogously for the vision-based term ($\alpha_n = 0$).

Optimizing w.r.t. $\boldsymbol{\Theta}$ With r being fixed to its current estimate, from (VII.33), we can write the optimization over $\boldsymbol{\Theta}$ as:

$$\begin{aligned} &\min_{\boldsymbol{\Theta}} \int_{\mathbb{Z},\mathbb{A}} q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi}) r(\boldsymbol{\alpha}) \log \frac{q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi}) r(\boldsymbol{\alpha})}{p(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}\mathrm{d}\boldsymbol{\alpha} \\ &= \min_{\boldsymbol{\Theta}} \mathbb{E}_{r(\boldsymbol{\alpha})} \Big[\int_{\mathbb{Z}} q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi}) \log \frac{q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\phi})}{p(\mathbf{s}|\mathbf{z}, \mathbf{v}; \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha})} \mathrm{d}\mathbf{z} \Big] + D_{KL}\Big(r(\boldsymbol{\alpha}) \parallel p(\boldsymbol{\alpha}) \Big) \\ &= \min_{\boldsymbol{\Theta}} \sum_{n=0}^{N_{tr}} \mathbb{E}_{r(\alpha_n)} \Big[J_n(\alpha_n) \Big] + D_{KL}\Big(r(\alpha_n) \parallel p(\alpha_n) \Big) \\ &= \min_{\boldsymbol{\Theta}} \sum_{n=0}^{N_{tr}} \pi_n \Big(D_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a) \| p(\mathbf{z}_n|\alpha_n = 1) \Big) - \mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a)} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}) \Big] \Big) + \\ &(1-\pi_n) \Big(D_{KL}\Big(q(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\phi}_v) \| p(\mathbf{z}_n|\alpha_n = 0) \Big) - \mathbb{E}_{q(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\phi}_v)} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}) \Big] \Big) + D_{KL}\Big(r(\alpha_n) \parallel p(\alpha_n) \Big). \end{aligned} \tag{VII.40}$$

As before, the expectations involved in the above equation are approximated with a single sample drawn from the associated posteriors, resulting in:

$$\sum_{n=1}^{N_{tr}} -\pi_n \ln p\left(\mathbf{s}_n|\mathbf{z}_n^a;\boldsymbol{\theta}\right) - (1-\pi_n)\ln p\left(\mathbf{s}_n|\mathbf{z}_n^v;\boldsymbol{\theta}\right) + D_{KL}\Big(r\left(\alpha_n\right) \parallel p(\alpha_n)\Big) + \tag{VII.41}$$
$$\pi_n D_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\phi}_a) \parallel p(\mathbf{z}_n|\alpha_n=1)\Big) + (1-\pi_n) D_{KL}\Big(q(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\phi}_v) \parallel p(\mathbf{z}_n|\alpha_n=0)\Big),$$

where, $\mathbf{z}_n^a \sim q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\phi}_a)$ and $\mathbf{z}_n^v \sim q(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\phi}_v)$. After computing the cost function, the parameters are updated using a re-parametrization trick along with a stochastic gradient descent algorithm, e.g. the Adam optimizer. Finally, optimizing (VII.41) over π leads to minimizing the following KL-divergence:

$$D_{KL}\Big(q\left(\alpha_n\right) \parallel p(\alpha_n)\Big) = \pi_n \log\frac{\pi_n}{\pi} + (1-\pi_n)\log\frac{1-\pi_n}{\pi}, \tag{VII.42}$$

yielding

$$\pi = \frac{1}{N_{tr}}\sum_{n=1}^{N_{tr}} \pi_n. \tag{VII.43}$$

Now, with the derived variational inference formulas, we obtain the inference mixture for the MIN-VAE:

$$p(\mathbf{z}_n|\mathbf{s}_n,\mathbf{v}_n) = \pi_n \mathcal{N}\Big(\boldsymbol{\mu}_z^a(\mathbf{s}_n), \text{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big) + (1-\pi_n)\mathcal{N}\Big(\boldsymbol{\mu}_z^v(\mathbf{v}_n), \text{diag}\Big(\boldsymbol{\sigma}_z^v(\mathbf{v}_n)\Big)\Big). \tag{VII.44}$$

The overall training algorithm then consists of alternating the variational distribution update of α_n via (VII.37), the update of $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$ via stochastic gradient descent of (VII.41), and the update of π using (VII.43).

VII.8 MIN-VAE for Speech enhancement

VII.8.1 Unsupervised Noise Modeling

At test time, once the MIN-VAE is trained, the STFT time frames of the observed noisy speech are modeled as $\mathbf{x}_n = \mathbf{s}_n + \mathbf{b}_n$, for $n = 0,\ldots,N-1$, with $\mathbf{b}_n$ denoting noise STFT time frame. For the probabilistic modeling of $\mathbf{s}_n$, we use the generative model trained on clean data (i.e. the previous section). For $\mathbf{b}_n$, the following NMF based model is considered [215]:[VII.1]

$$\mathbf{b}_n \sim \mathcal{N}\Big(\mathbf{0}, \text{diag}\Big(\mathbf{W}\boldsymbol{h}_n\Big)\Big), \tag{VII.45}$$

where, $\mathbf{W} \in \mathbb{R}_+^{F\times K}$, and $\boldsymbol{h}_n$ denotes the n-th column of $\mathbf{H} \in \mathbb{R}_+^{K\times N}$. The parameters, i.e. $\{\mathbf{W},\mathbf{H}\}$, as well as the unknown speech are then estimated following a Monte-Carlo Expectation-Maximization (MCEM) method [260]. This strategy is inspired by the recent literature [215], [261].

The generative model consists of (VII.24), (VII.25), and (VII.26), where all the parameters except π have already been trained on clean audio and visual data. The observations are noisy STFT frames $\mathbf{x} = \{\mathbf{x}_n\}_{n=0}^{N-1}$, as well as the visual data $\mathbf{v} = \{\mathbf{v}_n\}_{n=0}^{N-1}$. The latent variables of the model are $\mathbf{s} = \{\mathbf{s}_n\}_{n=0}^{N-1}$, $\mathbf{z} = \{\mathbf{z}_n\}_{n=0}^{N-1}$, and $\boldsymbol{\alpha} = \{\alpha_n\}_{n=0}^{N-1}$. Furthermore, the parameters of the model are $\Theta = \{\mathbf{W},\mathbf{H},\pi\}$.

VII.8.2 Parameter Estimation

The full posterior is written as:

$$p(\mathbf{s}_n,\mathbf{z}_n,\alpha_n|\mathbf{x}_n,\mathbf{v}_n;\Theta) \propto p(\mathbf{x}_n,\mathbf{s}_n,\mathbf{z}_n,\mathbf{v}_n,\alpha_n;\Theta) = p(\mathbf{x}_n|\mathbf{s}_n;\Theta)\times p(\mathbf{s}_n|\mathbf{z}_n,\mathbf{v}_n)\times p(\mathbf{z}_n|\alpha_n)\times p(\alpha_n) \tag{VII.46}$$

To estimate the parameter set, we use variational expectation-maximization (VEM) [260], where in the variational expectation step (VE-step), the above intractable posterior is approximated by a variational distribution $r(\mathbf{s}_n,\mathbf{z}_n,\alpha_n)$, as similarly done in [261]. The maximization step (M-step) performs parameters update using the obtained variational distributions. We assume that r factorizes as follows:

$$r(\mathbf{s}_n,\mathbf{z}_n,\alpha_n) = r(\mathbf{s}_n)\times r(\mathbf{z}_n)\times r(\alpha_n). \tag{VII.47}$$

Denoting the current estimate of the parameters as Θ^{old}, the VEM approach consists of iterating between the VE-steps and the M-step, which are detailed below.

[VII.1] Compared to the noise model used with AV-VAE, we discard here the gain term. Both models, AV-VAE and MIN-VAE can be used with or without the gain term. In the experiments we conducted, we did not see any advantage of using the gain, so we discarded it to reduce te computational complexity. We believe that this phenomenon is due to the fact that the dataset we use is well balanced across samples, and therefore the gain term does not play an important role. When using an imbalanced dataset, this term could have an important impact.

VE-$r(\mathbf{s}_n)$ step The variational distribution of $\mathbf{s}_n$ is computed as [260]:

$$\begin{aligned} r(\mathbf{s}_n) &\propto \exp\Big(\mathbb{E}_{r(\mathbf{z}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \Theta^{old})\Big]\Big) \\ &\propto \exp\Big(\mathbb{E}_{r(\mathbf{z}_n)}\Big[\log p(\mathbf{x}_n|\mathbf{s}_n; \Theta^{old}) + \log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n)\Big]\Big) \\ &= \exp\Big(-\sum_f \Big[\frac{|x_{fn} - s_{fn}|^2}{(\mathbf{WH})_{fn}} + \frac{|s_{fn}|^2}{\gamma_{fn}}\Big]\Big), \end{aligned} \tag{VII.48}$$

where,

$$\gamma_{fn}^{-1} = \mathbb{E}_{r(\mathbf{z}_n)}\left[\frac{1}{\sigma_{s,f}(\mathbf{z}_n^{(d)}, \mathbf{v}_n)}\right] \approx \frac{1}{D}\sum_{d=1}^{D}\frac{1}{\sigma_{s,f}(\mathbf{z}_n^{(d)}, \mathbf{v}_n)}, \tag{VII.49}$$

and $\left\{\mathbf{z}_n^{(d)}\right\}_{d=1}^{D}$ is a sequence sampled from $r(\mathbf{z}_n)$. From (VII.48), we can see that $r(s_{fn}) = \mathcal{N}_c(m_{fn}, \nu_{fn})$, where:

$$m_{fn} = \frac{\gamma_{fn}}{\gamma_{fn} + (\mathbf{WH})_{fn}} \cdot x_{fn} \qquad \text{and} \qquad \nu_{fn} = \frac{\gamma_{fn} \cdot (\mathbf{WH})_{fn}}{\gamma_{fn} + (\mathbf{WH})_{fn}} \tag{VII.50}$$

VE-$r(\mathbf{z}_n)$ step The variational distribution of $\mathbf{z}_n$ can be computed by the following standard formula:

$$\begin{aligned} r(\mathbf{z}_n) &\propto \exp\Big(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \Theta^{old})\Big]\Big) \propto \exp\Big(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n) + \log p(\mathbf{z}_n|\alpha_n)\Big]\Big) \\ &\propto \exp\Big(\sum_f -\log\Big(\sigma_{s,f}(\mathbf{z}_n, \mathbf{v}_n)\Big) - \frac{|m_{fn}|^2 + \nu_{fn}}{\sigma_{s,f}(\mathbf{z}_n, \mathbf{v}_n)} + \sum_{\alpha_n \in \{0,1\}} r(\alpha_n) \cdot \Big[\log p(\mathbf{z}_n|\alpha_n)\Big]\Big) \triangleq \tilde{r}(\mathbf{z}_n) \end{aligned} \tag{VII.51}$$

This gives us an unnormalized distribution $\tilde{r}(\mathbf{z}_n)$ whose normalization constant cannot be computed in closed-from, due to the non-linear terms. However, we use the Metropolis-Hastings algorithm [260] to sample from it. To that end, we need to start with an initialization, $\mathbf{z}^{(0)}$. At the beginning of the inference, $\mathbf{z}^{(0)}$ is set to be the posterior mean in the output of the visual-encoder, i.e. the bottom-left network in Fig. VII.4, where $\mathbf{v}_n$ is given as the input. Then, a candidate sample denoted $\mathbf{z}^{(c)}$ is obtained by sampling from a proposal distribution, usually chosen to be a Gaussian:

$$\mathbf{z}^{(c)}|\mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{z}^{(0)}, \epsilon\mathbf{I}), \tag{VII.52}$$

where, $\epsilon > 0$ controls the speed of convergence. Then, $\mathbf{z}^{(c)}$ is set to be the next sample $\mathbf{z}^{(1)}$ with the following probability:

$$p = \min\Big(1, \frac{\tilde{r}(\mathbf{z}^{(c)})}{\tilde{r}(\mathbf{z}^{(0)})}\Big). \tag{VII.53}$$

That means, some u is drawn from a uniform distribution between 0 and 1. Then, if $u < p$, the sample is accepted and $\mathbf{z}^{(1)} = \mathbf{z}^{(c)}$. Otherwise, it is rejected and $\mathbf{z}^{(1)} = \mathbf{z}^{(0)}$. This procedure is repeated until the required number of samples is achieved. The first few samples are usually discarded, as they are not so reliable.

VE-$r(\alpha_n)$ step The variational distribution of α_n is computed as:

$$\begin{aligned} r(\alpha_n) &\propto \exp\Big(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \Theta^{old})\Big]\Big) \\ &\propto p(\alpha_n) \times \exp\Big(\mathbb{E}_{r(\mathbf{z}_n)}\Big[\alpha_n \cdot \log p(\mathbf{z}_n|\alpha_n = 1) + (1 - \alpha_n) \cdot \log p(\mathbf{z}_n|\alpha_n = 0)\Big]\Big) \end{aligned} \tag{VII.54}$$

which is a Bernoulli distribution with the following parameter:

$$\pi_n = g\Big(\mathbb{E}_{r(\mathbf{z}_n)}\Big[\log\frac{p(\mathbf{z}_n|\alpha_n = 1)}{p(\mathbf{z}_n|\alpha_n = 0)}\Big] + \log\frac{\pi}{1 - \pi}\Big), \tag{VII.55}$$

with $g(.)$ being the sigmoid function.

M-step After updating all the variational distributions, the next step is to update the set of parameters, i.e. $\Theta = \{\mathbf{W}, \mathbf{H}, \pi\}$. To do so, we need to optimize the complete-data log-likelihood which reads:

$$\begin{aligned} Q(\Theta; \Theta^{old}) &= \mathbb{E}_{r(\mathbf{s})\cdot r(\mathbf{z})\cdot r(\boldsymbol{\alpha})}\Big[\log p(\mathbf{x}, \mathbf{s}, \mathbf{z}, \boldsymbol{\alpha}, \mathbf{v}; \Theta)\Big] \overset{cte.}{=} \mathbb{E}_{r(\mathbf{s})}\Big[\log p(\mathbf{x}|\mathbf{s}; \Theta)\Big] + \mathbb{E}_{r(\boldsymbol{\alpha})}\Big[\log p(\boldsymbol{\alpha})\Big] \\ &\overset{cte.}{=} \sum_{f,n} -\frac{|x_{fn} - m_{fn}|^2 + \nu_{fn}}{(\mathbf{WH})_{fn}} - \log(\mathbf{WH})_{fn} + \pi_n \log\pi + (1 - \pi_n)\log(1 - \pi) \end{aligned} \tag{VII.56}$$

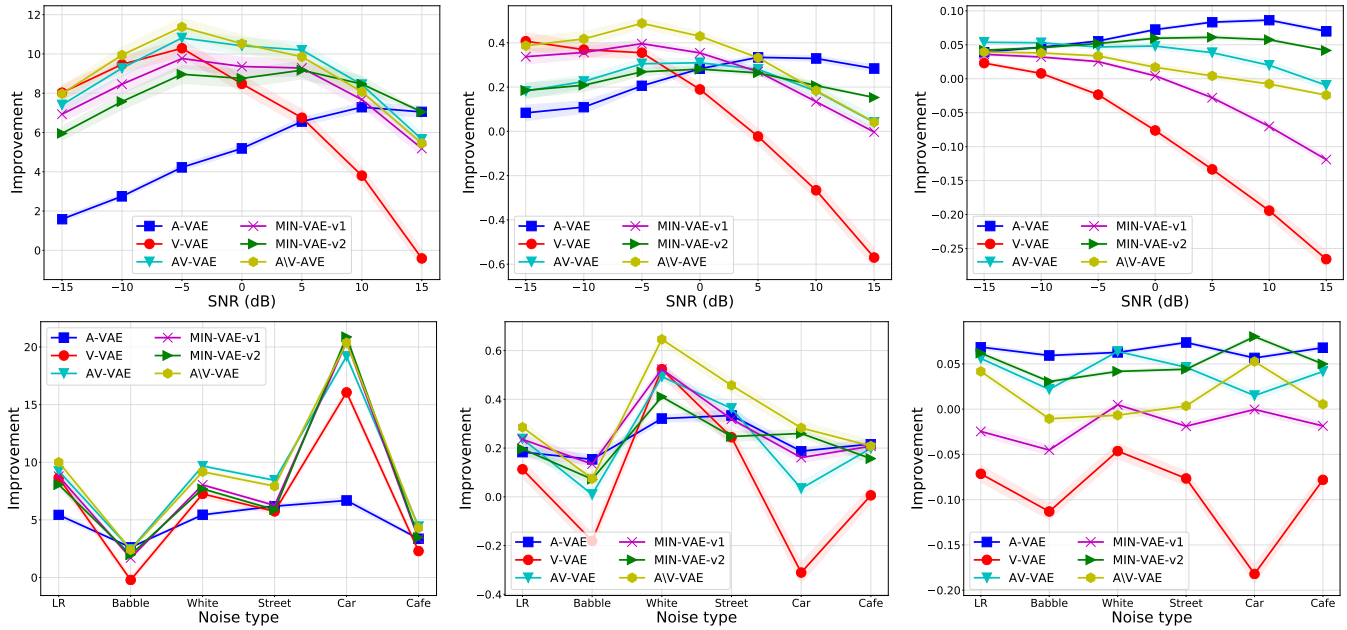


Figure VII.5: *Performance comparison of different VAE architectures for speech enhancement (left SDR, middle PESQ, right STOI). Top row shows the averaged results in terms of input noise levels, whereas the bottom row reports the averaged results versus different noise types. Here, no noise was added to the input of the audio-encoders of MIN-VAE-v1 and MIN-VAE-v2 during training.*

The update formulas for $\mathbf{W}$ and $\mathbf{H}$ can be obtained by using standard multiplicative updates [262]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top \left(\mathbf{V} \odot (\mathbf{W}\mathbf{H})^{\odot -2}\right)}{\mathbf{W}^\top (\mathbf{W}\mathbf{H})^{\odot -1}}, \tag{VII.57}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\left(\mathbf{V} \odot (\mathbf{W}\mathbf{H})^{\odot -2}\right) \mathbf{H}^\top}{(\mathbf{W}\mathbf{H})^{\odot -1} \mathbf{H}^\top}, \tag{VII.58}$$

where $\mathbf{V} = \left[|x_{fn} - m_{fn}|^2 + \nu_{fn}\right]_{(f,n)}$. Optimizing over π leads to a similar update formula as in (VII.43):

$$\pi = \frac{1}{N} \sum_{n=1}^{N} \pi_n. \tag{VII.59}$$

VII.8.3 Speech Reconstruction with MIN-VAE

Let $\Theta^* = \{\mathbf{W}^*, \mathbf{H}^*, \pi^*\}$ denote the optimal set of parameters found by the above VEM procedure. An estimation of the clean speech is then obtained as the variational posterior mean ($\forall f, n$):

$$\hat{s}_{fn} = \mathbb{E}_{r(s_{fn})}[s_{fn}] = \frac{\gamma^*_{fn}}{\gamma^*_{fn} + (\mathbf{W}^*\mathbf{H}^*)_{fn}} \cdot x_{fn}, \tag{VII.60}$$

where, γ^*_{fn}, defined in (VII.49), is computed using the optimal parameters.

VII.9 Experiments

In this section, we aim to evaluate the speech enhancement performance of different VAE architectures, including A-VAE [215], V-VAE [263], AV-VAE [263], and the proposed MIN-VAE. We consider two versions of our proposed network. The first one, named MIN-VAE-v1, is shown in Fig. VII.4. The second version, referred to as MIN-VAE-v2, shares the same architecture as MIN-VAE-v1 except that the visual features are not used in the decoder. To measure the performance, we use standard scores, including the signal-to-distortion ratio (SDR) [264], the perceptual evaluation of speech quality (PESQ) [265], and the short-time objective intelligibility (STOI) [266]. SDR is measured in decibels (dB), while PESQ and STOI values lie in the intervals $[-0.5, 4.5]$ and $[0, 1]$, respectively (the higher the better). For each measure, we report the averaged difference between the output value (evaluated on the enhanced speech signal) and the input value (evaluated on the noisy/unprocessed mixture signal).

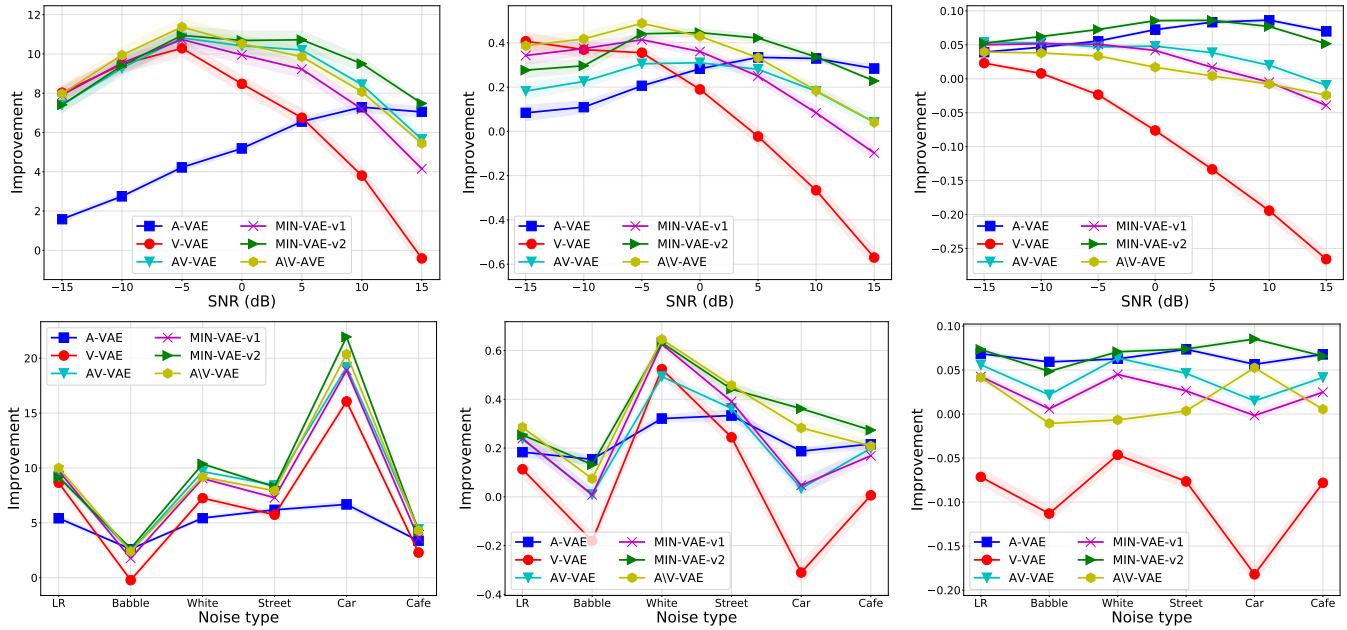


Figure VII.6: *Performance comparison of different VAE architectures for speech enhancement (left SDR, middle PESQ, right STOI). Top row shows the averaged results in terms of input noise levels, whereas the bottom row reports the averaged results versus different noise types. Here, some uniform noise was added to the input of the audio-encoders in MIN-VAE-v1 and MIN-VAE-v2 during training.*

VII.9.1 Experimental Set-up

Dataset We use the NTCD-TIMIT dataset [237], which contains AV recordings from 56 English speakers with an Irish accent, uttering 5488 different TIMIT sentences [267]. The visual data consists of 30 FPS videos of lips ROIs. Each frame (ROI) is of size 67×67 pixels. The speech signal is sampled at 16 kHz, and the audio spectral features are computed using an STFT window of 64 ms (1024 samples per frame) with 47.9% overlap, hence $F = 513$. The dataset is divided into 39 speakers for training, 8 speakers for validation, and 9 speakers for testing, as proposed in [237]. The test set includes about 1 hour noisy speech, along with their corresponding lips ROIs, with six different noise types, including *Living Room (LR)*, *White*, *Cafe*, *Car*, *Babble*, and *Street*, with noise levels: $\{-15, -10, -5, 0, 5, 10, 15\}$ dB.

Architecture and training details The generative networks (decoders) of A-VAE and V-VAE consist of a single hidden layer with 128 nodes and hyperbolic tangent activations. The dimension of the latent space is $L = 32$. The A-VAE encoder has a single hidden layer with 128 nodes and hyperbolic tangent activations. The V-VAE encoder is similar to that, except for extracting visual features embedding lip ROIs into a feature vector $\mathbf{v}_n \in \mathbb{R}^M$, with $M = 128$. This is composed of two fully-connected layers with 512 and 128 nodes, respectively. The dimension of the input corresponds to a single vectorized frame, namely $4489 = 67 \times 67$. AV-VAE combines the architectures of A-VAE and V-VAE as illustrated in Fig.VII.2 and VII.2. The audio and the video encoders in Fig. VII.4 share also the same architectures as those of A-VAE and V-VAE encoders, respectively.

To have a fair comparison, we fine-tunned the A-VAE and V-VAE of [263], which have been trained with a standard Gaussian prior for the latent variables, by using a parametric Gaussian prior, as the ones in (VII.25). The decoder parameters of MIN-VAE-v1 and MIN-VAE-v2 are initialized with those of the pretrained AV-VAE and A-VAE, respectively. The parameters of the audio and the video encoders are also initialized with the corresponding parameters in the pretrained A-VAE and V-VAE encoders. Then, all the parameters are fine-tuned using the Adam optimizer [268] with a step size of 10^{-4}, for 100 epochs, and with a batch-size of 128.

We also considered another way to combine A-VAE with V-VAE, in which these two VAE architectures share the same decoder, and they are trained alternately. That is, at each epoch, the shared decoder is trained using latent samples coming from either the encoder of A-VAE or that of V-VAE. As a result, at each epoch, only the encoder parameters of the corresponding VAE, i.e. A-VAE or V-VAE, are updated while those of the other encoder are kept fixed. We refer to the resulting VAE as A\V-VAE.

Speech enhancement parameters For all the methods, the rank of $\mathbf{W}$ and $\mathbf{H}$ in the noise model (VII.45) is set to $K = 10$, and these matrices are randomly initialized with non-negative entries. At the first iteration of the inference algorithms, the Markov chain of the Metropolis-Hastings algorithm (see Section VII.8.2) is initialized by using the noisy

observed speech and the visual features as input to the associated encoders, and taking the posterior mean as the initialization of the latent codes. For the proposed VAE architectures, i.e. MIN-VAE-v1, MIN-VAE-v2, and A\V-VAE, the visual-encoders were used.

VII.9.2 Results and Discussion

Figure VII.5 summarizes the results of all the VAE architectures, in terms of SDR, PESQ, and STOI. The top row of this figure reports the averaged results versus different noise levels, whereas the bottom row shows the averaged results in terms of noise type. From this figure we can see that V-VAE performs pretty well at high noise levels. However, the intelligibility improvements in terms of STOI are not as good as those of the other algorithms. A\V-VAE outperforms other methods in terms of SDR and PESQ. Nevertheless, its intelligibility improvement is not satisfactory. The proposed MIN-VAE methods also outperform A-VAE, especially at high noise levels. As explained earlier, this might be due to the facts that the proposed networks efficiently make use of the robust initialization provided by the visual data, and also by the richer generative models (decoders) which are trained using both audio and visual latent codes. At high noise levels, MIN-VAE-v1 outperforms MIN-VAE-v2, implying the importance of using visual modality in the decoder when the input speech is very noisy. A related observation is that, MIN-VAE-v2 outperforms both MIN-VAE-v1 and AV-VAE when the level of noise is low, implying that the visual features in the generative model contribute mainly in high noise regimes. Part of the worse performance of AV-VAE could be explained by the way the latent codes are initialized, which is based on concatenation of noisy audio and clean visual data. It is worth mentioning that in the low noise regime, the amount of performance improvement is decreasing for all the methods. In fact, it is difficult to enhance a less noisy speech signal.

Regarding noise type, we see that the algorithms perform very differently. The *Babble* noise is the most difficult noise environment according to the bottom row of Fig. VII.5. In terms of SDR, all the methods show their best performance for the *Car* noise, with a very large improvement achieved by the audio-visual based methods. In terms of PESQ, the *White* noise is the easiest one for all the methods, especially A\V-VAE that shows the best performance. Finally, in terms of STOI, MIN-VAE-v1 achieves the best performance for the *Car* noise.

To encourage the proposed MIN-VAE networks to make use of the visual data in the encoder more efficiently, we added some uniform noise to about one-third of speech spectrogram time frames that are fed to the audio encoder of the proposed VAE architectures. Figure VII.6 presents the results of this experiment. A clear performance improvement is observed compared to Fig. VII.5, especially for ME-AVE-v2. With this new training, the proposed algorithms outperform AV-VAE in all noise levels. The SDR improvements for high noise levels, however, are very close. As a conclusion, the best performing algorithm turns out to be MIN-VAE-v2, outperforming A\V-VAE, especially at low levels of noise. Some audio examples are available at `https://team.inria.fr/perception/research/av-vae-se/` and `https://team.inria.fr/perception/research/min-vae-se/`.

VII.10 Conclusions

Inspired by the importance of latent variable initialization for VAE-based speech enhancement, and as another way than simple concatenation to effectively fuse audio and visual modalities in the encoder of VAE, we proposed two VAE-based audio-visual speech enhancement methods. First, AV-VAE which systematically uses both audio and visual information. Second, MIN-VAE which uses a mixture of inference (audio and visual encoder) networks, which are jointly trained with a shared generative network. The respective training algorithms are also provided. After training, we propose to enhance the speech contaminated with an unknown noise type. Therefore, the noise model must be estimate at test time, and two inference procedures (one for each generative speech model) are proposed. The MCEM inference method of MIN-VAE is initialised on the visual modality, which is assumed to be clean in contrast to audio data, as opposed to AV-VAE, which requires to be initialised with audio-visual data. Extensive experiments allowed us to compare various VAE-based speech enhancement models. Some future works include making the proposed algorithms robust to noisy visual data, e.g. by using the mixture idea suggested in [261], incorporating the time dependency between audio and visual time frames by utilizing recurrent layers, and reducing the computational complexity of the inference.

Chapter *VIII*

Conditional Adversarial Networks for Unsupervised Person Re-Identification

Abstract *Unsupervised person re-ID is the task of identifying people on a target data set for which the ID labels are unavailable during training. In this chapter, we propose to unify two trends in unsupervised person re-ID: clustering & fine-tuning and adversarial learning. On one side, clustering groups training images into pseudo-ID labels, and uses them to fine-tune the feature extractor. On the other side, adversarial learning is used, inspired by domain adaptation, to match distributions from different domains. Since target data is distributed across different camera viewpoints, we propose to model each camera as an independent domain, and aim to learn domain-independent features. Straightforward adversarial learning yields negative transfer, we thus introduce a conditioning vector to mitigate this undesirable effect. In our framework, the centroid of the cluster to which the visual sample belongs is used as conditioning vector of our conditional adversarial network, where the vector is permutation invariant (clusters ordering does not matter) and its size is independent of the number of clusters. To our knowledge, we are the first to propose the use of conditional adversarial networks for unsupervised person re-ID. We evaluate the proposed architecture on top of two state-of-the-art clustering-based unsupervised person re-identification (re-ID) methods on four different experimental settings with three different data sets and set the new state-of-the-art performance on all four of them.*

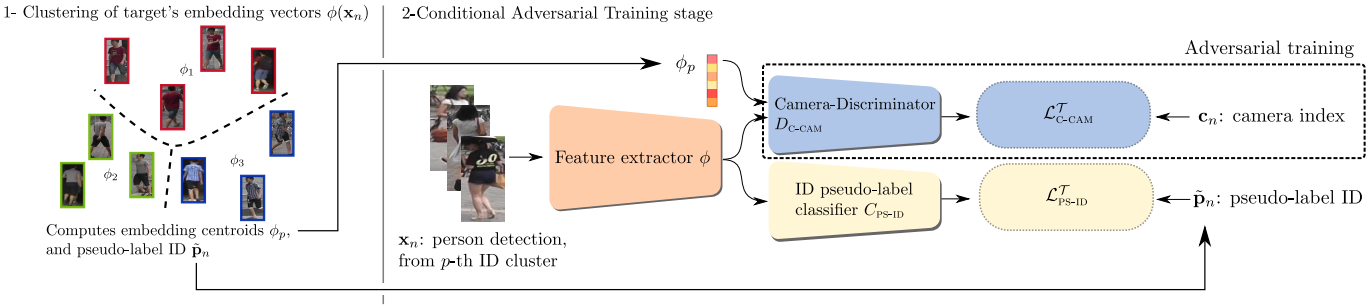


Figure VIII.1: *Pipeline of our method: alternatively (1) clustering target's training data set using ϕ representation, producing noisy pseudo-label ID $\tilde{p}_n$ alongside centroids ϕ_p, and (2) conditional adversarial training, using a Camera-Discriminator D_{CAM} conditioned by ϕ_p to enforce camera invariance on a per identity basis to avoid negative transfer. Pseudo-label ID are used to train an ID classifier C_{PS-ID} alongside the discriminator.*

Chapter Pitch

Methodological contribution A camera-conditional adversarial training framework for robust feature learning.

Applicative task Unsupervised person re-identification.

Interesting insight Setting an adversarial loss based on the cameras directly leads to a negative transfer effect, since the distribution of identities is not uniform w.r.t. the camera. We exploit clustering unsupervised re-id techniques, and condition the camera adversarial network to the "pseudo-label" provided by the clustering algorithm.

Dissemination CANUREID will appear at the International Conference on Pattern Recognition [269].

VIII.1 Introduction

Person re-identification (re-ID) is a well-studied retrieval task [270]–[272] that consists in associating images of the same person across cameras, places and time. Given a query image of a person, we aim to recover his/her identity (ID) from a set of identity-labeled gallery images. The person re-ID task is particularly challenging for two reasons. First, query and gallery images contain only IDs which have never been seen during training. Second, gallery and query images are captured under a variety of background, illumination, viewpoints and occlusions.

Most re-ID models assume the availability of heavily labeled datasets and focus on improving their performance on the very same data sets, see for instance [273], [274]. The limited generalization capabilities of such methods were pointed out in previous literature [275], [276]. In the recent past, researchers attempted to overcome this limitation by investigating a new person re-ID task, where there is a *source* dataset annotated with person IDs and another unlabeled *target* dataset. This is called *unsupervised* person re-ID. Roughly speaking, the current trend is to use a pre-trained base architecture to extract visual features, cluster them, and use the cluster assignments as *pseudo-labels* to re-train the base architecture using standard supervised re-ID loss functions [277], [278].

In parallel, since generative adversarial networks were proposed, adversarial learning has gained popularity in the domain adaptation field [279]–[281]. The underlying intuition is that learning a feature generator robust to the domain shift between *source* and *target* would improve the target performance. The adversarial learning paradigm has been successfully used for person re-ID in both the supervised [282], [283], and the unsupervised [276], [284] learning paradigms.

We propose to unify these two trends in unsupervised person re-ID: hence using conditional adversarial networks for unsupervised person re-ID. Our intuition is that good person re-ID visual features should be independent of the camera/viewpoint, see Fig. 1. Naturally, one would expect that an adversarial game between a generator (feature extractor) and a discriminator (camera classifier) should suffice. However, because the ID presence is not uniform in all cameras, such simple strategy implies some negative transfer and limits – often decreases – the representational power of the visual feature extractor. To overcome this issue, we propose to use conditional adversarial networks, thus providing an additional identity representation to the camera discriminator. Since in the target dataset, the ID labels are unavailable, we exploit the pseudo-labels. More precisely, we provide, as conditioning vector, the centroid of the cluster to which the image belongs. Our contributions are the following:

- We investigate the impact of a camera-adversarial strategy in the unsupervised person re-ID task.
- We realize the negative transfer effect, and propose to use conditional adversarial networks.
- The proposed method can be easily plugged into any unsupervised clustering-based person re-ID methods. We experimentally combine **CANU** with two clustering-based unsupervised person re-ID methods, and propose to use their cluster centroids as conditioning labels.
- Finally, we perform an extensive experimental validation on four different unsupervised re-ID experimental settings and outperform current state-of-the-art methods by a large margin on all settings.

The rest of the chapter is organized as follows. Section VIII.2 describes the state-of-the-art. Section VIII.3 discusses the basics of clustering-based unsupervised person re-ID and sets the notations. The proposed conditional adversarial strategy is presented in Section VIII.4. The extensive experimental validation is discussed in Section VIII.5 before drawing the conclusions in Section VIII.6.

VIII.2 Related work

Unsupervised person re-identification (re-ID) has drawn growing attention in the last few years, taking advantage of the recent achievements of supervised person re-ID models, without requiring an expansive and tedious labeling process of the target data set. A very important line of research starts from a pre-trained model on the source data set and is based on *clustering* and *fine-tuning* [276]–[278], [284], [285]. It alternates between a clustering step generating noisy pseudo-labels, and a fine-tuning step adapting the network to the target data set distribution, leading to a progressive label refinement. Thus, these methods do not use the source data set during adaptation. A lot of effort has been invested in improving the quality of the pseudo-labels. Sampling from reliable clusters during adaptation [276], gradually reducing the number of clusters and merging by exploiting intrinsic inter-ID diversity and intra-ID similarity [284], or performing multiple clustering on visual sub-domains and enforcing consistency [277] have been investigated. More recently, [278] investigated the interaction of two different models to assess and incorporate pseudo-label reliability within a teacher-student framework.

A different approach is directly inspired by Unsupervised Domain Adaptation (UDA) [275], [286]–[290]: using both the source and target data sets during adaptation. These methods aim to match the distributions on the two data sets while keeping its discriminative ability leveraging source ground truth ID labels. A first strategy learns to map source's detections to target's style detections, and train a re-ID model in a supervised setting using those only those transferred detections [275], or in combination with the original target detections [286]. More standard UDA strategies use adversarial learning to match the source and target distributions [280], [288].

Negative transfer has been investigated in unsupervised domain adaptation [291], especially for Partial Domain Adaptation (PDA) [292]–[294], where target labels are only a subset of the source's. Negative transfer is defined as the inability of an adaptation method to find underlying common representation between data sets and is generally caused by the gap between the distributions of the two data sets being too wide [295] for the algorithm to transfer knowledge. Weighting mechanisms are generally employed to remove the impact of source's outliers class on the adaptation process, either for the matching part [293], [294], [296], the classification part [295], or both [292]. Interestingly, [295] uses a domain discriminator conditioned by source label to perform conditional distribution matching. Investigating negative transfer is not limited to UDA settings. For example, a similar method has been proposed for domain generalization [297], implementing a conditional discriminator to match conditioned domain distributions. By doing so, the impact of the difference between prior label distributions on the discriminative ability of the model is alleviated.

Within the task of unsupervised person re-ID, different cameras could be considered as different domains, and standard matching strategies could be used. However, they would inevitably induce negative transfer as described before for generic domain adaptation. Direct application of PDA methods into the person re-ID tasks is neither simple nor expected to be successful. The main reason is that, while PDA methods handle a few dozens of classes, standard re-ID data sets contain a few thousands of IDs. This change of scale requires a different strategy, and we propose to use conditional adversarial networks, with a conditioning label that describes the average sample in the cluster, rather than representing the cluster index. In conclusion, different from clustering and fine-tuning unsupervised person re-ID methods, we propose to exploit (conditional) adversarial networks to learn visual features that are camera independent and thus more robust to appear changes. Different from previous domain adaptation methods, we propose to match domains (cameras) with a conditioning label that evolves during training, since it is the centroid of the cluster to which the visual sample is assigned, allowing us having a representation that is independent of the number of clusters and the cluster index.

VIII.3 Clustering based Unsupervised Person Re-ID

We propose to combine conditional adversarial networks with clustering-based unsupervised person Re-ID. To detail our contributions, we first set up the basics and notations of existing methods for unsupervised person re-ID.

Let $\mathcal{S}$ denote a source ID-annotated person re-ID dataset, containing $N^{\mathcal{S}}$ images corresponding to $M^{\mathcal{S}}$ different person identities captured by $K^{\mathcal{S}}$ cameras. We write $\mathcal{S} = \{(\mathbf{x}_n^{\mathcal{S}}, \mathbf{p}_n^{\mathcal{S}}, \mathbf{c}_n^{\mathcal{S}})\}_{n=1}^{N^{\mathcal{S}}}$, where each three-tuple consists of a detection image, $\mathbf{x}_n^{\mathcal{S}}$, a person ID one-hot vector, $\mathbf{p}_n^{\mathcal{S}} \in \{0,1\}^{M^{\mathcal{S}}}$ and a camera index one-hot vector, $\mathbf{c}_n^{\mathcal{S}} \in \{0,1\}^{K^{\mathcal{S}}}$. Similarly, we define $\mathcal{T} = \{(\mathbf{x}_n^{\mathcal{T}}, \mathbf{c}_n^{\mathcal{T}})\}_{n=1}^{N^{\mathcal{T}}}$ a target person re-ID dataset, with $K^{\mathcal{T}}$ cameras and $N^{\mathcal{T}}$ element, without ID labels.

Source pre-training Let ϕ be a convolutional neural network backbone (*e.g.* ResNet-50 [298]) served as a trainable *feature extractor*. The goal of person re-ID is to be able to discriminate person identities, and therefore an identity classifier C_{ID} is required. The output of C_{ID} is a $M^{\mathcal{S}}$-dimensional stochastic vector, encoding the probability of the input to belong to each of the identities. The cross-entropy and triplet losses are usually employed:

$$\mathcal{L}_{\text{CE}}^{\mathcal{S}}(\phi, C_{\text{ID}}) = -\mathbb{E}_{(\mathbf{x}^{\mathcal{S}}, \mathbf{p}^{\mathcal{S}}) \sim \mathcal{S}} \left\{ \log \left\langle C_{\text{ID}}(\phi(\mathbf{x}^{\mathcal{S}})), \mathbf{p}^{\mathcal{S}} \right\rangle \right\}, \tag{VIII.1}$$

$$\mathcal{L}_{\text{TRI}}^{\mathcal{S}}(\phi) = \mathbb{E}_{(\mathbf{x}^{\mathcal{S}}, \mathbf{x}_p^{\mathcal{S}}, \mathbf{x}_n^{\mathcal{S}}) \sim \mathcal{P}_{\mathcal{S}}} \{ \max(0, \|\phi(\mathbf{x}^{\mathcal{S}}) - \phi(\mathbf{x}_p^{\mathcal{S}})\| + m - \|\phi(\mathbf{x}^{\mathcal{S}}) - \phi(\mathbf{x}_n^{\mathcal{S}})\|) \}, \tag{VIII.2}$$

where $\mathbb{E}$ denotes the expectation, $\langle \cdot, \cdot \rangle$ the scalar product, $\|\cdot\|$ the L^2-norm distance, $\mathbf{x}_p^{\mathcal{S}}$ and $\mathbf{x}_n^{\mathcal{S}}$ are the hardest positive and negative example for $\mathbf{x}^{\mathcal{S}}$ in $\mathcal{P}_{\mathcal{S}}$ the set of all triplets in $\mathcal{S}$, and $m = 0.5$. We similarly denote $\mathcal{L}_{\text{CE}}^{\mathcal{T}}$ and $\mathcal{L}_{\text{TRI}}^{\mathcal{S}}$ the cross-entropy and triplet losses evaluated on the target dataset. However, in unsupervised reID settings, target ID labels are unavailable, and therefore we will need to use alternative *pseudo-ID labels*. The re-ID feature extractor ϕ is typically trained using:

$$\mathcal{L}_{\text{ID}}^{\mathcal{S}}(\phi, C_{\text{ID}}) = \mathcal{L}_{\text{CE}}^{\mathcal{S}}(\phi, C_{\text{ID}}) + \lambda \mathcal{L}_{\text{TRI}}^{\mathcal{S}}(\phi), \tag{VIII.3}$$

for a fixed balancing value λ, achieving competitive performance on the source test set [299]. However, they notoriously lack generalization power and perform badly on datasets unseen during training [275], thus requiring adaptation.

Target fine-tuning As discussed above, target ID labels are unavailable. To overcome this while leveraging the discriminative power of widely-used losses described in Eq. VIII.3, methods like [277], [278] use pseudo-labels. The hypothesis of these methods is that the features learned during the pre-training stage are exploitable for the inference of target's ID labels to a certain extent. Starting from the pre-trained model, these methods alternate between (i) pseudo ID label generation $\{\tilde{\mathbf{p}}_n^{\mathcal{T}}\}_{n=1}^{N^{\mathcal{T}}}$ using a standard clustering algorithm (k-means or DBSCAN [300]) on the target training set $\{\phi(\mathbf{x}_n^{\mathcal{T}})\}_{n=1}^{N^{\mathcal{T}}}$ and (ii) the update of ϕ using losses similar to Eq. VIII.3 supervised by $\{\tilde{\mathbf{p}}_n^{\mathcal{T}}\}_{n=1}^{N^{\mathcal{T}}}$. Since our approach is agnostic to the ID loss used at this step, we choose to denote it by $\mathcal{L}_{\text{PS-ID}}(\phi, C_{\text{PS-ID}})$, $C_{\text{PS-ID}}$ being an optional classifier layer for the pseudo-labels, and develop it further in the experimental section.

VIII.4 Beyond Unsupervised Person Re-ID with Adversarial Networks

In this section we discuss the main limitation of clustering-based unsupervised re-ID methods: we hypothesize that viewpoint variability can make things difficult for clustering methods and propose two alternatives. First, an adversarial network architecture targeting re-ID features that are camera-independent. This strategy could, however, induce some negative transfer when the correlation between cameras and IDs is strong. Second, a conditional adversarial network architecture specifically designed to overcome this negative transfer.

Camera adversarial-guided clustering We hypothesize that camera (viewpoint) variability is one of the major limiting factors for clustering-based unsupervised re-ID methods. In plain, if the embedding space variance explained by camera changes is high, the clustering method could be clustering images from the same camera, rather than images from the same ID. Therefore, ϕ will produce features that can very well discriminate the camera at the expense of the ID. To alleviate this problem, we propose to directly enforce camera invariance in ϕ's representation by using an adversarial strategy, where the discriminator is trained to recognize the camera used to capture the image. Consequently, the generator, in our case ϕ, is trained to remove any trace from the camera index (denoted by $\mathbf{c}$). Intuitively, this should reduce the viewpoint variance in the embedding space, improve pseudo-labels quality and increase the generalization ability of ϕ to unseen IDs.

To do so, we require a camera discriminator D_{CAM} (see Fig. VIII.1 for a complete overview of the architecture). The generator ϕ and the discriminator D_{CAM} will be trained through a min-max formulation:

$$\min_{\phi, C_{\text{PS-ID}}} \max_{D_{\text{CAM}}} \mathcal{L}_{\text{PS-ID}}^{\mathcal{T}}(\phi, C_{\text{PS-ID}}) - \mu \mathcal{L}_{\text{CAM}}^{\mathcal{T}}(\phi, D_{\text{CAM}}), \tag{VIII.4}$$

where $\mu > 0$ is a balance hyper-parameter that can be interpreted as a regularization parameter [280], and $\mathcal{L}_{\text{CAM}}^{\mathcal{T}}$ is defined via the cross-entropy loss:

$$\mathcal{L}_{\text{CAM}}^{\mathcal{T}}(\phi, D_{\text{CAM}}) = -\mathbb{E}_{(\mathbf{x}^{\mathcal{T}}, \mathbf{c}^{\mathcal{T}}) \sim \mathcal{T}} \left\{ \log \left\langle D_{\text{CAM}}(\phi(\mathbf{x}^{\mathcal{T}})), \mathbf{c}^{\mathcal{T}} \right\rangle \right\} \tag{VIII.5}$$

On one side, the feature extractor ϕ must minimize the person re-ID loss $\mathcal{L}_{\text{PS-ID}}$ at the same time as making the problem more challenging for the camera discriminator. On the other side, the camera discriminator tries to learn to recognize the camera corresponding to the input image.

Adversarial negative transfer It has been shown [297] that minimizing (VIII.4) is equivalent to the following problem:

$$\begin{aligned} &\min_{\phi, C_{\text{PS-ID}}} \mathcal{L}_{\text{PS-ID}}^{\mathcal{T}}(\phi, C_{\text{PS-ID}}) \\ &\text{s.t. } \text{JSD}_{\mathcal{T}}(p(\phi(\mathbf{x})|\mathbf{c}=1), \dots, p(\phi(\mathbf{x})|\mathbf{c}=K)) = 0, \end{aligned} \tag{VIII.6}$$

where $\text{JSD}_{\mathcal{T}}$ stands for the multi-distribution Jensen-Shanon divergence [301] on the target set $\mathcal{T}$, and we dropped the superscript $\mathcal{T}$ in the variables to ease the reading.

Since the distribution of ID labels may strongly depend on the camera, the plain adversarial strategy in (VIII.6) can introduce negative transfer [295]. Formally, since we have:

$$p(\mathbf{p}|\mathbf{c}=i) \neq p(\mathbf{p}|\mathbf{c}=j), i \neq j$$

then solving (VIII.6) is not equivalent (see [297]) to:

$$\begin{aligned} &\min_{\phi, C_{\text{PS-ID}}} \mathcal{L}_{\text{PS-ID}}^{\mathcal{T}}(\phi, C_{\text{PS-ID}}) \\ &\text{s.t. } \text{JSD}_{\mathcal{T}}(p(\phi(\mathbf{x})|\mathbf{p}, \mathbf{c}=1), \dots, p(\phi(\mathbf{x})|\mathbf{p}, \mathbf{c}=K)) = 0, \end{aligned} \tag{VIII.7}$$

which is the problem we would implicitly want to solve. Intuitively, *negative transfer* means that the camera discriminator learns $p(\mathbf{c}|\mathbf{p})$ instead of $p(\mathbf{c}|\mathbf{x}, \mathbf{p})$, exploiting ID to infer camera information and decreasing the representation power of ϕ due to the adversarial loss.

Conditional adversarial networks We propose to directly solve the optimization problem in Eq. VIII.7 to alleviate the negative transfer. Similar to the original conditional GAN formulation [302], we condition the adversarial discriminator with the input ID $\mathbf{p}$. Given that ID labels are unavailable on the target set, we replace them by the pseudo-labels obtained during the clustering phase.

However, since we are handling a large number of IDs (700 to 1500 in standard re-ID datasets), using a one-hot representation turned out to be very ineffective. Indeed, such representation is not permutation-invariant, meaning that if the clusters are re-ordered, the associated conditional vector changes, which does not make sense. We, therefore, need a permutation-invariant conditioning label.

To do so, we propose to use the cluster centroids $\phi_{\mathbf{p}}$ which are provided by the clustering algorithms at no extra cost. This conditioning vectors are permutation invariant. Importantly, we do not back-propagate the adversarial loss through the ID-branch, to avoid using an ID-dependant gradient from the adversarial loss. This boils down to defining $\mathcal{L}_{\text{C-CAM}}$ as:

$$\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi, D_{\text{C-CAM}}) = -\mathbb{E}_{(\mathbf{x},\mathbf{p},\mathbf{c})\sim\mathcal{T}}\left\{\log\left\langle D_{\text{C-CAM}}(\phi(\mathbf{x}), \phi_{\mathbf{p}}), \mathbf{c}\right\rangle\right\} \tag{VIII.8}$$

and then solving:

$$\min_{\phi, C_{\text{PS-ID}}} \max_{D_{\text{C-CAM}}} \mathcal{L}^{\mathcal{T}}_{\text{PS-ID}}(\phi, C_{\text{PS-ID}}) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi, D_{\text{C-CAM}}). \tag{VIII.9}$$

VIII.5 Experimental Validation

In this section, we provide implementation details and an in-depth evaluation of the proposed methodology, setting the new state-of-the-art in four different unsupervised person re-ID experimental settings. We also provide an ablation study and insights on why conditional adversarial networks outperform existing approaches.

VIII.5.1 Evaluation Protocol

We first describe here the baselines, on which our proposed **CANU** is built and tested. The used datasets and the evaluation metrics are then introduced.

Baselines The proposed **CANU** can be easily plugged into any clustering-based unsupervised person re-ID methods. Here, we experimentally test it on two state-of-the-art clustering-based unsupervised person re-ID methods, as baselines.

First, self-similarity grouping [277] (**SSG**) performs independent clustering on the upper-, lower- and full-body features, denoted as ϕ^{U}, ϕ^{L} and ϕ^{F}. They are extracted from three global average pooling layers of the convolutional feature map of ResNet-50 [298]. The underlying hypothesis is that noisy global pseudo-label generation can be improved by using multiple, but related clustering results, and enforcing consistency between them. The triplet loss is used to train the overall architecture.

To implement **CANU**-SSG, we define three different camera discriminators, one for each embedding, $D^{\text{U}}_{\text{C-CAM}}$, $D^{\text{L}}_{\text{C-CAM}}$ and $D^{\text{F}}_{\text{C-CAM}}$ respectively, each fed with samples from the related representation and conditioned by the global embedding ϕ^{F}. In the particular case of **CANU**-SSG, the generic optimisation problem in Eq. VIII.9 instantiates as:

$$\min_{\phi} \max_{D^{\text{U,L,F}}_{\text{C-CAM}}} \mathcal{L}^{\mathcal{T}}_{\text{SSG}}(\phi) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi^{\text{U}}, D^{\text{U}}_{\text{C-CAM}}) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi^{\text{L}}, D^{\text{L}}_{\text{C-CAM}}) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi^{\text{F}}, D^{\text{F}}_{\text{C-CAM}}). \tag{VIII.10}$$

Second, Mutual Mean-Teaching [278] (**MMT**) reduces pseudo-label noise by using a combination of hard and soft assignment: using hard labeling reduces the amount of information given to the model, and using soft labeling allows the cluster's confidence to be taken into account. MMT defines two different models $(\phi^1, C^1_{\text{PS-ID}})$ and $(\phi^2, C^2_{\text{PS-ID}})$, both implemented with a IBN-ResNet-50 [303] backbone, initialized with two different pre-trainings on the source dataset. They are then jointly trained using pseudo labels as hard assignments, and inspired by teacher-student methods, using their own pseudo ID predictions as soft pseudo-labels to supervise each other. Soft versions of cross-entropy and triplet loss are used.

To implement **CANU**-MMT, similar to **CANU**-SSG, we define two camera discriminators $D^1_{\text{C-CAM}}$ and $D^2_{\text{C-CAM}}$, each dedicated to one embedding, and train it using the following instantiation of the generic optimisation problem in Eq. VIII.9:

$$\min_{\phi^{1,2}, C^{1,2}_{\text{PS-ID}}} \max_{D^{1,2}_{\text{C-CAM}}} \mathcal{L}^{\mathcal{T}}_{\text{MMT}}(\phi^1, C^1_{\text{PS-ID}}) + \mathcal{L}^{\mathcal{T}}_{\text{MMT}}(\phi^2, C^2_{\text{PS-ID}}) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi^1, D^1_{\text{C-CAM}}) - \mu\mathcal{L}^{\mathcal{T}}_{\text{C-CAM}}(\phi^2, D^2_{\text{C-CAM}}). \quad \text{(VIII.11)}$$

While the clustering strategy used in SSG is DBSCAN [300], the one used in MMT is standard k-means. For a fair comparison, we implemented **CANU** with DBSCAN, which has the advantage of automatically selecting the number of clusters. We also evaluate the performance of MMT using the DBSCAN clustering strategy without **CANU**, to evaluate the impact of our method on a fair basis.

Datasets The proposed adversarial strategies are evaluated using three datasets: Market-1501 (Mkt) [274], DukeMTMC-reID (Duke) [273] and MSMT17 (MSMT) [304]. In all three cases, the dataset is divided into three parts: training, gallery, and query. The query and the gallery are never available during training and only used for testing.

Mkt is composed of $M = 1,501$ (half for training and half for testing) different identities, observed through $K = 6$ different cameras (viewpoints). The deformable parts model [305] is used for person detection. As a consequence, there are $N = 12,936$ training images and $19,732$ gallery images. In addition, there are $3,368$ hand-drawn bounding box queries.

Duke is composed of $M = 1,404$ (half for training and half for testing) identities captured from $K = 8$ cameras. In addition, 408 other ID, called "distractors", are added to the gallery. Detections are manually selected, leading to $N = 16,522$ images for train, $17,661$ for the gallery and $2,228$ queries.

MSMT is the largest and most competitive dataset available, with $M = 4,101$ identities ($1,041$ for training, and $3,060$ for test), $K = 15$ cameras, with $N = 32,621$ images for training, $82,161$ for the Gallery and $11,659$ queries.

The unsupervised person re-ID experimental setting using dataset A as source and dataset B as the target is denoted by A ► B. We compare the proposed methodology in four different settings: Mkt ► Duke, Duke ► Mkt, Mkt ► MSMT and Duke ► MSMT.

Evaluation metrics In order to provide an objective evaluation of the performance, we employ two standard metrics in person re-ID [274]: Rank-1 (R1) and mean average-precision (mAP). Precisely, for each query image, we extract visual features employing ϕ, and we compare them to the features extracted from the gallery using the cosine distance. Importantly, the gallery images captured with the same camera as the query image are not considered. For R1, a query is well identified if the closest gallery feature vector corresponds to the same identity. In the case of mAP, the whole list of gallery images is considered, and precision at different ranking positions is averaged. See [274] for details. For both metrics, the mean over the query set is reported.

Implementation details For both MMT and SSG, we use the models pre-trained on the source datasets (e.g. For Mkt ►Duke, we use the model pre-trained on the Market dataset and provided by [277] and [278]). DBSCAN is used at the beginning of each training epoch, the parameters for DBSCAN are the same described as in [277]. The weight for (conditional) adversarial losses μ is set to 0.1 for MMT and to 0.05 for SSG, chosen according to a grid search with values between $[0.01, 1.8]$ (see below). The used conditional discriminator has two input branches, one as the (conditional) ID branch and the other is the camera branch, both consist of four fully-connected layers, of size $[2048, 1024]$, $[2048, 1024]$, $[1024, 1024]$, $[1024,$ number of cameras$]$, respectively. Batch normalization [306] and ReLU activation are used. For MMT, during the unsupervised learning, we train the IBN-ResNet-50 [303] feature extractor with Adam [307] optimizer using a learning rate of 0.00035. As default in [278], the network is trained for 40 epochs but with fewer iterations per epoch (400 v.s. 800 iterations) while keeping a similar or better performance. For SSG, we train the ResNet-50 [298] with SGD optimizer using a learning rate of $6e$-5. At each epoch, unlike MMT, we iterate through the whole training set instead of training with a fix number of iterations.

After training, the discriminator is discarded and only the feature extractor is kept for evaluations. For SSG, first, it combines the features extracted from the original image and the horizontally flipped image with a simple sum. Second, the summed features are normalized by their L_2 norm. Finally, The full-, upper- and, lower-body normalized features are concatenated to form the final features. For MMT, the features extracted from the feature extractor are directly used for evaluations. Our code and model will be made publicly available at `https://team.inria.fr/perception/canu-reid/`.

In the following, we first compare the proposed methodology with the state-of-the-art (see Sec. VIII.5.2). Secondly, we discuss the benefit of using conditional camera-adversarial training in the ablation study (see Sec. VIII.5.3), and include several insights on the performance of **CANU**.

Table VIII.1: *Comparison of the proposed **CANU** methodology on the Mkt ► Duke and Duke ► Mkt unsupervised person re-ID settings. **CANU-MMT** establishes a new state-of-the-art in both settings, and **CANU-SGG** outperforms **SSG**.*

Method	Mkt ► Duke		Duke ► Mkt	
	R1	mAP	R1	mAP
PUL [276]	30.0	16.4	45.5	20.5
TJ-AIDL [308]	44.3	23.0	58.2	26.5
SPGAN [275]	41.1	22.3	51.5	22.8
HHL[286]	46.9	27.2	62.2	31.4
CFSM [287]	49.8	27.3	61.2	28.3
BUC [284]	47.4	27.5	66.2	38.3
ARN [309]	60.2	33.4	70.3	39.4
UDAP [289]	68.4	49.0	75.8	53.7
ENC [290]	63.3	40.4	75.1	43.0
UCDA-CCE [288]	47.7	31.0	60.4	30.9
PDA-Net [310]	63.2	45.1	75.2	47.6
PCB-PAST [285]	72.4	54.3	78.4	54.6
Co-teaching [311]	77.6	61.7	87.8	71.7
SSG [277]	73.0	53.4	80.0	58.3
CANU-SSG (ours)	76.1	57.0	83.3	61.9
MMT [278]	81.8	68.7	91.1	74.5
MMT (DBSCAN)	80.2	67.2	91.7	79.3
CANU-MMT (ours)	**83.3**	**70.3**	**94.2**	**83.0**

Table VIII.2: *Comparison of the proposed **CANU** methodology on the Mkt ► MSMT and Duke ► MSMT unsupervised person re-ID settings. **CANU-MMT** establishes a new state-of-the-art in both settings, and **CANU-SGG** outperforms **SSG**.*

Method	Mkt ► MSMT		Duke ► MSMT	
	R1	mAP	R1	mAP
PTGAN [312]	10.2	2.9	11.8	3.3
ENC [290]	25.3	8.5	30.2	10.2
SSG [277]	31.6	13.2	32.2	13.3
CANU-SSG (ours)	45.5	19.1	43.3	17.9
MMT [278]	54.4	26.6	58.2	29.3
MMT (DBSCAN)	51.6	26.6	59.0	32.0
CANU-MMT (ours)	**61.7**	**34.6**	**66.9**	**38.3**

VIII.5.2 Comparison with the State-of-the-Art

We compare **CANU**-SSG and **CANU**-MMT to the state-of-the-art methods and we demonstrate in Tables VIII.1 and VIII.2 that **CANU**-MMT sets a new state-of-the-art result compared to the existing unsupervised person re-ID methods by a large margin. In addition, **CANU**-SSG outperforms SSG in all settings. Since the MSMT dataset is more recent, fewer comparisons are available in the experiments involving this dataset, hence the two different tables.

More precisely, the proposed **CANU** significantly improves the performance of the baselines, SSG [277] and MMT [278]. In Mkt ►Duke and Duke ►Mkt (Table VIII.1), **CANU**-SSG improves SSG by ↑3.1%/↑3.6% (R1/mAP, same in the following.) and ↑3.3%/↑3.6% respectively, and **CANU**-MMT significantly outperforms MMT by ↑1.5%/↑1.6% and ↑3.1%/↑8.5% respectively. Moreover, for the more challenging setting (Table VIII.2), the improvement brought by **CANU** is even more evident. For SSG, for example, we increase the R1/mAP by ↑13.9%/↑5.9% in Mkt ►MSMT, and by ↑11.1%/↑4.6% in Duke ►MSMT. For MMT, **CANU**-MMT outperforms MMT by ↑7.3%/↑8.0% in Mkt ►MSMT, and by ↑8.7%/↑9.0% in Duke ► MSMT. Finally, the consistent improvement in the four settings of **CANU**-MMT over MMT (DBSCAN) and the inconsistent improvement of MMT (DBSCAN) over standard MMT proves that the increase of the performance is due to the proposed methodology. To summarize, we greatly improve the baselines using the proposed **CANU**. More importantly, to our best knowledge, we outperform the existing methods by a large margin and establish a new state-of-the-art result.

Table VIII.3: *Impact of μ in the performance of **CANU**. When the mAP values are equal, we highlight the one corresponding to higher R1.*

Method	μ	Mkt ▶ Duke		Duke ▶ Mkt	
		R1	mAP	R1	mAP
CANU-SSG	0.01	72.8	53.3	79.7	57.2
	0.05	**76.1**	**57.0**	**83.3**	**61.9**
	0.1	74.7	56.2	82.7	61.1
	0.2	75.3	56.5	81.8	60.3
	0.4	73.3	53.5	80.4	59.2
	1.8	7.1	2.9	39.1	17.1
CANU-MMT	0.01	81.3	68.9	92.6	79.2
	0.05	82.4	70.3	93.0	81.3
	0.1	**83.3**	**70.3**	**94.2**	**83.0**
	0.2	82.7	70.3	93.4	82.5
	0.4	82.5	70.3	93.8	82.0
	1.8	82.8	69.9	93.1	81.3

Table VIII.4: *Evaluation of the impact of the conditional strategy on SGG [277] and MMT [278] (using DSCAN). When the mAP values are equal, we highlight the one corresponding to higher R1.*

Method	Mkt ▶ Duke		Duke ▶ Mkt	
	R1	mAP	R1	mAP
SSG [277]	73.0	53.4	80.0	58.3
SSG+Adv.	75.4	56.4	**83.8**	**62.7**
CANU-SSG	**76.1**	**57.0**	83.3	61.9
MMT (DBSCAN)	80.2	67.2	91.7	79.3
MMT+Adv.	82.6	70.3	93.6	82.2
CANU-MMT	**83.3**	**70.3**	**94.2**	**83.0**

VIII.5.3 Ablation Study

In this section, we first perform a study to evaluate the impact of the value of μ. Secondly, we demonstrate the interest of the conditional strategy, versus its non-conditional counterpart. Thirdly, we study the evolution of the mutual information between ground-truth camera indexes and pseudo-labels using MTT (DBSCAN), thus providing some insights on the quality of the pseudo-labels and the impact of the conditional strategy on it. Finally, we visualize the evolution of the number of lost person identities at each training epoch, to assess the impact of the variability of the training set.

Selection of μ We ablate the value μ by comparing the performance (R1 and mAP) of models trained within the range $[0.01, 1.8]$. From Tab. VIII.3, $\mu = 0.1$ (**CANU**-MMT) and $\mu = 0.05$ (**CANU**-SSG) yield the best person re-ID performance.

Is conditional necessary? From Table VIII.4, we show that the camera adversarial network can help the person re-ID networks trained with clustering-based unsupervised methods better capture the person identity features: **CANU** and adding a simple adversarial discriminator (+Adv.) significantly outperform the baseline methods in all settings. This is due to the combination of the camera adversarial network with unsupervised clustering-based methods. By doing so, the camera dependency is removed from the features of each person thus increasing the quality of the overall clustering. However, because of the negative transfer effect, the camera adversarial network cannot fully exploit the camera information while discarding the person ID information. For this reason, the proposed method **CANU** improves the capacity of the camera adversarial network over the simple adversarial strategy. In summary, we demonstrate that the camera adversarial network can help improve the results of unsupervised clustering-based person re-ID. Moreover, the proposed **CANU** further improves the results by removing the link between camera and IDs.

Removing camera information Table VIII.4 demonstrates that removing camera information is globally positive, but that can also be harmful if it is not done with care. In this section, we further demonstrate that the proposed

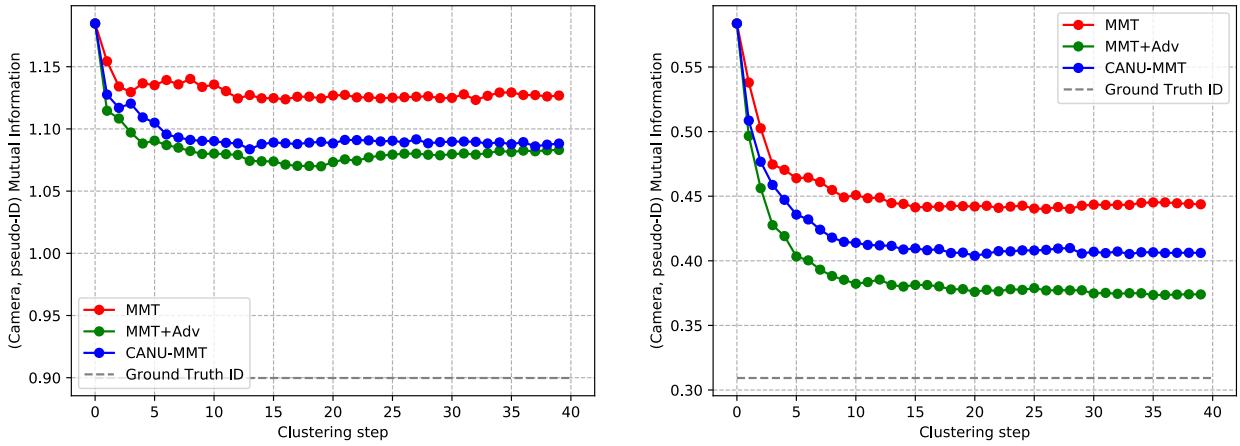


Figure VIII.2: *Mutual information between pseudo labels and camera index evolution for the MMT setting. Ground-truth ID comparison is displayed in dashed lines for both datasets: Mkt ► Duke (left) and Duke ► Mkt (right).*

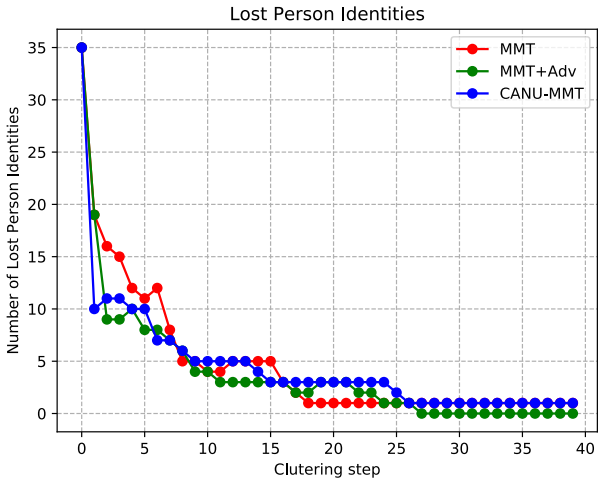


Figure VIII.3: *Evolution of the number of lost person IDs during training using MMT on Duke ► Mkt.*

adversarial strategies actually reduce the camera dependency in clustering results and present some insights on why the conditional strategy is better than the plain adversarial network. To do so, we plot the mutual information between the pseudo-labels provided by DBSCAN, and the fixed camera index information, at each clustering stage (i.e. training epoch) in Fig. VIII.2. Intuitively, the mutual information between two variables is a measure of mutual dependence between them: the higher it is, the more predictable one is from knowing the other. We report the results for MMT on Duke ► Mkt and Mkt ► Duke, **CANU**-MMT and the simple adversarial strategy. We observe that the mutual information is systematically decreasing with the training, even for plain MMT. Both adversarial strategies significantly outperform plain MMT at reducing the camera-pseudo-ID dependency, **CANU**-MMT being slightly less effective than MMT+Adv. This is consistent with our theoretical framework, since matching ID-conditioned camera distribution in ϕ does not account for the ID-Camera dependency, and thus is less effective in terms of camera dependency, but preserves identity information, see Table VIII.4. We also observe that there is a significant gap between the target mutual information (i.e. measured between ground truth ID and camera index) for all methods, which exhibits the performance gap between supervised and unsupervised person re-ID methods.

Evolution of the number of lost IDs Since we train the target dataset using unsupervised techniques, we do not use the ground-truth labels in the target dataset during training. Instead, we make use of the pseudo labels provided by DBSCAN. DBSCAN discards the outliers i.e. features that are not closed to others. It is natural to wonder how many identities are "lost" at every iteration. We here visualize the number of lost ID (all those that are not present in a training epoch) after each clustering step. We plot the evolution of this number with the training epoch for MMT, MMT+Adv. and **CANU**-MMT on Duke ► Mkt in Fig. VIII.3. The dual experiment, i.e. on Mkt ► Duke revealed that no ID was lost by any method. In Fig. VIII.3, we first observe that the loss of person identities decreases with the clustering steps. It means that the feature extractor provides more and more precise features representing person identities. Secondly, the use of camera adversarial training can reduce the loss of person identities in the clustering algorithm,

which reflects the benefit of camera adversarial networks to the clustering algorithm and thus to the unsupervised person re-ID task.

VIII.6 Conclusions

We demonstrate the benefit of unifying adversarial learning with current unsupervised clustering-based person re-identification methods. We propose to condition the adversarial learning with the cluster centroids, being these representations independent of the number of clusters and invariant to cluster index permutations. The proposed strategy boosts existing clustering-based unsupervised person re-ID baselines and sets the new state-of-the-art performance in four different unsupervised person re-ID experimental settings. We believe that the proposed method **CANU** was a missing component in training unsupervised person re-identification networks and we hope that our work can give insight to this direction in the person re-identification domain.

Chapter *IX*

Conclusion and Future Research Opportunities

IX.1 Summary

In this manuscript we discussed several approaches allowing to endow a robot – or an autonomous system – with social intelligence. As presented in the introduction, we priviledge probabilistic generative models processing auditory and visual data. However, when required, we focus on developing mono-modal approaches. Very importantly, all these tasks have an underlying common motivation: they are building blocks useful to provide social intelligence to robotic platforms. Chapter I describes the motivation of this global research direction, as well as the main methodological tools and the phylosophy behind. Inthere, we discussed probabilistic generative models, those with exact and approximate inference, as well as the interest of learning from audio-visual data. After, we presented several contributions that are here summarised.

Robust clustering for audio-visual speaker detection is discussed in Chapter II. To that aim, a weighted-data Gaussian mixture model is proposed with two variants. The first one deals with fixed weights, while the second one models the weights as random variables. In both cases, we present an exact expectation-maximisation algorithm. A minimum message length criterion is derived to automatically infer the number of clusters – or speakers. Experiments on standard clustering datasets as well as on audio-visual speakers data are presented and discussed. The detection of speakers in the scene, together with their speaking status, is one of the most basic tasks to build social intelligence.

Chapter III introduces a generic framework to adapt a regression model between two modalities to a new distribution of one of the modalities. Gaussian mixture models are selected to address this problem, and are experimentally validated with the acoustico-articulatory task. Even if this task does not directly apply to social robotics, it is interesting to be able to adapt a map between two modalities when there are changes in the data distribution of one modality. For instance, one can imagine to adapt a previously learned audio-visual mapping, to the accoustic conditions of a different room. Two version of the regression adaptation method are presented, differing on a probabilistic link between two of the random variables. The respective EM-algorithms are discussed, together with the relationship with between the two GMM-based models.

A probabilistic model for robust deep regresion is introduced in Chapter IV. Classical deep regression methods are typically trained with the Euclean (distance) loss. For carefully annotated datasets, this is an appropriate loss function to train a neural network for regression. However, curating large-scale datasets is very costly, and some times a chimeric task. As a consequence, the community often relies in automatic annotation procedures and the final dataset is prone to annotation errors. Ideally, one would like an automatic method to clean the training set at the same time as (or prior to) training the deep neural network. We propose a mixture model consisting on a zero-centered Gaussian plus a uniform distribution. While the Gaussian models the variance of the correctly annotated samples, the uniform distribution allows to model large regression errors due to the annotation mistakes. The mixture model is trained together with the network. The responsibilities computed with the E-step of the EM algorithm are then exploited to weight the Euclidean loss used to train the neural network. In this way, the identification of outliers is unsupervised, and the neural network is supervised only with clean data.

After three chapters using probabilistic models with exact inference procedures, the second part of the manuscript discusses models with approximate inference algorithms. Firstly, we discuss a variational EM algorithm for tracking multiple speakers with audio-visual data in Chapter V. We model the dynamics with a Gaussian distribution, and propose a model able to automatically assign the many auditory and visual observations to the sources (persons). To that aim, observation-to-source latent variables are defined, and together with the position of the persons form the set of hidden random variables. The posterior distribution of these latent variables is computationally intractable, and we propose to approximate it with a separable distribution. The associated VEM consists on alternating a frame-wise E-step of a GMM with several Kalman filters in parallel and the M-step. Additionally we managed to assess when each of the speakers is active thanks to the audio-to-person assingment variables.

After, in Chapter VI, we discuss the use of conditional random fields for deep pixel-level inference. We re-introduce the concept of gates within a probabilistic model on the top of a deep neural network. After proposing and deriving the associated variational inference procedure, we can reinterpret the gates as attention variables. In addition, the inference procedure is implemented within the neural network, allowing a fine merge between the probabilistic model and the computational flow of a deep net.

Chapter VII discusses the use of variational autoencoders for speech enhancement through the exploitation of audio-visual data. Two models are presented: one systematically using auditory and visual data, and the second, automatically finding the optimal mix between the two modalitis to enhance the trained model. For each of the models, an associated EM-like training procedure is discussed at enhancement time. Indeed, the models we proposed are inspired from the unsupervised speech enhancement literature, meaning that the noise type is not known during training time. Therefore, we use a noise model that can be estimated at the same time as the speech signal is enhanced.

Finally, an adversarial strategy for unsupervised person re-identification is discussed in Chapter VIII. The intuition behind the method introduced in this chapter is to develop a visual representation for person re-id that is camera-independent. Simply applying an adversarial game between a feature generator and a discriminator trying to recognise which camera was the image taken with, is not a successful strategy. This is because there is a negative transfer effect between the ID label and the camera label. We therefore provide the ID label to the network so as to palliate with this negative transfer effect. In addition, since we aim to propose a method working in the unsupervised settings, we cannot have access to any ID labels in the target dataset/task. The direct application of a conditional adversarial network is not possible. We exploit recent advances in culstering-based person re-id, allowing us to use the cluster membership as ID pseudo-labels.

IX.2 Conclusions

Overall, the work presented in this manuscript leads to several conslusions. First, even if the raise of deep neural networks allowed great advances in several tasks, there are many others for which the combination of deep networks with other frameworks – probabilistic models in particular – provides a good balance between representation power and robustness to clutter. Importantly, probabilistic models also allow to exploit and account for uncertainty. Therefore, combinations of probabilistic models and deep networks seem to be a good methodological framework when dealing with realistic environments, for instance then ones derived from robotic applications. Another advantage of probabilistic models is that they allow for a certain level of interpretability when, for instance, fusing information from different modalities or discarding data points corrupted by noise.

Secondly, the fusion of auditory and visual data is a challenging research field, in which one tries to successfully exploit the complementary nature of the two modalities. On the one hand, auditory data perceives sources standing everywhere, and merges them by summing their corresponding signals. On the other hand, video data perceives only within the camera(s) field of view, and merges the sources by superposition. Subsequently, beyond noise and clutter, one of the main differences between audio and video is that while the audio signal within a time interval is the combination of all active sound sources, the video signal will mostly contain sources on the foreground, since the ones in the background will be occluded. The differences between the audio and video modalities provide the right frame for scientific research: their fusion is as interesting as it is challenging.

Thirdly, developing learning models for robotic platforms is also motivating and not straightforward. While one may perform computationally expensive tasks off-line, we must also be sure that the inference algorithms at test (runtime) are light enough to fit the computational resources of the robotic platform. This constraint is difficult to respect, since many of the high performance models used to process auditory and/or visual data are computationally very costly. Therefore, the path towards socially intelligent robots does not only require the design of smart strategies for audio-visual fusion, but also the derivation of light-weight algorithms for inference and on-line model update.

Finally, in this manuscript we have only discussed models, tools and algorithms for robotic perception. However, a robot cannot be socially intelligent without taking socially pertinent actions. How to learn action policies that are socially pertinent is out of the scope of my previous research, while it will be one of my future research guidelines.

IX.3 Future Research Opportunities

In the future we will investigate several directions of research. First a purely methodological direction based on deep probabilistic models, then how to exploit these kind of methods to continue our research on robot perception, as well as take the risk and investigate learning optimal action policies for socially pertinent robots.

IX.3.1 Deep probabilistic models

Among the most common tools for processing sequences of data, there are probabilistic state-space models and recurrent networks. While the later can effective learn complex temporal patterns, the former can exploit uncertainty through time. VAE are a third class of methods able to model uncertainty through the use of deep architectures. To overcome the limitations of this three classes of methods, a few recent studies were published at the crossroads of deep recurrent networks and of probabilistic models, and we reviewed them in a recent preprint [313]. In this monograph, we have also proposed a new class of models, that is an umbrella of the recent literature on the topic, unified the notation, and proposed several interesting future research lines. We termed this class of models Dynamical Variational Autoencoders, or DVAE. In a sentence, this means that we aim to model a recurrent process and its uncertainty by means of deep neural networks and probabilistic models. We name the big family of all these methods as "Deep Probabilistic Models."

Learning deep probabilistic models is challenging from the theoretical, methodological and computational points of view. Indeed, the problem of learning, for instance, deep generative filters in the framework of nonlinear and non-Gaussian SSMs remains intractable and approximate solutions, that are both optimal from a theoretical point of view and efficient from a computational point of view, remain to be proposed. We plan to investigate both discriminative and generative deep recurrent networks and to apply them to audio, visual and audio-visual processing tasks.

- *Discriminative deep filters*. We plan to address challenging problems associated with the temporal modeling of human-behavior recognition. In particular we plan to devise novel algorithms to robustly track visual focus of attention, eye-gaze, head-gaze, facial expressions, lip movements, as well as hand and body gestures. These tasks require end-to-end learning, from the detection of facial and body landmarks to the prediction of their trajectories and activity recognition. In particular, we will address the task of characterizing temporal patterns of behavior in flexible settings, e.g. users not facing the camera. For example, lip reading for speech enhancement and speech recognition must be performed in unconstrained settings, e.g. in the presence of rigid head motions or when the user's face is partially occluded.

- *Generative recurrent deep networks*. Most of the VAE-based methods in the literature are tailored to use unimodal data. VAE models for multimodal data are merely available and we are among the first to propose an audio-visual VAE model for speech enhancement [261]. Nevertheless, the proposed framework treats the two modalities unevenly. We started to investigate the use of mixture models in an attempt to put the two modalities on an equal footing [198], [199]. However, this is a long term endeavor since it raises many difficult questions from both theoretical and algorithmic points of view. Indeed, while the concept of noisy speech is well formalized in the audio signal processing domain, it is not understood in the computer vision domain. We plan to thoroughly address the combination of generative deep networks with robust mixture modeling. We plan to address the added complexity in the framework of variational approximation, possibly using robust probability distributions. Eventually, we would like to combine VAEs with RNNs. As already mentioned, we started to investigate this problem in the framework of our work on speech enhancement [314], which may be viewed either as a recurrent VAE or, more generally, as a non-linear DNN-based formulation of SSMs. We will apply this kind of deep generative/recurrent architectures to other problems that are encountered in audio-visual perception and we will propose case-by-case tractable and efficient solvers.

IX.3.2 Human behavior understanding

Interactions between a robot and a group of people require human behavior understanding (HBU) methods. Consider for example the tasks of detecting eye-gaze and head-gaze and of tracking the gaze directions associated with a group of participants. This means that, in addition to gaze detection and gaze tracking, it is important to detect persons and to track them as well. Additionally, it is important to extract segments of speech, to associate these segments with persons and hence to be able to determine over time who looks to whom and who is the speaker and who are the listeners. The temporal and spatial fusion of visual and audio cues stands at the basis of understanding social roles and of building a multimodal conversational model.

We propose to perform audio-visual HBU by taking explicitly into account the complementary nature of the audio and video modalities. Indeed, in face-to-face communication, the robot must choose with whom it should engage dialog, e.g. based on proximity, eye gaze, head movements, lip movements, facial expressions, etc., in addition to speech. Unlike in the single-user human-robot interaction case, it is crucial to associate temporal segments of speech with participants and hence to combine speech diarization with spoken dialog. Under such scenarios, speech signals are perturbed by noise, reverberation and competing audio sources, hence speech localization and speech enhancement methods must be used in conjunction with speech recognition and with spoken dialog.

- *Deep visual descriptors.* One of the most important ingredients of HBU is to learn visual representations of humans using deep discriminative networks. This process comprises detecting people and body parts in images and then extracting 2D or 3D landmarks. We plan to combine body landmark detectors and facial landmark detectors, based on feedforward architectures, with landmark tracking based on recurrent neural networks. The advantage is twofold: to eliminate noise, outliers and artefacts, which are inherent to any imaging process, and to build spatio-temporal representations for higher-level processes such as action and gesture recognition. While the task of noise filtering can be carried out using existing techniques, the task of removing outliers and artefacts is more difficult. Based on our recent work on robust deep regression, we plan to develop robust deep learning methods to extract body and facial landmarks. In addition to the Gaussian-uniform mixture used in [80], we plan to investigate the Student t-distribution and its variants as it has interesting statistical properties. Moreover, we plan to combine deep learning methods with robust rigid registration methods in order to distinguish between rigid and non-rigid motion and to separate them. This research will combine robust probability distributions functions with deep learning and hence will lead to novel algorithms for robustly detecting landmarks and tracking them over time. Simultaneously, we will address the problem of assessing the quality of the landmarks without systematic recourse to annotated datasets.

- *Deep audio descriptors.* We will also investigate methods for extracting descriptors from audio signals. These descriptors must be free of noise and reverberation. While there are many noise filtering and dereverberation methods available, they are not necessarily well adapted to the tasks involved in live interaction between a robot and a group of people. In particular, they often treat the case of a static acoustic scene: both the sources and the microphones remain fixed. This represents a strong limitation and the existing methods must be extended to deal with dynamic acoustic scenes, e.g. [315]. We plan to develop deep audio descriptors that are robust against noise and reverberation. In particular we plan to address speech enhancement and speech dereverbereration in order to facilitate the tasks of speech-source localization and speech recognition. Moreover, we plan to develop a speaker recognition method that can operate in a complex acoustic environment. Recent work and recently released datasets provide a solid starting point for the task of in-the-wild speaker recognition.

IX.3.3 Learning robot actions

Whenever a robot acts in a populated space, it must perform *social actions*. Such robot social actions are typically associated with the need to perceive a person or a group of persons in an optimal way as well as to take appropriate decisions such as to safely move towards a selected group, to pop into a conversation or to answer a question. Therefore, one can distinguish between two types of robot social actions: (i) *physical actions* which correspond to synthesizing appropriate motions using the robot actuators (motors), possibly within a sensorimotor loop, so as to enhance perception and maintain a natural interaction and (ii) *spoken actions* which correspond to synthesizing appropriate speech utterances needed by a spoken dialog system. We will focus on the former, and integrate the latter via collaborations with research groups having with established expertise in speech technologies.

In this context, we face three problems. First, given the complexity of the environment and the inherent limitations of the robot's perception capabilities, e.g. limited camera field of view, cluttered spaces, complex acoustic conditions, etc., the robot will only have access to a partial representation of the environment, and up to a certain degree of accuracy. Second, for learning purposes, there is no easy way to annotate which are the best actions the robot must choose given a situation: supervised methods are therefore not an option. Finally, given that the robot moves within a populated environment, it is desirable to have the capability to enforce certain constrains, thus limiting the range of possible robot actions. There are mainly two methodologies for robot action. On the one hand we have sensor-based robot control techniques, such as model predictive control (MPC), that require a faithful representation of the transition function so as to compute the optimal action trajectory. On the other hand we have learning-based techniques, such as deep Q networks, that allow to learn the transition function together with the optimal policy function, but they cannot be coupled with hard-constraints. Our scenario is complex enough to require learning (part of) the transition function, and at the same time we would like to enforce constraints when controling the robot.

- *Constrained RL.* Naturally one may be tempted to combine MPC and DQN, but this is unfortunately not possible. Indeed, DQN cannot disentangle the policy π from the environment f, and MPC requires an explicit expression for f to solve the associated optimisation problem, their direct combination is not possible. We will investigate two directions. First, to devise methodologies able to efficiently learn the transition function f, to later on use it within the MPC framework. Second, to design learning methodologies that are combined with MPC, so that the actions taken within the learning process satisfy the required constraints. A few combinations of RL and MPC for robot navigation in human-free scenarios [316]–[318] as well as MPC variants driven by data [319], [320] have recently appeared in the literature. How to adapt this recent trend to dynamic complex environments such as a multi-party conversational situation is still to be investigated.

- *Meta RL.* An additional challenge, independent to the learning and control combination foreseen, is the data distribution gap between the simulations and the real-world. Meta-learning, or the ability to learn how to learn, can provide partial answers to this problem. Indeed, developing machine learning methods able to understand how the learning is achieved can be used to extend this learning to a new task and speed up the learning process on the new task. Recent developments proposed meta-learning strategies specifically conceived for reinforcement learning, leading to Meta-RL methods [321]. One promising trend in Meta-RL is to have a probabilistic formulation involving SSMs and VAEs, i.e. hence sharing the methodology based on dynamical variational autoencoders described before [322]. Very importantly, we are not aware of any studies able to combine Meta-RL with MPC to handle the constraints, and within a unified formulation. From a methodological perspective, this is an important challenge we face in the next few years.

IX.3.4 Statement of Scientific Ambition

In the near future, I would like to develop machine learning methods that enable social skills in robotic platforms. To do so, I will continue deriving models, training and inference algorithms, and associated implementations, at the crossroads of probabilistic models and deep neural networks. In addition, I will do my best in contributing fundamentally to the understanding of how to properly fuse auditory and visual information. Importantly, and this is a scientific risk that I would like to embrace, I will invest significant efforts in developing learning methods for social robot actions, thus endowing robotic platforms with action policies for social interaction. I hope that in a few years from now, I will be able to present to the scientific community a consistent sequence of impactful contributions in these directions.

Bibliography

[1] R. J. Sternberg, *Handbook of intelligence*. Cambridge University Press, 2000.

[2] M. Ganaie and H. Mudasir, "A study of social intelligence & academic achievement of college students of district srinagar, j&k, india," *Journal of American Science*, vol. 11, no. 3, pp. 23–27, 2015.

[3] P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

[4] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc., Series B (methodological)*, pp. 1–38, 1977.

[7] V. Smidl and A. Quinn, *The Variational Bayes Method in Signal Processing*. Springer, 2006.

[8] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Adv. Neural Information Processing Systems (NIPS)*, 2014, pp. 3581–3589.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[11] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[12] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: Adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2015.

[13] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman, "Audio-visual waypoints for navigation," *arXiv preprint arXiv:2008.09622*, 2020.

[14] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani, "Listen to look: Action recognition by previewing audio," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 457–10 467.

[15] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[16] I. D. Gebru, X. Alameda-Pineda, R. Horaud, and F. Forbes, "Audio-visual speaker localization via weighted clustering," in *IEEE Workshop on Machine Learning for Signal Processing*, 2014.

[17] I.-D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM algorithms for weighted-data clustering with application to audio-visual scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2402–2415, 2016.

[18] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.

[19] ——, "Robust mixture modelling using the t distribution," *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000.

[20] C. M. Bishop and M. Svensen, "Robust bayesian mixture modelling," *Neurocomputing*, vol. 64, pp. 235–252, 2005.

[21] C. Archambeau and M. Verleysen, "Robust Bayesian clustering," *Neural Networks*, vol. 20, no. 1, pp. 129–138, 2007.

[22] J. Sun, A. Kabán, and J. M. Garibaldi, "Robust mixture clustering using pearson type VII distribution," *Pattern Recognition Letters*, vol. 31, no. 16, pp. 2447–2454, 2010.

[23] J. L. Andrews and P. D. McNicholas, "Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions," *Statistics and Computing*, vol. 22, no. 5, pp. 1021–1029, 2012.

[24] F. Forbes and D. Wraith, "A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweight: Application to robust clustering," *Statistics and Computing*, vol. 24, no. 6, pp. 971–984, Nov. 2014.

[25] S. Lee and G. McLachlan, "Finite mixtures of multivariate skew t-distributions: Some recent and new results," *Statistics and Computing*, vol. 24, no. 2, pp. 181–202, 2014.

[26] S. Kotz and S. Nadarajah, *Multivariate t Distributions and their Applications*. Cambridge, 2004.

[27] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu, "Spectral clustering for multi-type relational data," in *Proceedings of the 23rd International Conference on Machine learning*, ACM, 2006, pp. 585–592.

[28] G. Tseng, "Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data," *Bioinformatics*, vol. 23, no. 17, pp. 2247–2255, 2007.

[29] M. Ackerman, S. Ben-David, S. Branzei, and D. Loker, "Weighted clustering," in *Proceedings of AAAI*, 2012.

[30] D. Feldman and L. Schulman, "Data reduction for weighted and outlier-resistant clustering," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2012, pp. 1343–1354.

[31] F. Forbes, S. Doyle, D. Garcia-Lorenzo, C. Barillot, and M. Dojat, "A weighted multi-sequence Markov model for brain lesion segmentation," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 9, Sardinia, Italy, 2010, pp. 225–232.

[32] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[33] C. Hennig, "Methods for merging gaussian mixture components," *Advances in Data Analysis and Classification*, vol. 4, no. 1, pp. 3–34, 2010.

[34] J. P. Baudry, E. A. Raftery, G. Celeux, K. Lo, and R. Gottardo, "Combining mixture components for clustering," *Journal of Computational and Graphical Statistics*, vol. 19, no. 2, 2010.

[35] V. Melnykov, "Merging mixture components for clustering through pairwise overlap," *Journal of Computational and Graphical Statistics*, 2014.

[36] C. E. Rasmussen, "The infinite gaussian mixture model.," in *NIPS*, vol. 12, 1999, pp. 554–560.

[37] D. Gorur and C. Rasmussen, "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 653–664, 2010.

[38] H. Z. Yerebakan, B. Rajwa, and M. Dundar, "The infinite mixture of infinite gaussian mixtures," in *Advances in Neural Information Processing Systems*, 2014, pp. 28–36.

[39] X. Wei and C. Li, "The infinite student t-mixture for robust modeling," *Signal Processing*, vol. 92, no. 1, pp. 224–234, 2012.

[40] G. Schwarz *et al.*, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[41] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for mixtures," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, 2001.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. CRC Press, 1984.

[44] W. Street, W. Wolberg, and O. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, International Society for Optics and Photonics, 1993, pp. 861–870.

[45] P. W. Frey and D. J. Slate, "Letter recognition using holland-style adaptive classifiers," *Machine Learning*, vol. 6, no. 2, pp. 161–182, 1991.

[46] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN: 0387310738.

[47] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.

[48] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel K-means: spectral clustering and normalized cuts," in *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining*, ACM, 2004, pp. 551–556.

[49] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[50] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ACM, 2002, pp. 515–524.

[51] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 224–227, 1979.

[52] A. Deleforge, R. Horaud, Y. Y. Schechner, and L. Girin, "Co-localization of audio sources in images using binaural features and locally-linear regression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 4, pp. 718–731, Apr. 2015.

[53] V. Ferrari, M.-J. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[54] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[55] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[56] T. Hueber, L. Girin, X. Alameda-Pineda, and G. Bailly, "Speaker-adaptive acoustic-articulatory inversion using cascaded gaussian mixture regression," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2246–2259, 2015.

[57] L. Girin, T. Hueber, and X. Alameda-Pineda, "Extending the cascaded gaussian mixture regression framework for cross-speaker acoustic-articulatory mapping," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 662–673, 2017.

[58] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, 1998.

[59] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007, ISSN: 1558-7916.

[60] Y. Tian, L. Sigal, H. Badino, F. de la Torre, and Y. Liu, "Latent gaussian mixture regression for human pose estimation," in *Asian Conf. Comp. Vision*, Queenstown, New Zealand, 2010, pp. 679–690.

[61] A. Chowriappa, R. Rodrigues, T. Kesavadas, V. Govindaraju, and A. Bisantz, "Generation of handwriting by active shape modeling and global local approximation (GLA) adaptation," in *Int. Conf. Frontiers in Handwriting Recognition*, Kolkata, India, 2010, pp. 206–211. DOI: 10.1109/ICFHR.2010.40.

[62] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden markov model and gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[63] X. Alameda-Pineda and R. Horaud, "Vision-guided robot hearing," *Int. J. Rob. Res.*, vol. 34, no. 4-5, pp. 437–456, 2015, ISSN: 0278-3649.

[64] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[65] H. Zen, Y. Nankaku, and K. Tokuda, "Continuous stochastic feature mapping based on trajectory HMMs," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 2, pp. 417–430, 2011.

[66] G. Ananthakrishnan and O. Engwall, "Mapping between acoustic and articulatory gestures," *Speech Communication*, vol. 53, no. 4, pp. 567–589, 2011.

[67] S. Dusan and L. Deng, "Vocal-tract length normalization for acoustic-to-articulatory mapping using neural networks," *Proc. Acoust. Soc. Am.*, vol. 106, no. 4, p. 2181, 1999.

[68] G. Sivaraman, V. Mitra, H. Nam, M. Tiede, and C. Espy-Wilson, "Vocal tract length normalization for speaker independent acoustic-to-articulatory speech inversion," *Proc. Interspeech*, pp. 455–459, 2016.

[69] T. Hueber, G. Bailly, P. Badin, and F. Elisei, "Speaker adaptation of an acoustic-articulatory inversion model using cascaded gaussian mixture regressions," in *Proc. Interspeech*, Lyon, France, 2013, pp. 2753–2757.

[70] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Trans. Speech, Audio Process.*, vol. 2, no. 2, pp. 291–298, 1994, ISSN: 1063-6676.

[71] M. J. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996.

[72] G. McLachlan and K. Thriyambakam, *The EM algorithm and extensions*. New-York, USA: John Wiley and sons, 1997.

[73] Z. Ghahramani and M. I. Jordan, "Learning from incomplete data," Cambridge, MA, USA, Tech. Rep., 1994.

[74] H. Uchida, D. Saito, N. Minematsu, and K. Hirose, "Statistical acoustic-to-articulatory mapping unified with speaker normalization based on voice conversion," in *Proc. Interspeech*, Dresden, Germany, 2015, pp. 588–592.

[75] L. Ménard, J.-L. Schwartz, L.-J. Boë, and J. Aubin, "Articulatory–acoustic relationships during vocal tract growth for french vowels: Analysis of real data and simulations with an articulatory model," *Journal of Phonetics*, vol. 35, no. 1, pp. 1–19, 2007.

[76] P. Badin and G. Fant, "Notes on vocal tract computation," *Quarterly Progress and Status Report, Dept for Speech, Music and Hearing, KTH, Stockholm*, pp. 53–108, 1984.

[77] A. A. Wrench, "A multi-channel/multi-speaker articulatory database for continuous speech recognition research.," *Phonus.*, vol. 5, pp. 1–13, 2000.

[78] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," PhD thesis, University of Edinburgh, 2002.

[79] K. B. Petersen, M. S. Pedersen, *et al.*, *The matrix cookbook*, 2012.

[80] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "Deepgum: Learning deep robust regression with a gaussian-uniform mixture model," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 202–217.

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[82] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[83] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *CVPR*, 2014.

[84] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.

[85] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *CVPR*, 2014.

[86] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *IJCV*, 2016.

[87] M. Demirkus, D. Precup, J. J. Clark, and T. Arbel, "Hierarchical temporal graphical model for head pose estimation and subsequent attribute classification in real-world videos," *CVIU*, pp. 128–145, 2015.

[88] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *CVPR*, 2013.

[89] P. Huber, *Robust Statistics*. Wiley, 2004.

[90] R. A. Maronna, D. R. Martin, and V. J. Yohai, *Robust statistics*. John Wiley & Sons, 2006.

[91] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & sons, 2005, vol. 589.

[92] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *IJCV*, vol. 19, no. 1, pp. 57–91, 1996.

[93] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," *IJCV*, vol. 6, no. 1, pp. 59–70, 1991.

[94] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, pp. 513–537, 1999.

[95] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, ser. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2003.

[96] G. Beliakov, A. V. Kelarev, and J. Yearwood, "Robust artificial neural networks and outlier detection. Technical report," *CoRR*, vol. abs/1110.0169, 2011.

[97] R. Neuneier and H. G. Zimmermann, "How to train neural networks," in *Neural Networks: Tricks of the Trade*, Springer Berlin Heidelberg, 1998, pp. 373–423.

[98] P. Coretto and C. Hennig, "Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust Gaussian clustering," *JASA*, vol. 111, pp. 1648–1659, 2016.

[99] A. Galimzianova, F. Pernus, B. Likar, and Z. Spiclin, "Robust estimation of unbalanced mixture models on samples with outliers," *TPAMI*, vol. 37, no. 11, pp. 2273–2285, 2015.

[100] N. Neykov, P. Filzmoser, R. Dimova, and P. Neytchev, "Robust fitting of mixtures using the trimmed likelihood estimator," *CSDA*, vol. 52, no. 1, pp. 299–308, 2007.

[101] A. J. Bekker and J. Goldberger, "Training deep neural-networks based on unreliable labels," in *ICASSP*, 2016, pp. 2682–2686.

[102] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *CVPR*, 2015.

[103] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and J. Li, "Learning from Noisy Labels with Distillation," *arXiv preprint arXiv:1703.02391*, 2017.

[104] S. Mukherjee and N. Robertson, "Deep Head Pose: Gaze-Direction Estimation in Multimodal Video," *TMM*, vol. 17, no. 11, pp. 2094–2107, 2015.

[105] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *CoRR*, vol. abs/1603.01249, 2016.

[106] S. Lathuilière, R. Juge, P. Mesejo, R. Muñoz-Salinas, and R. Horaud, "Deep Mixture of Linear Inverse Regressions Applied to Head-Pose Estimation," in *CVPR*, 2017.

[107] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *ICCV*, 2015.

[108] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[109] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[110] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, pp. 80–83, 6 1945.

[111] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016.

[112] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, "Fashion Landmark Detection in the Wild," in *ECCV*, 2016.

[113] P. J. Huber, "Robust estimation of a location parameter," *The annals of mathematical statistics*, pp. 73–101, 1964.

[114] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Cross-age reference coding for age-invariant face recognition and retrieval," in *ECCV*, 2014.

[115] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *TIP*, vol. 26, pp. 1428–1440, 2017.

[116] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *CVPR*, Jun. 2013, pp. 532–539.

[117] Y. Ban, X. Alameda-Pineda, L. Girin, and R. Horaud, "Variational bayesian inference for audio-visual tracking of multiple speakers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[118] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 2, pp. 601–616, 2007.

[119] T. Hospedales and S. Vijayakumar, "Structure inference for bayesian multisensory scene understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2140–2157, 2008.

[120] S. Naqvi, M. Yu, and J. Chambers, "A multimodal approach to blind source separation of moving sources," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 895–910, 2010.

[121] V. Kılıç, M. Barnard, W. Wang, and J. Kittler, "Audio assisted robust visual tracking with adaptive particle filtering," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 186–200, 2015.

[122] N. Schult, T. Reineking, T. Kluss, and C. Zetzsche, "Information-driven active audio-visual source localization," *PloS one*, vol. 10, no. 9, 2015.

[123] M. Barnard, W. Wang, A. Hilton, J. Kittler, *et al.*, "Mean-shift and sparse sampling-based SMC-PHD filtering for audio informed visual speaker tracking," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2417–2431, 2016.

[124] V. Kılıç, M. Barnard, W. Wang, A. Hilton, and J. Kittler, "Mean-shift and sparse sampling-based SMC-PHD filtering for audio informed visual speaker tracking," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2417–2431, 2016.

[125] S. Ba, X. Alameda-Pineda, A. Xompero, and R. Horaud, "An on-line variational Bayesian model for multi-person tracking from cluttered scenes," *Computer Vision and Image Understanding*, vol. 153, pp. 64–76, 2016.

[126] S. Bae and K. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[127] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 216–228, 2007.

[128] A. Lombard, Y. Zheng, H. Buchner, and W. Kellermann, "TDOA estimation for multiple sound sources in noisy and reverberant environments using broadband independent component analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1490–1503, 2011.

[129] X. Alameda-Pineda and R. Horaud, "A geometric approach to sound source localization from time-delay estimates," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1082–1095, 2014.

[130] Y. Dorfan and S. Gannot, "Tree-based recursive expectation-maximization algorithm for localization of acoustic sources," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1692–1703, 2015.

[131] X. Li, L. Girin, R. Horaud, and S. Gannot, "Estimation of the direct-path relative transfer function for supervised sound-source localization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2171–2186, 2016.

[132] X. Li, L. Girin, R. Horaud, S. Gannot, X. Li, L. Girin, R. Horaud, and S. Gannot, "Multiple-speaker localization based on direct-path features and likelihood maximization with spatial sparsity regularization," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 10, pp. 1997–2012, 2017.

[133] A. Deleforge, R. Horaud, Y. Y. Schechner, and L. Girin, "Co-localization of audio sources in images using binaural features and locally-linear regression," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 4, pp. 718–731, 2015.

[134] B. Gold, N. Morgan, and D. Ellis, *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.

[135] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez, "AV16.3: An audio-visual corpus for speaker localization and tracking," in *Machine Learning for Multimodal Interaction*, Springer, 2004, pp. 182–195.

[136] I. D. Gebru, S. Ba, X. Li, and R. Horaud, "Audio-visual speaker diarization based on spatiotemporal Bayesian fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1086–1099, 2018. DOI: 10.1109/TPAMI.2017.2648793.

[137] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, 2011.

[138] D. Vijayasenan and F. Valente, "DiarTk: an open source toolkit for research in multistream speaker diarization and its application to meeting recordings," in *INTERSPEECH*, Portland, OR, USA, 2012, pp. 2170–2173.

[139] N. Checka, K. Wilson, M. Siracusa, and T. Darrell, "Multiple person and speaker activity tracking with a particle filter," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004, pp. 881–884.

[140] Y. Liu, W. Wang, J. Chambers, V. Kilic, and A. Hilton, "Particle flow SMC-PHD filter for audio-visual multi-speaker tracking," in *International Conference on Latent Variable Analysis and Signal Separation*, 2017, pp. 344–353.

[141] Y. Liu, A. Hilton, J. Chambers, Y. Zhao, and W. Wang, "Non-zero diffusion particle flow SMC-PHD filter for audio-visual multi-speaker tracking," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4304–4308, Apr. 2018.

[142] X. Qian, A. Brutti, M. Omologo, and A. Cavallaro, "3d audio-visual speaker tracking with an adaptive particle filter," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, New-Orleans, Louisiana, 2017, pp. 2896–2900.

[143] X. Li, Y. Ban, L. Girin, X. Alameda-Pineda, and R. Horaud, "Online Localization and Tracking of Multiple Moving Speakers in Reverberant Environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 88–103, Mar. 2019.

[144] Y. Ban, X. Alameda-Pineda, C. Evers, and R. Horaud, "Tracking Multiple Audio Sources with the Von Mises Distribution and Variational EM," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 798–802, Jun. 2019.

[145] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[146] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[147] A. Noulas, G. Englebienne, and B. Krose, "Multimodal speaker diarization," *IEEE TPAMI*, vol. 34, no. 1, pp. 79–93, 2012.

[148] G. Lathoud and M. Magimai-Doss, "A sector-based, frequency-domain approach to detection and localization of multiple speakers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 3, 2005, pp. 265–268.

[149] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, USA, 2017, pp. 7291–7299.

[150] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, USA, 2017, pp. 1367–1376.

[151] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[152] V. P. Minotto, C. R. Jung, and B. Lee, "Multimodal multi-channel on-line speaker diarization using sensor fusion through SVM," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1694–1705, 2015.

[153] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe, "Learning cross-modal deep representations for robust pedestrian detection," 2017.

[154] J. Canny, "A computational approach to edge detection," *TPAMI*, no. 6, pp. 679–698, 1986.

[155] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *TPAMI*, vol. 26, no. 5, pp. 530–549, 2004.

[156] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *CVPR*, 2013.

[157] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *CVPR*, 2015.

[158] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *CVPR*, 2015.

[159] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015.

[160] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," *arXiv preprint arXiv:1511.07386*, 2015.

[161] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Convolutional oriented boundaries," in *ECCV*, 2016.

[162] X. Ren, "Multi-scale improves boundary detection in natural images," in *ECCV*, 2008.

[163] X. Chu, W. Ouyang, X. Wang, *et al.*, "Crf-cnn: Modeling structured information in human pose estimation," in *NIPS*, 2016.

[164] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," *CVPR*, 2017.

[165] V. Mnih, N. Heess, A. Graves, *et al.*, "Recurrent models of visual attention," in *NIPS*, 2014, pp. 2204–2212.

[166] T. Minka and J. Winn, "Gates," in *NIPS*, 2009.

[167] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *TPAMI*, vol. 33, no. 5, 2011.

[168] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[169] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," 2016.

[170] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," *arXiv preprint arXiv:1612.02103*, 2016.

[171] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[172] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.

[173] S. Yang and D. Ramanan, "Multi-scale recognition with dag-cnns," in *ICCV*, 2015.

[174] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *CVPR*, 2015.

[175] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015.

[176] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention.," in *ICML*, 2015.

[177] Y. Tang, "Gated boltzmann machine for recognition under occlusion," in *NIPS Workshop on Transfer Learning by Learning Rich Generative Models*, 2010.

[178] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[179] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[180] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, *et al.*, "Crafting gbd-net for object detection," *arXiv preprint arXiv:1610.02579*, 2016.

[181] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, 2015.

[182] J. Winn, "Causality with gates," in *AISTATS*, 2012.

[183] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[184] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[185] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *ICCV*, 2013.

[186] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *ECCV*, 2014.

[187] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014.

[188] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[189] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.

[190] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[191] Z. Ren and G. Shakhnarovich, "Image segmentation by cascaded region agglomeration," in *CVPR*, 2013.

[192] J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *TPAMI*, 2016.

[193] Q. Zhao, "Segmenting natural images with the least effort as humans," in *BMVC*, 2015.

[194] S. Hallman and C. C. Fowlkes, "Oriented edge forests for boundary detection," in *CVPR*, 2015.

[195] Z. Zhang, F. Xing, X. Shi, and L. Yang, "Semicontour: A semi-supervised learning approach for contour detection," in *CVPR*, 2016.

[196] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *TPAMI*, vol. 37, no. 8, pp. 1558–1570, 2015.

[197] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from rgb-d images," in *CVPR*, 2013.

[198] M. Sadeghi, S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "Audio-visual speech enhancement using conditional variational auto-encoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1788–1800, 2020.

[199] M. Sadeghi and X. Alameda-Pineda, "Mixture of inference networks for vae-based audio-visual speech enhancement," *Submitted to IEEE Transactions on Signal Processing*, 2020.

[200] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[201] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2007.

[202] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[203] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[204] W. DeLiang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[205] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[206] X. Li and R. Horaud, "Multichannel speech enhancement based on time-frequency masking using subband long short-term memory," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, pp. 1–5.

[207] K. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, 2008, pp. 4029–4032.

[208] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[209] F. Sedighin, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Multimodal soft nonnegative matrix co-factorization for convolutive source separation," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3179–3190, 2017.

[210] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[211] F. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proc. Int. Conf. Indep. Component Analysis and Signal Separation*, 2007, pp. 414–421.

[212] G. J. Mysore and P. Smaragdis, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[213] M. Sun, X. Zhang, and T. F. Zheng, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 93–104, 2016.

[214] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 716–720.

[215] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.

[216] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian multichannel speech enhancement with a deep speech prior," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 1233–1239.

[217] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.

[218] S. Leglaive, U. Şimşekli, A. Liutkus, L. Girin, and R. Horaud, "Speech enhancement with variational autoencoders and alpha-stable distributions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 541–545.

[219] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Computation*, vol. 31, no. 9, pp. 1–24, 2019.

[220] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

[221] W. Sumby and I. Pollack, "Visual contribution to speech intelligibility in noise," *The Journal of the Acoustical Society of America*, vol. 26, no. 2, pp. 212–215, 1954.

[222] N. Erber, "Auditory-visual perception of speech," *Journal of Speech and Hearing Disorders*, vol. 40, no. 4, pp. 481–492, 1975.

[223] A. MacLeod and Q. Summerfield, "Quantifying the contribution of vision to speech perception in noise," *British Journal of Audiology*, vol. 21, no. 2, pp. 131–141, 1987.

[224] L. Girin, G. Feng, and J.-L. Schwartz, "Noisy speech enhancement with filters estimated from the speaker's lips," in *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, Madrid, Spain, 1995, pp. 1559–1562.

[225] L. Girin, J.-L. Schwartz, and G. Feng, "Audio-visual enhancement of speech in noise," *The Journal of the Acoustical Society of America*, vol. 109, no. 6, pp. 3007–3020, 2001.

[226] J. W. Fisher III, T. Darrell, W. T. Freeman, and P. A. Viola, "Learning joint statistical models for audio-visual fusion and segregation," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2001, pp. 772–778.

[227] S. Deligne, G. Potamianos, and C. Neti, "Audio-visual speech enhancement with AVCDCN (audio-visual codebook dependent cepstral normalization)," in *Proc. IEEE International Workshop on Sensor Array and Multichannel Signal Processing*, 2002, pp. 68–71.

[228] R. Goecke, G. Potamianos, and C. Neti, "Noisy audio feature enhancement using audio-visual speech data," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, pp. 2025–2028.

[229] J. R. Hershey and M. Casey, "Audio-visual sound separation via hidden markov models," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 1173–1180.

[230] A. H. Abdelaziz, S. Zeiler, and D. Kolossa, "Twin-HMM-based audio-visual speech enhancement," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3726–3730.

[231] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3244–3248.

[232] A. Gabbay, A. Shamir, and S. Peleg, "Visual speech enhancement," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1170–1174.

[233] A. Gabbay, A. Ephart, T. Halperin, and S. Peleg, "Seeing through noise: Speaker separation and enhancement using visually-derived speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 3051–3055.

[234] J.-C. Hou, S.-S. Wang, Y.-H. Lai, Y. Tsao, H.-W. Chang, and H.-M. Wang, "Audio-visual speech enhancement using multimodal deep convolutional neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 117–128, 2018.

[235] M. Gogate, A. Adeel, R. Marxer, J. Barker, and A. Hussain, "DNN driven speaker independent audio-visual mask estimation for speech separation," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2723–2727.

[236] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 101–105.

[237] A.-H. Abdelaziz, "NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 3752–3756.

[238] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[239] R. Martin, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 845–856, 2005.

[240] J. Erkelens, R. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1741–1752, 2007.

[241] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[242] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal processing*, vol. 81, no. 11, pp. 2403–2418, 2001.

[243] B. Raj, R. Singh, and T. Virtanen, "Phoneme-dependent NMF for speech enhancement in monaural mixtures," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 1217–1220.

[244] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2013, pp. 436–440.

[245] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement.," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 3768–3772.

[246] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[247] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[248] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2015, pp. 91–99.

[249] L. Li, H. Kameoka, and S. Makino, "Fast MVAE: Joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 546–550.

[250] S. Inoue, H. Kameoka, L. Li, S. Seki, and S. Makino, "Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 96–100.

[251] I. Almajai and B. Milner, "Visually derived wiener filters for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1642–1651, 2010.

[252] G. C. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[253] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[254] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[255] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 6548–6552.

[256] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 3483–3491.

[257] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-vae: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations (ICLR)*, 2017.

[258] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005, ISBN: 0387212396.

[259] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[260] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag Berlin, Heidelberg, 2006.

[261] M. Sadeghi and X. Alameda-Pineda, "Robust unsupervised audio-visual speech enhancement using a mixture of variational autoencoders," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[262] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[263] M. Sadeghi, S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "Audio-visual speech enhancement using conditional variational auto-encoder," *arXiv preprint arxiv.org/abs/1908.02590*, 2019.

[264] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[265] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 749–752.

[266] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[267] M. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic phonetic continuous speech corpus," in *Linguistic data consortium*, 1993.

[268] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[269] G. Delorme, Y. Xu, S. Lathuilière, R. Horaud, and X. Alameda-Pineda, "Canu-reid: A conditional adversarial network for unsupervised person re-identification," in *International Conference on Pattern Recognition*, 2020.

[270] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *ICPR*, 2014.

[271] T. Matsukawa and E. Suzuki, "Person re-identification using cnn features learned from combination of attributes," in *ICPR*, 2017.

[272] N. Jiang, J. Liu, C. Sun, Y. Wang, Z. Zhou, and W. Wu, "Orientation-guided similarity learning for person re-identification," in *ICPR*, 2018.

[273] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV Workshops*, 2016.

[274] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *IEEE ICCV*, 2015.

[275] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *IEEE CVPR*, 2018.

[276] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2018.

[277] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, and T. S. Huang, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," in *IEEE ICCV*, 2019.

[278] Y. Ge, D. Chen, and H. Li, "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification," *ICLR*, 2020.

[279] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *IEEE CVPR*, 2017.

[280] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, 2016.

[281] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *NIPS*, 2016.

[282] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *IEEE CVPR*, 2018.

[283] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *IEEE ICCV*, 2017.

[284] Y. Lin, X. Dong, L. Zheng, Y. Yan, and Y. Yang, "A bottom-up clustering approach to unsupervised person re-identification," in *AAAI*, 2019.

[285] X. Zhang, J. Cao, C. Shen, and M. You, "Self-training with progressive augmentation for unsupervised cross-domain person re-identification," in *IEEE CVPR*, 2019.

[286] Z. Zhong, L. Zheng, S. Li, and Y. Yang, "Generalizing a person retrieval model hetero-and homogeneously," in *ECCV*, 2018.

[287] X. Chang, Y. Yang, T. Xiang, and T. M. Hospedales, "Disjoint label space transfer learning with common factorised space," in *AAAI*, 2019.

[288] L. Qi, L. Wang, J. Huo, L. Zhou, Y. Shi, and Y. Gao, "A novel unsupervised camera-aware domain adaptation framework for person re-identification," in *IEEE CVPR*, 2019.

[289] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, and X. Wang, "Unsupervised domain adaptive re-identification: Theory and practice," *Pattern Recognition*, 2020.

[290] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *IEEE CVPR*, 2019.

[291] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications*, E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano, Eds., 2009.

[292] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *IEEE CVPR*, Jun. 2019.

[293] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *IEEE CVPR*, 2018.

[294] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," *IEEE CVPR*, 2018.

[295] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *IEEE CVPR*, 2019, pp. 11 293–11 302.

[296] Y. Yao, Y. Zhang, X. Li, and Y. Ye, "Heterogeneous domain adaptation via soft transfer network," in *ACM MM*, 2019.

[297] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *ECCV*, 2018.

[298] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[299] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *ArXiv preprint*, 2017.

[300] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, 1996, pp. 226–231.

[301] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, 1991.

[302] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[303] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018.

[304] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *IEEE CVPR*, 2018.

[305] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.

[306] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[307] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[308] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *IEEE CVPR*, 2018, pp. 2275–2284.

[309] Y.-J. Li, F.-E. Yang, Y.-C. Liu, Y.-Y. Yeh, X. Du, and Y.-C. Frank Wang, "Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification," in *IEEE CVPR Workshops*, 2018, pp. 172–178.

[310] Y.-J. Li, C.-S. Lin, Y.-B. Lin, and Y.-C. F. Wang, "Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation," in *IEEE ICCV*, 2019.

[311] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NeurIPS*, 2018.

[312] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *IEEE CVPR*, 2018.

[313] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[314] S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "A recurrent variational autoencoder for speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 371–375.

[315] D. Kounades-Bastian, L. Girin, X. Alameda-Pineda, S. Gannot, and R. Horaud, "A variational EM algorithm for the separation of time-varying convolutive audio mixtures," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1408–1423, 2016.

[316] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7559–7566.

[317] K. Napat, M. I. Valls, D. Hoeller, and M. Hutter, "Practical reinforcement learning for mpc: Learning from sparse objectives in under an hour on a real robot," in *2nd Annual Conference on Learning for Dynamics and Control (L4DC 2020)*, 2020.

[318] D. Hoeller, F. Farshidian, and M. Hutter, "Deep value model predictive control," in *Conference on Robot Learning*, 2020, pp. 990–1004.

[319] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven mpc design," *IEEE control systems letters*, vol. 3, no. 3, pp. 577–582, 2019.

[320] J. Berberich, J. Köhler, M. A. Muller, and F. Allgower, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, 2020.

[321] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," in *Advances in Neural Information Processing Systems*, 2018, pp. 5302–5311.

[322] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *International conference on machine learning*, 2019, pp. 5331–5340.