Deep Reinforcement Learning for Audio-Visual Gaze Control

Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, Radu Horaud

Deep Reinforcement Learning for Audio-Visual Gaze Control

Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, and Radu Horaud

Abstract— **We address the problem of audio-visual gaze control in the specific context of human-robot interaction, namely how controlled robot motions are combined with visual and acoustic observations in order to direct the robot head towards targets of interest. The paper has the following contributions: (i) a novel audio-visual fusion framework that is well suited for controlling the gaze of a robotic head; (ii) a reinforcement learning (RL) formulation for the gaze control problem, using a reward function based on the available temporal sequence of camera and microphone observations; and (iii) several deep architectures that allow to experiment with early and late fusion of audio and visual data. We introduce a simulated environment that enables us to learn the proposed deep RL model without the need of spending hours of tedious interaction. By thoroughly experimenting on a publicly available dataset and on a real robot, we provide empirical evidence that our method achieves state-of-the-art performance.**

I. Introduction

In recent years there has been a growing interest in the development of robotic systems able to communicate with people, i.e. human-robot interaction (HRI). Unlike traditional robot perception systems that have primarily been used for robot localization and navigation, HRI implies that there are people in the loop, therefore the robot must take decisions in order to optimally interact with users. For example, a robot can recognize an user's gestures, intentions, or speech only if the robot faces that user, i.e. dyadic interaction. Moreover, robots are likely to be present in populated spaces, such as hospitals, museums, hotel lobbies, etc. Consequently, a robot should be able to interact with a group of people or be part of a team. In situations such as the ones cited, a robot teammate must constantly maintain the participants in its visual and acoustic fields of view such that it can easily receive instructions while respecting social etiquette.

In this paper, we address the problem of audio-visual gaze control, or more precisely, how a robot should combine controlled motions with acoustic and visual observations in order to direct its head towards groups of people. *Active perception* is necessary for making inferences from observations; it is equally needed for deciding to look at something or to speak with someone. The objective is to design a methodology that enables robots to learn gazing strategies from data; for example, to maximize the number of persons that are present in its visual field of view and, possibly, to favor people engaged in spoken communication.

Also, it is interesting to note that gaze control has been mainly addressed for dyadic interaction. In multi-party scenarios, focusing on only one person may lead to miss important information such as who looks at whom and who is the speaker and who are the listeners [1]. Hence there is a danger that the controller makes suboptimal decisions with respect to the task at hand. We address gaze control in the specific case of multi-party interaction.

Gaze control was already addressed within the framework of sensor-based robot servoing. For example, visual servoing consists of designing a control loop that aligns the observed position of an object with a targeted position [2]. This implies that the direct and inverse robot Jacobians are known. Alternatively, these Jacobians may be estimated via reinforcement learning [3]. Recently, the concept of sensor-based servoing was applied to the audio modality by directly linking observed acoustic features to robot control. However this approach makes the strong assumption that there is a single sound source that emits continuously. e.g. [4], [5]. Unfortunately this cannot be applied to speech uttered by several participants. Currently, sensor-based servoing methods that are able to combine visual and audio features, possibly associated with several persons, are not available. When several modalities and hence several types of sensors are available, it is difficult to optimally fuse the available sensory data and to implement an optimal controller, based on handcrafted rules that must consider all the situations that may occur.

In this paper, we propose a reinforcement learning approach [6] to the gaze control problem, e.g. Fig. 1. Reinforcement learning (RL) has several advantages over sensor-based servoing as it replaces a handcrafted control strategy with a trial-and-error learning model. Over time, the agent, e.g. the robot, refines its behavior via optimization of a reward-based function that may well be viewed as a feedback signal that indicates whether the robot actions are beneficial or not. The model can be trained both offline and online, which yields interesting adaptation capabilities. As it will be described in detail below, there is no need of an annotated training dataset as is often the case with machine learning techniques.

The paper has the following contributions. We built a novel audio-visual fusing framework that is well suited for controlling the gaze of a robotic head in a multi-party interaction scenario. We map the gaze control problem in the framework of RL and we propose a reward function based on the available temporal sequence of camera and microphone observations. We use deep RL to model the action-value function, and suggest several deep architectures based on

The authors are with INRIA Grenoble Rhône-Alpes and Univ. Grenoble Alpes. Work supported by ERC Adv. Grant #340113 *"Vision and Hearing in Action"*.

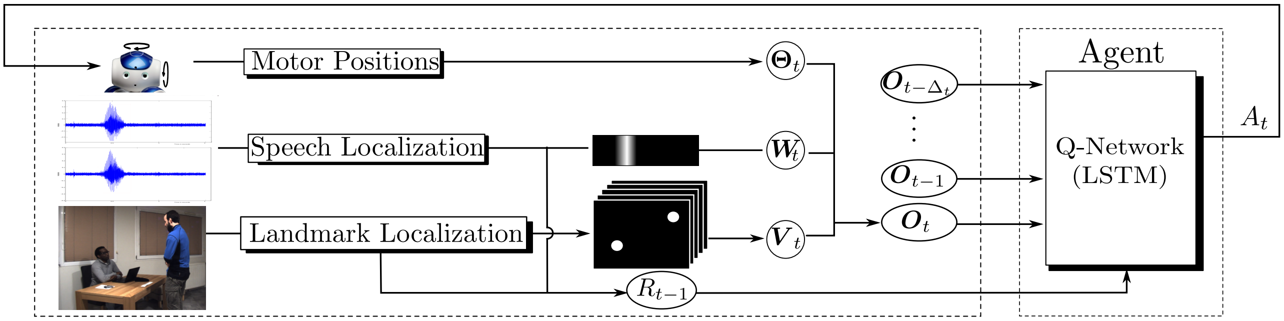


Fig. 1: Overview of the proposed deep RL method for controlling the gaze of a robot. At each time index t, audio and visual data are represented as features maps which, together with motor positions, form the set of observations $\boldsymbol{O}_t$. A motor action A_t (rotate left, right, up, down, or stay still) is selected based on past and present observations via maximization of current and future rewards. The rewards R are based on the number of visible persons as well as on the presence of speech sources in the camera field of view. We use a deep Q-network (DQN) model that can be learned both offline and online. Please refer to Section III and Section IV for the mathematical notations and detailed problem formulation.

LSTM (a recurrent neural network model) that allow us to experiment with early fusion and late fusion of audio and visual data. We introduce a simulated environment that enables us to learn the proposed deep RL model without the need of spending hours of tedious interaction. By experimenting on a publicly available dataset and on a real robot, we provide empirical evidence that our method achieves state-of-the-art performance.

The remainder of this article is organized as follows. Section II describes related work. Section III presents the proposed mathematical formulation and Section IV describes the deep reinforcement learning architectures. Section V briefly describes the simulated environment needed for offline training. Section VI reports experiments and results obtained with a publicly available dataset and with a Nao robot.

II. Related Work

There are several RL-based HRI methods relevant to our work. In [7] an RL algorithm is used for a robot to learn to play a game with a human partner. The algorithm uses vision and force/torque feedback to choose the motor commands. The uncertainty associated with human actions is modeled via a Gaussian process model, and Bayesian optimization selects an optimal action at each time step. In [8] RL is employed to adjust motion speed, timing, interaction distances, and gaze in the context of HRI. The reward is based on the amount of movement of the subject and the time spent gazing at the robot in one interaction. As external cameras are required, this cannot be easily applied in scenarios where the robot has to keep learning in a real environment. Moreover, the method is limited to the case of a single human participant. Another example of RL applied to HRI can be found in [9], where a human-provided reward is used to teach a robot. This idea of interactive RL is also exploited in [10] in the context of a table-cleaning robot. Visual and speech recognition are used to get advice from a parent-like trainer to enable the robot to learn a good policy efficiently. An extrinsic reward is used in [11] to learn how to point a camera towards the active speaker in a conversation. Audio information is used to determine where to point the camera, while the reward is provided using visual information: the active speaker raises a blue card that can be easily identified by the robot. The use of a multimodal deep Q-network (DQN) to learn human-like interactions is proposed in both [12] and [13]. The robot must choose an action to shake hands with a person. The reward is either negative, if the robot tries unsuccessfully to shake hands, positive, if the hand-shake is successful, or null otherwise. In practice, the reward is obtained from a sensor located in the hand of the robot and it takes fourteen training days to learn this skill successfully. To the best of our knowledge, the closest work to ours is [14] where an RL approach learns good policies to control the orientation of a mobile robot during social group conversations. The robot learns to turn its head towards the speaking person. However, their model is learned on simulated data that are restricted to a few predefined scenarios with static people and a predefined spatial organization of the group.

As already mentioned, gaze control has been addressed in the framework of sensor-based servoing. In [15] a method is proposed that uses audio-visual input to detect, track, and involve multiple persons into an interaction. In a multi-person scenario, [16] investigated the complementary nature of tracking and visual servoing that enables the system to track several persons and to visually control the gaze such as to keep a selected person in the camera field of view. Also, in [17], a system for gaze control of socially interactive robots in multiple-person scenarios is presented. This method requires external sensors to locate human participants. However, in opposition to all these works, we aim at learning the optimal behavior for gaze control, using the minimal supervision represented by a reward function, instead of adopting an arbitrary and handcrafted gaze control strategy.

III. Reinforcement Learning for Gaze Control

We consider a robot which should gaze towards a group of people. Hence, the robot must learn by itself a gaze control strategy via a trial-and-error procedure. The desired robot action is to rotate its head, on which are mounted a camera and two microphones, such as to maximize the number of persons visible in the camera field-of-view. Moreover, the robot should prefer to look at speaking people. The overall architecture of the proposed methodology is shown in Fig. 1. The terms *agent* and *robot* will be used indistinctly.

Random variables and their realizations are denoted with uppercase and lowercase letters, respectively. Vectors and matrices are in bold italic. At each time index t, the agent gathers motor $\boldsymbol{\Theta}_t$, visual $\boldsymbol{V}_t$, and audio $\boldsymbol{W}_t$ observations and performs an action $A_t \in \mathcal{A}$ from an action set according to a policy π, i.e. controlling the head motors such that the robot gazes in a selected direction. Once an action is performed, the agent receives a reward R_t, as explained in detail below.

Without loss of generality we consider the companion robot Nao whose head has two rotational degrees of freedom, pan and tilt. Motor observations correspond to pan and tilt angles, $\boldsymbol{\Theta}_t = (\Theta_t^1, \Theta_t^2)$. The values of these angles are relative to a reference head orientation, e.g. aligned with the robot body. This reference orientation together with the motor limits define the robot-centered *motor field-of-view*, or M-FOV.

We use the multiple person detector of [18] to estimate visual landmarks for each detected person, namely the nose, eyes, ears, neck, shoulders, elbows, wrists, hip, knees and ankles, or a total of $J = 18$ possible landmarks for each person. Based on the detection of these landmarks, one can determine the number of (totally or partially) observed persons, N_t, as well as the number of observed faces, F_t. Notice that in general the number of faces that are present in the image (i.e. detection of nose, eyes or ears) may be smaller than the number of detected persons. The landmark coordinates are described in image coordinates. Since the camera is mounted onto the robot head, the landmarks are described in a head-centered reference system. The visual landmarks are represented by J binary grids of size $K_v \times L_v$, namely $\boldsymbol{V}_t \in \{0,1\}^{K_v \times L_v \times J}$, where 1 (or zero) corresponds to the presence (or absence) of a landmark. Notice that this representation gathers all the detected landmarks associated with the N_t detected persons. K_v and L_v are the horizontal and vertical resolution of the visual grids.

Audio observations are provided by the multiple speech-source localization method described in [19]. Audio observations are also represented with a binary grid of size $K_a \times L_a$, namely $\boldsymbol{W}_t \in \{0,1\}^{K_a \times L_a}$. A grid cell is set to 1 if a speech source is detected at that grid location and 0 otherwise. Similarly, K_a and L_a represent the resolution of the audio grid. The audio grid is robot-centered and hence it remains fixed whenever the robot turns its head. Moreover, the audio grid spans an *acoustic field-of-view*, or A-FOV, which is much wider than the *visual field-of-view*, or V-FOV, associated with the camera mounted onto the head. The motor observations allow to estimate the relative alignment between the audio and visual grids and to determine whether a speech source lies within the visual field-of-view or not. This is represented by the binary variable $\Sigma_t \in \{0,1\}$, such that $\Sigma_t = 1$ if a speech source lies in the visual field-of-view and $\Sigma_t = 0$ if none of the speech sources lies inside the visual field-of-view.

Let $\boldsymbol{O}_t = \{\boldsymbol{\Theta}_t, \boldsymbol{V}_t, \boldsymbol{W}_t\}$ and let $\boldsymbol{S}_t = \{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_t\}$ denote the state variable. Let the set of actions be defined by $\mathcal{A} = \{\varnothing, \leftarrow, \uparrow, \rightarrow, \downarrow\}$, namely either remain in the same position or turn the head by a fixed angle in one of the four cardinal directions. We propose to define the reward R_t as follows:

$$R_t = F_{t+1} + \alpha \Sigma_{t+1}, \tag{1}$$

where $\alpha \geq 0$ is an adjustment parameter. High α values return high rewards when speech sources lie within the camera field-of-view. We consider two types of rewards which are referred to in Section VI as *Face_reward* ($\alpha = 0$) and *Speaker_reward* ($\alpha > 0$). Notice that the number of observed faces F_t is independent of each person's speaking status. Upon the application at hand, the value of α allows one to weight the importance given to speaking persons.

In RL, the model parameters are learned on sequences of states, actions and rewards, called episodes. At each time index t, an optimal action A_t should be chosen in order to maximize future rewards, $R_t, R_{t+1}, \ldots, R_T$. We make the standard assumption that future rewards are discounted by a factor γ that defines the importance of short-term rewards as opposed to longer term ones. We define the discounted future return $\bar{R}_t$ as the discounted sum of future rewards, $\bar{R}_t = \sum_{\tau=0}^{T-t} \gamma^\tau R_{\tau+t}$. If $\gamma = 0$, $\bar{R}_t = R_t$ and, consequently, we aim at maximizing only the immediate reward whereas when $\gamma \approx 1$, we favor policies that leads to better rewards in the long term. Considering a fixed value of γ, we now aim at maximizing $\bar{R}_t$ at each time index t. In other words, the goal is to learn a policy, $\pi(a_t, \boldsymbol{s}_t) = P(A_t = a_t | \boldsymbol{S}_t = \boldsymbol{s}_t)$ with $(a_t, \boldsymbol{s}_t) \in \mathcal{A} \times \mathcal{S}$, such that if the agent chooses its actions according to the policy π, the expected $\bar{R}_t$ should be maximized. The Q-function (or the action-value function) is defined as the expected future return from state $\boldsymbol{S}_t$, taking action A_t and then following any given policy π:

$$Q_\pi(\boldsymbol{s}_t, a_t) = \mathbb{E}_\pi[\bar{R}_t | \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t]. \tag{2}$$

Learning the best policy corresponds to the following optimization problem $Q^*(\boldsymbol{s}_t, a_t) = \max_\pi [Q_\pi(\boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t)]$. In practice, the evaluation of the Q-function from (2) is intractable. Following [20], we propose to use a deep Q-network (DQN) $Q(\boldsymbol{s}, a, \boldsymbol{\omega})$, parametrized by $\boldsymbol{\omega}$ yielding the approximation $Q(\boldsymbol{s}, a, \boldsymbol{\omega}) \approx Q^*(\boldsymbol{s}, a)$. We minimize the following loss:

$$\mathcal{L}(\boldsymbol{\omega}^{(i)}) = \mathbb{E}_{\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{R}_t, \boldsymbol{S}_{t+1}} \Big[(Y^{(i-1)} - Q(\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{\omega}^{(i)}))^2 \Big], \tag{3}$$

where the superscript ${}^{(i)}$ denotes the iteration index associated with the optimization procedure and with $Y^{(i-1)} = R_t + \gamma \max_{a}(Q(\boldsymbol{S}_{t+1}, a, \boldsymbol{\omega}^{(i-1)}))$. In order to compute (3), we sample quadruplets $(\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{R}_t, \boldsymbol{S}_{t+1})$ following the policy implied by:

$$a_t = \operatorname*{argmax}_{a \in \mathcal{A}} Q(\boldsymbol{s}_t, a, \omega^{(i-1)}) \tag{4}$$

However, instead of only sampling according to (4), random actions a_t are taken in ϵ percents of the time indices in order to explore new policies. This approach is known as the ϵ-greedy policy. $\mathcal{L}$ is minimized over $\boldsymbol{\omega}^{(i)}$ by stochastic gradient descent. Refer to [20] for more technical details about the training algorithm.

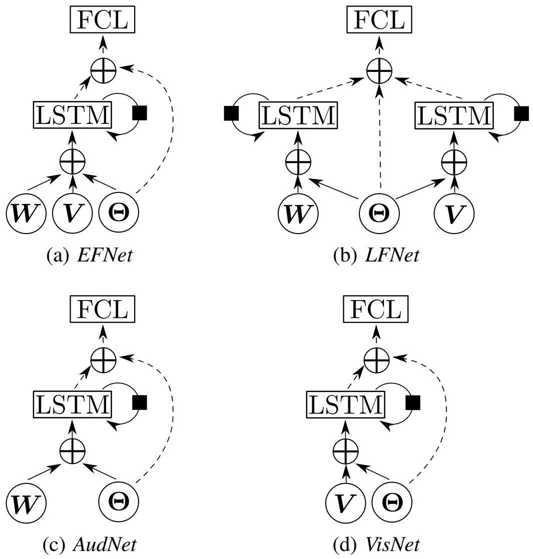
IV. Proposed DQN Architectures

We propose to model the Q-function with a long short-term memory (LSTM) [21] recurrent neural network that takes as input $\boldsymbol{S}_t^{\Delta t} = \{\boldsymbol{O}_{t-\Delta t}, \ldots, \boldsymbol{O}_t\}$ and that outputs a vector of size $\#\mathcal{A}$ that corresponds to each $Q(\boldsymbol{s}_t^{\Delta t}, a_t, \boldsymbol{\omega})$ with $a_t \in \mathcal{A}$, i.e. Section III. We argue that LSTM is well-suited for our task as it is capable of learning temporal dependencies better than other recurrent neural networks and than hidden Markov models. In practice, when a person is not detected anymore, the network should be able to use previous detections (back in time) in order to predict the direction towards which the robot should be gazing. The J grids of $\boldsymbol{V}_t$ are flattened before the LSTM layers. Batch normalization is applied to the output of the LSTM in order to accelerate training [22]. Following [20], the output layer is a fully-connected layer (FCL) with linear activations.

Four different network architectures were tested. They are described below and evaluated in Section VI. In order to evaluate how the visual and audio streams of information should be fused, we propose to compare two strategies: *early fusion* and *late fusion*. In early fusion, *EFNet*, the unimodal features are combined into a single representation before modeling time dependencies, i.e. Fig. 2a. In late fusion, *LFNet*, visual and audio features are processed separately before they are fused, i.e. Fig. 2b. In order to measure the impact of each modality, we propose two more network architectures that use either visual-only, *VisNet*, or audio-only, *AudNet*, input, e.g. Fig. 2c, where we used the compact graphical representation proposed in [23].

V. Simulated Environment for Training

Training a DQN model from scratch may require long periods until convergence, e.g. of the order of 150000 time steps in our case. Moreover, using a robot for training may not be convenient for two reasons. First, each robotic action takes an irreducible time. Second, in the case of HRI, participants would need to be actually present in front of a robot for tens of hours and to mimic realistic behaviors. Therefore, we propose to perform training using a simulated environment. DQN is learned using a simulated robot and people that move and speak. Then the Q-function thus learned is used to initialize the DQN associated with real gaze control in the presence of people. Importantly, the network learned from this simulated environment can be successfully used by the robot without the need of fine-tuning with real data. In this



(a) *EFNet*

(b) *LFNet*

(c) *AudNet*

(d) *VisNet*

Fig. 2: Proposed architectures to model the Q-function. Dashed lines indicate connections only used in the last time step. Black squares represent a delay of a single time step. Circled crosses represent the concatenation of inputs. FCL outputs a Q-value for each action.

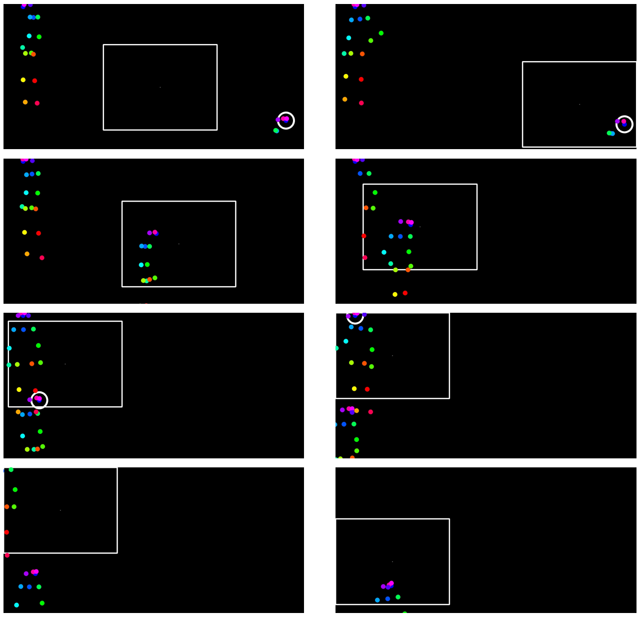


Fig. 3: Example of a simulated sequence used for offline training. The field of view is shown with a white rectangle. Visual landmarks associated with two persons are shown as colored dots. The white circles correspond to simulated speech sources that may correspond to a person.

simulated environment, we do not need to generate realistic images and sounds, instead we directly generate observations and rewards as needed by DQN learning.

Using the formulation introduced in Section III we defined motor, acoustic, and visual fields of view that correspond to the robot characteristics, e.g. Fig. 3. Without loss of generality we assumed that the motor field of view is the same as the acoustic field of view.

We simulated people that can freely move in a space that is larger than the motor/acoustic field of view. This allows us to consider people that randomly enter and quit this field of view. To simulate realistic human movements, we applied the person detector of [18] to the AVDIAR dataset [24] in order to collect a large number person poses and their associated landmarks. Realistic human trajectories were obtained using a smoother. Then the landmarks and their trajectories were mapped onto our simulated environment such that people move at different speeds, suddenly change their trajectories, come in and out the motor field of view, etc. Speech sources were simulated as follows. Three situations were randomly selected: one speaking person, two speaking persons, and no speaking person. A Markovian model was used to enforce temporal continuity of the speaking status. In addition, we also simulated speech sources that do not correspond to a person location.

VI. Experiments

A. Evaluation with Recorded Data

The evaluation of HRI systems is not an easy task. In order to fairly compare different models, we need to train and test each model on the exact same data. In the context of RL and HRI, this is problematic because the data, i.e. what the robot actually sees and hears, depends on the action taken by the robot. Thus, we propose to first evaluate our model with the AVDIAR dataset [24]. This dataset was recorded with four microphones and one high-resolution camera (1920×1080 pixels). These images, due to their wide field of view, are suitable to simulate the motor field of view of the robot. In practical terms, only a small box of the full image simulates the robot's camera field of view.

However, it is important to highlight that transferring the model learned using AVDIAR to Nao is problematic. First, faces are almost always located at the same position (around the image center). Second, all videos are recorded indoors using only two different rooms, and participants are not moving too much. Finally, the audio setting is unrealistic for a robotics scenario, e.g. absence of motor noise. Therefore, the main reason for using the AVDIAR dataset is to compare our method with other methods.

B. Live Experiments with Nao

In order to carry out an online evaluation of our method, we performed experiments with a Nao robot. Nao has a 640×480 pixels cameras and four microphones. This robot is particularly well suited for HRI applications because of its design, hardware specifications and affordable cost. Nao's commercially available software can detect people, locate sounds, understand some spoken words, synthesize speech and engage itself in simple and goal-directed dialogs. Our gaze control system is implemented on top of the NAOLab middleware [25] that synchronizes proprioceptive data (motor readings) and sensor information (image sequences and acoustic signals). The reason why we use a middleware is threefold. First, the implementation is platform-independent and, thus, easily portable. Platform-independence is crucial since we employ a transfer learning approach to transfer the model parameters, obtained with the proposed simulated environment, to the Nao software/hardware platform. Second, the use of external computational resources is transparent. This is also a crucial matter in our case, since visual processing is implemented on a GPU which is not available onboard of the robot. Third, the use of middleware makes prototyping much faster. For all these reasons, we employ the remote and modular layer-based middleware architecture named NAOLab. NAOLab consists of four layers: drivers, shared memory, synchronization engine and application programming interface (API). Each layer is divided into three modules devoted to vision, audio and proprioception, respectively. The last layer of NAOLab provides a general programming interface in C++ to handle the sensory data and to manage its actuators. NAOLab provides, at each time step, an image and the direction of the detected sound sources using [19], [26].

It is important to highlight that we pre-train the proposed model using the simulated environment before running live experiments on Nao. This environment is flexible and allows us to be closer to the actual conditions that Nao would face in practice (field of view range, uniform location of the people, etc.). For instance, in AVDIAR, heads are almost always at the same height. As a consequence, the learned model would not be general enough to perform well in real scenarios.

C. Implementation Details

We managed to obtain the full-body pose using [18] in less than 100 ms by carefully selecting the resolution used to perform the detection. Considering that NAOLab gathers images at 10 FPS, this pose estimator can be considered as fast enough for our scenario. Moreover, [18] follows a bottom-up approach, which allows us to speed-up landmark detection by skipping the costly association step.

The parameters of our model are based on a preliminary experimentation. We set $\Delta_T = 4$ in all scenarios, such that each decision is based on the last 5 observations. The output size of LSTM is set to 30 (since a larger size does not provide an improvement in performance), and the output size of the FCL is set to 5 (one per action). We use a discount factor (γ) of 0.90. Concerning the training phases, we employed the Adam optimizer [27] and a batch size of 128. In order to help the model to explore the policy space, we use an ϵ-greedy algorithm: while training, a random action is chosen in $\epsilon\%$ of the cases; we decrease linearly the ϵ value from $\epsilon = 90\%$ to $\epsilon = 10\%$ after 120000 iterations. Concerning the observations, we employ visual and audio grids of sizes 7×5 for the AVDIAR environment. However, due to the acoustic properties of the robot, it is very difficult to estimate the elevation of a speech source. Then, on the simulated environment and the Nao, the audio grids are 7×1. The models were trained in approximately 45 minutes on both AVDIAR and the simulated environment. It is interesting to notice that we obtain this training time without using GPUs. A GPU is only needed for person detection and estimation of visual landmarks (in our case, a Nvidia GTX 1070 GPU).

We also provide details specifically related to the Nao implementation. The delay between two successive observations is ∼0.3 seconds. The head has a motor field of view 180 degrees. The head motion parameters are chosen such that a single action corresponds to 0.15 radians (∼9°) and 0.10 radians (∼6°) for horizontal and vertical motions, respectively. Concerning the AVDIAR dataset, we employ 16 videos for training. The amount of training data is doubled by flipping the video and the speech grids. In order to save computation time, the original videos are down-sampled to 1024×640 pixels. The size of the camera field of view where faces can be detected is set to 300×200 pixels using motion steps of 36 pixels each. These dimensions approximately correspond the coverage angle and motion of Nao. At the beginning of each episode, the position of the camera field of view is selected such that it contains no face. We noticed that this initialization procedure favors the exploration abilities of the agent. To avoid a bias due to the initialization procedure, we used the same seed for all our experiments and iterated three times over the 10 test videos (20 when counting the flipped sequences). An action is taken every 5 frames (0.2 seconds). In the simulated environment, the size of field in which the people can move is set to $\xi = 1.4$. In the case of Nao, the audio observations are provided by the multiple speech-source localization method described in [19].

D. Results and Discussion

In all our experiments, we run five times each model and display the mean of five runs to lower the impact of the stochastic training procedure. On AVDIAR, the results on both training and test sets are reported in the tables. As described previously, the simulated environment is randomly generated in real time, so there is no need for a separated test set. Consequently, the mean reward over the last 10000 time steps is reported as test score.

In Table I, we compare the final reward obtained while training on the AVDIAR dataset and on our simulated environment with the two proposed rewards (*Face_reward* and *Speaker_reward*). Four different networks are tested: *EFNet*, *LFNet*, *VisNet*, and *AudNet*. The best results are generally provided by the late and early fusion strategies (*LFNet* and *EFNet*), showing that our model is able to effectively exploit the complementarity of both modalities. We observe that the rewards we obtain on AVDIAR are higher than those obtained on the simulated environment. We suggest two possible reasons. First, the simulated environment has been specifically designed to enforce exploration and tracking abilities. Consequently, it poses a more difficult problem to solve. Second, the number of people in AVDIAR is higher (about 4 in average), thus finding a first person to track would be easier. We notice that, on the AVDIAR dataset using the *Face_reward*, we obtain a mean reward greater than 1, meaning that, on average, our model can see more than one face per frame. We also observe that *AudNet* is the worst performing approach. However, it performs quite well on *AVDIAR* compared to the simulated environment. This behavior can be explained by the fact that, on *AVDIAR*, the speech source detector returns a 2D heatmap whereas only the yaw angle is used in the simulated environment. As conclusion, we select *LFNet* to perform experiments on Nao.

TABLE I: Comparison of the reward obtained with different architectures. The best results obtained are displayed in bold.

	AVDIAR		Simulated	
	Face	*Speaker*	*Face*	*Speaker*
AudNet	1.47 ± 0.04	1.82 ± 0.03	0.21 ± 0.01	0.33 ± 0.01
VisNet	$\mathbf{1.85 \pm 0.02}$	2.23 ± 0.03	0.37 ± 0.04	0.45 ± 0.06
EFNet	1.81 ± 0.04	2.22 ± 0.03	0.41 ± 0.03	$\mathbf{0.53 \pm 0.03}$
LFNet	1.83 ± 0.02	$\mathbf{2.29 \pm 0.02}$	$\mathbf{0.42 \pm 0.01}$	0.52 ± 0.03

Concerning the experiments performed on Nao, Figure 5 shows an example of a two-person scenario using the *LFNet* architecture. We managed to transfer the exploration and tracking abilities learned using the simulated environment. In our experiments, we see that our model behaves well independently of the number of participants, and the main failure cases are related to quick movements of the participants.

We now perform a comparative evaluation with respect to the state of the art. To the best of our knowledge, there is no existing work that tackles the problem of finding an optimal head motion policy in the HRI context. Only Bennewitz et al. [15] propose a heuristic that uses an audio-visual input to detect, track and involve multiple persons into interaction. We compare our learned policy with their proposed algorithm. On the simulated environment, as the speech source detector does not provide vertical information (see section VI-C), in the case where no person has been observed so far but a sound is detected, we randomly move along the vertical axis corresponding to the horizontal speech source position. In their experiments, Ban et al. [16] propose two strategies to evaluate their visual head control method. A first strategy consists in following a person and orienting the robot head in order to align the person's face with the image center. A second strategy consists in randomly jumping every 3 seconds between persons. Obviously, the second strategy was designed as a toy experiment and does not correspond to a natural behavior. Therefore, we compare our RL approach with their first strategy. Unfortunately, the case where nobody

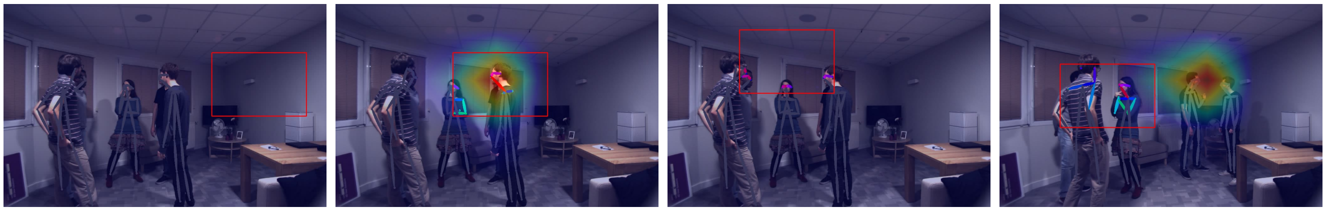


Fig. 4: Example of a sequence from the AVDIAR dataset. The speech direction binary grid is superimposed on the image, and the visible landmarks are displayed using a colored skeleton. The camera field of view (in red) is randomly initialized (far left). The agent controls its gaze, explores, and finds a person where it detects a speech source (left). The agent manages to get all the persons in the field of view (right), and it gazes at three persons when the group split (far right).

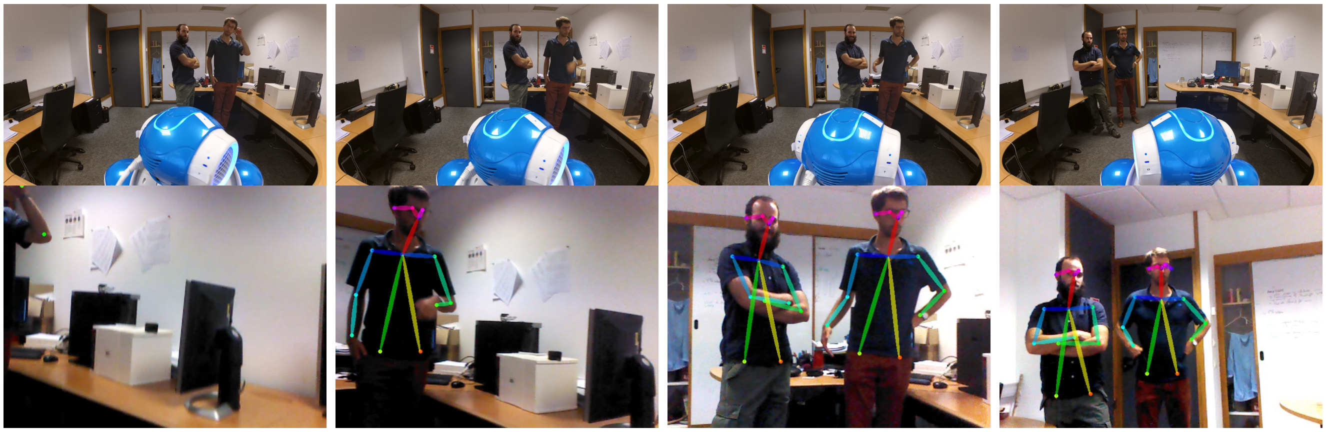


Fig. 5: Example of a live sequence with two persons. First row shows an overview of the scene, including the participants and the robot. Second row shows the images gathered with the camera mounted onto the robot head. The robot head is first initialized in a position where no face is visible (first column), and the model uses the available landmarks (elbow and wrist) to find the person onto the right (second column). The robot detects the second person by looking around while keeping the first person in its field of view (third column), and gazes the two people walking together (fourth column).

is in the field of view is not considered in [16]. To be able to compare their method in the more general scenario we tackle, we propose the following handcrafted policy in the case no face is detected in the field of view:

- *Rand*: A random action is chosen.
- *Center*: Go towards the center of the *acoustic field-of-view*.
- *Body*: Go up (↑) if a limb is detected, to find the corresponding head. Otherwise, *Rand* is followed.
- *Speech*: Go towards the position of the last detected speech source.

Importantly, in our model the head motion speed is limited, since the robot can only select unitary actions. When implementing other methods, one could argue that this speed limitation is inherent to our approach and that other methods may not suffer from it. However, it is not realistic to consider that the head can move between two opposite locations of the auditory field in two consecutive frames with an infinite speed. Therefore, we report two scores in our comparison. The first one is obtained using the same speed value than the one used in our model (referred to as *equal*). The second score is obtained by making the unrealistic assumption that the head motion speed is infinite (referred to as *infinite*). This second evaluation protocol is, therefore, biased in favor of handcrafted methods. The results obtained are reported in Table II.

First, we observe that no handcrafted policy can compete with our RL approach when considering models with equal head motion speeds. On both environments, *LFNet* largely outperforms all handcrafted policies. This clearly justifies the need of policy learning and the use of RL for the audio-visual gaze control. Concerning [16], *Center* obtains the best result among the [16]'s variances on AVDIAR and the worst on Simulated according to the *Face_reward* metric. It can be explained by the fact that, as mentioned in section VI-A, most people are located around the image center and, therefore, this dummy strategy works better than more sophisticated ones. A similar behavior can be observed with the *Speaker_reward* metric. We observe that, in both environments, using speech source localization when no face is detected improves the performance with respect to *Rand*. Concerning [15], it obtains the second best performance on AVDIAR with *Speaker_reward*. On the simulated environment, they equal the score obtained by our proposal when making the unrealistic assumption of infinite head motion speed. In that case, their performance is marginally

TABLE II: Comparison of the rewards obtained with different handcrafted policies. The performances of competitor methods are reported considering the two speed assumptions (*equal/infinite*) described in the text.

	AVDIAR		Simulated	
	Face_reward	*Speaker_reward*	*Face_reward*	*Speaker_reward*
Ban et al.[16]+*Rand*	1.19/1.21	1.45/1.59	0.25/0.26	0.40/0.37
Ban et al.[16]+*Center*	1.62/1.68	1.95/2.01	0.14/0.11	0.28/0.29
Ban et al.[16]+*Body*	1.23/1.20	1.40/1.52	0.27/0.26	0.39/0.37
Ban et al.[16]+*Speech*	1.54/1.63	1.84/2.06	0.32/0.39	0.43/0.48
Bennewitz et al.[15]	1.56/1.55	2.07/2.05	0.30/ **0.42**	0.35/0.50
LFNet	$\mathbf{1.83 \pm 0.02}$	$\mathbf{2.29 \pm 0.02}$	$\mathbf{0.42 \pm 0.01}$	$\mathbf{0.52 \pm 0.03}$

inferior to our proposal according to the *Speaker_reward*. When considering equal speed limit, our RL approach significantly outperforms their handcrafted approach (26% and 48% higher according to *Face_reward* and *Speaker_reward* respectively).

All these results highlight the major importance of audio-visual fusion in the context of gaze control for HRI, and that RL is an effective tool to tackle this task. The high variances on AVDIAR are coming from the impact of the random initial head orientation. On the contrary, our method has a low variance as it is able to adapt to any initialization. This illustrates the importance of combining tracking ability with an exploration strategy when no or only a single face is detected.

VII. Conclusions

In this paper, we presented a deep reinforcement learning approach to solve the gaze control problem in the specific context of human-robot interaction. In particular, our agent is able to autonomously learn how to find people in the environment by maximizing the number of people present in its field of view, while favoring people that speak. We built a simulated environment for offline training in order to avoid live training which implies hours of tedious interaction between a group of people and a robot. Neither external sensors nor human intervention are necessary to provide a reward. Several architectures and rewards are compared on three different environments: two offline (a recorded dataset and a simulated one) and one online (live experiments using the Nao robot). Our results suggest that fusion of audio and visual information yields state-of-the-art performance, that reinforcement learning outperforms handcrafted strategies, and that pre-training using a simulated environment is beneficial.

References

[1] B. Massé, S. Ba, and R. Horaud, "Tracking gaze and visual focus of attention of people involved in social interaction," *IEEE TPAMI*, 2017.

[2] A. Cretual and F. Chaumette, "Application of motion-based visual servoing to target tracking," *IJRR*, 2001.

[3] C. Gaskett, L. Fletcher, A. Zelinsky, *et al.*, "Reinforcement learning for visual servoing of a mobile robot," in *Australian Conference on Robotics and Automation*, 2000.

[4] G. Bustamante, P. Danés, T. Forgue, and A. Podlubne, "Towards information-based feedback control for binaural active localization," in *IEEE ICASSP*, 2016.

[5] A. Magassouba, N. Bertin, and F. Chaumette, "Aural servo: sensor-based control from robot audition," *IEEE TRO*, 2018.

[6] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. MIT Press, 1998.

[7] A. Ghadirzadeh, J. Bütepage, A. Maki, D. Kragic, and M. Björkman, "A sensorimotor reinforcement learning framework for physical Human-Robot Interaction," in *IEEE/RSJ IROS*, 2016, pp. 2682–2688.

[8] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning," *JRSJ*, 2006.

[9] A. L. Thomaz, G. Hoffman, and C. Breazeal, "Reinforcement learning with human teachers: Understanding how people want to teach robots," in *IEEE RO-MAN*, 2006, pp. 352–357.

[10] F. Cruz, G. I. Parisi, J. Twiefel, and S. Wermter, "Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario," in *IEEE/RSJ IROS*, 2016, pp. 759–766.

[11] M. Rothbucher, C. Denk, and K. Diepold, "Robotic gaze control using reinforcement learning," in *IEEE HAVE*, 2012.

[12] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *IEEE Humanoids*, 2016, pp. 745–751.

[13] ——, "Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network," in *IEEE ICRA*, 2017.

[14] M. Vázquez, A. Steinfeld, and S. E. Hudson, "Maintaining awareness of the focus of attention of a conversation: A robot-centric reinforcement learning approach," in *IEEE RO-MAN*, 2016.

[15] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and S. Behnke, "Towards a humanoid museum guide robot that interacts with multiple persons," in *IEEE-RAS*, 2005, pp. 418–423.

[16] Y. Ban, X. Alameda-Pineda, F. Badeig, S. Ba, and R. Horaud, "Tracking a varying number of people with a visually-controlled robotic head," in *IEEE/RSJ IROS*, 2017.

[17] S.-S. Yun, "A gaze control of socially interactive robots in multiple-person interaction," *Robotica*, vol. 35, no. 11, pp. 2122–2138, 2017.

[18] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *IEEE CVPR*, 2017.

[19] 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 TASLP*, 2017.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[24] I. Gebru, S. Ba, X. Li, and R. Horaud, "Audio-visual speaker diarization based on spatiotemporal bayesian fusion," *IEEE TPAMI*, 2017.

[25] F. Badeig, Q. Pelorson, S. Arias, V. Drouard, I. Gebru, X. Li, G. Evangelidis, and R. Horaud, "A distributed architecture for interacting with nao," in *ACM ICMI*, 2015, pp. 385–386.

[26] X. Li, L. Girin, F. Badeig, and R. Horaud, "Reverberant sound localization with a robot head based on direct-path relative transfer function," in *IEEE/RSJ IROS*, 2016.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.