Pseudo-healthy image reconstruction with variational autoencoders for anomaly detection: A benchmark on 3D brain FDG PET

Ravi Hassanaly, Maëlys Solal, Olivier Colliot, Ninon Burgos

Pseudo-healthy image reconstruction with variational autoencoders for anomaly detection: A benchmark on 3D brain FDG PET

Ravi Hassanaly[a], Maëlys Solal[a], Olivier Colliot[a], Ninon Burgos[a], for the Alzheimer's Disease Neuroimaging Initiative[1]

[a]*Sorbonne Université, Institut du Cerveau - Paris Brain Institute - ICM, CNRS, Inria, Inserm, AP-HP, Hôpital de la Pitié Salpêtrière, F-75013, Paris, France*

Abstract

Many deep generative models have been proposed to reconstruct pseudo-healthy images for anomaly detection. Among these models, the variational autoencoder (VAE) has emerged as both simple and efficient. While significant progress has been made in refining the VAE within the field of computer vision, these advancements have not been extensively applied to medical imaging applications.

We present a benchmark that assesses the ability of multiple VAEs to reconstruct pseudo-healthy neuroimages for anomaly detection in the context of dementia. We first propose a rigorous methodology to define the optimal architecture of the vanilla VAE and select the best hyper-parameters of the VAE variants. Relying on a simulation-based evaluation framework, we thoroughly assess the ability of 20 VAE models to reconstruct pseudo-healthy images for the detection of dementia-related anomalies in 3D brain FDG PET and compare their performance.

This benchmark demonstrated that the majority of the VAE models tested were able to reconstruct images of good quality and generate healthy looking images from simulated images presenting anomalies. Even if no model clearly outperformed all the others, the benchmark allowed identifying a few models that perform slightly better than the vanilla VAE. It further showed that many VAE-based models can generalize to the detection of anomalies of various intensities, shapes and locations in 3D brain FDG PET.

Keywords: Variational autoencoder, Unsupervised anomaly detection, Deep generative models, PET, Alzheimer's disease

1. Introduction

The synergy between innovations in imaging technologies, the growing volume of medical data, and sophisticated machine learning algorithms have given rise to algorithms capable of performing complex tasks such as anomaly detection for computer aided diagnosis [1]. A first approach consists in using supervised learning algorithms to segment lesions. Whilst this approach performs well, it requires human annotations which are costly in time and money, and leads to very specialized models. Another strategy consists in using unsupervised algorithms in order to learn the distribution of healthy data and to detect anomalies as deviations from this learned prior knowledge [2]. This is particularly useful as it does not require annotated data, and should generalize to any type of anomalies, without having seen them before.

A prevailing approach in unsupervised anomaly detection (UAD) consists in using generative models to reconstruct healthy looking images, or pseudo-healthy images [1, 2, 3]. The pseudo-healthy image can then be compared with the real image to detect anomalies, and possibly further segment lesions. The underlying assumption is that a model trained on images from subjects diagnosed as healthy should generate images that do not contain pathology-specific features and look like a healthy image, even when provided a pathological image as input.

Pseudo-healthy reconstruction approaches that have been developed for medical imaging often rely on generative models such as variational autoencoders (VAEs) [4], generative adversarial networks (GANs) [5] and more recently diffusion models [6]. Even though diffusion models have shown remarkable performance for image generation, they do not easily scale to 3D images [7], mainly because of memory issues. On the

[1]Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

GAN side, after the foundational work of Schlegl et al., AnoGAN [8] and f-AnoGAN [9], only a few works have been published. They either use cycle GANs [10] or combine GANs with autoencoders [11, 12]. On the other hand, even though VAEs' image generation quality is lower, they are easy to train, they scale well to high-dimensional data, provide good interpretation capacity thanks to their regularized latent space, and are able to handle small datasets. Many new VAE extensions have shown their efficacy in the computer vision literature [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29], but only a handful have been applied to medical imaging [3, 30, 31, 32, 33].

In 2021, Baur et al. [3] compared VAE-based approaches to the best GANs for unsupervised anomaly segmentation in brain structural magnetic resonance imaging (MRI). It was conducted on models that had already been employed for UAD in the medical imaging context such as Context VAE [34], Constrained AAE [30], or AnoVAEGAN [35]. They showed that the vanilla VAE used for density-based restoration [36] outperforms other models, including GAN approaches, at the cost of a longer inference time. They compared the performance of their models using segmentation metrics such as the dice similarity coefficient (DSC) and the area under profile curve ratio (AUPCR) computed between the residual (i.e. the difference between the input and the reconstructed image) and the ground truth anomaly mask provided in the datasets they used. This study focused on the segmentation of glioblastoma and multiple sclerosis lesions, which consist in sharp and intense anomalies that are segmented in 2D slices extracted from MRI volumes.

In this work, we aim to apply UAD methods to identify metabolic changes associated with Alzheimer's disease and other dementias [37] that are visible in brain ^{18}F-fluorodeoxyglucose (FDG) positron emission tomography (PET) images. ^{18}F-FDG PET images are 3D images that highlight the concentration of administered FDG, a tracer used to localize hypometabolism in the case of neurodegeneration [38]. This application is particularly interesting as deep learning methods for UAD have rarely been applied for the diagnosis of dementia [31], whereas this approach could enable early diagnosis since changes visible in neuroimaging can occur years before the onset of initial symptoms [39]. The metabolic abnormalities are however subtle and difficult to detect as they are diffuse and of low intensity [40], contrary to glioblastoma or white matter hyper-intensities usually studied on structural MRI [3, 10, 30, 34].

To continue the work of Baur et al. [3], we propose a benchmark of 20 VAE-based models focused on the pseudo-healthy reconstruction of 3D FDG PET images in the context of dementia. We compare many VAE-based models that have not been applied to medical image analysis yet, thanks to the benchmark of Chadebec et al. [41]. In contrast to computer vision works, where datasets typically contain several tens of thousands images, it will be interesting to examine the performance of such models when trained on a relatively small dataset, comprising only a few hundred images, which is typical in medical imaging. Our contributions are threefold:

1. first, we propose a rigorous method and provide the associated software tool that we used to define the optimal architecture of the vanilla VAE and select the best hyper-parameters of the VAE variants in the context of neuroimaging;
2. then, we put in application the evaluation framework introduced by Hassanaly et al. [42] to thoroughly assess the ability of 20 VAE models to reconstruct pseudo-healthy images for the detection of dementia-related anomalies in 3D brain FDG PET and compare their performance;
3. finally, we conclude on the best performing models, providing a state-of-the-art on the use of 3D convolutional VAEs in such context.

A preliminary version of this work was published as a conference paper [43]. The present article extends the previous work mainly with: (i) the addition of new VAE-based models; (ii) an extensive search of the best encoder-decoder architecture and hyper-parameters for each model; (iii) the use of full resolution 3D brain FDG PET; (iv) and an extensive evaluation of the different models.

2. Pseudo-healthy image reconstruction with variational autoencoders

2.1. Variational autoencoder framework

The VAE framework [4] assumes that a latent variable $\mathbf{z}$ is involved in the generation process of the input data $\mathbf{x}$: $p_\theta(\mathbf{x}) = \int_z p(\mathbf{z})p_\theta(\mathbf{x} \mid \mathbf{z})\mathrm{d}z$ where $\mathbf{z} \sim p_\theta(\mathbf{z})$ is the prior distribution on the latent space and $p_\theta(\mathbf{x} \mid \mathbf{z})$ is the generative model (or the decoder). To compute the appropriate $\mathbf{z}$ for each data input $\mathbf{x}$ of the dataset, the posterior distribution $p_\theta(\mathbf{z} \mid \mathbf{x})$ needs to be modeled. Since it is untractable, the posterior distribution is approximated using variational inference by introducing another model $q_\phi(\mathbf{z} \mid \mathbf{x})$ such that $q_\phi(\mathbf{z} \mid \mathbf{x}) \approx p_\theta(\mathbf{z} \mid \mathbf{x})$. $q_\phi(\mathbf{z} \mid \mathbf{x})$ is the inference model (or encoder). Both the decoder and encoder are parametric models whose parameters θ and ϕ are given by a neural network.

The objective is to maximize the likelihood of $p_\theta(\mathbf{x})$, which is equivalent to maximizing the evidence lower bound, which defines the loss function $\mathcal{L}_{\theta,\phi}$

$$\log\left(p_\theta(\mathbf{x})\right) \geq \mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log\big(p_\theta(\mathbf{x} \mid \mathbf{z})\big)\Big] - D_{\mathrm{KL}}\Big(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\Big) \tag{1}$$

with D_{KL} the Kullback-Leibler divergence.

The VAE is particularly suited for pseudo-healthy reconstruction. Let's consider D a set of medical images collected following a similar protocol. D contains healthy and pathological images and is the union of two complementary subsets D_h and D_p. For instance, D could be a set of healthy FDG PET images $\mathbf{x} \in D$ whose distribution is $p(\mathbf{x})$. The goal of pseudo-healthy image reconstruction is to reconstruct an FDG PET image of healthy appearance given an input $\mathbf{x} \in D$. During the training process, an approximation of the posterior distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ is learned for $x \in D_h$ as the model is trained using only healthy subjects. In other words, the healthy image true distribution $p(\mathbf{x})$ is approximated with the learned parametric distribution $p_\theta(\mathbf{x})$ such that $p_\theta(\mathbf{x}) \approx p(\mathbf{x})$. During reconstruction, this approximate posterior is used to estimate the latent variable $\mathbf{z}$ for $\mathbf{x} \in D$ (it can be from D_h or D_p), i.e., the images (of healthy subjects or patients) are projected into that "healthy images" learned subspace. Then, the decoder can generate healthy images from $\mathbf{z}$.

2.2. Extensions to the variational autoencoder framework

Several contributions have been proposed to improve the VAE framework [41]. These contributions can be divided into four categories that correspond to different objectives.

The aim of the first category of approaches is to improve the prior distribution $p(\mathbf{z})$ by using a variational mixture of posteriors as prior (VAMP) [27] or using a specific geometry in the latent space such as hyperspherical VAE (SVAE) [17], by learning the prior on a discrete latent space with vector quantized-VAE (VQVAE) [28], or by substituting the prior with a density estimation method using regularization with a gradient penalty (RAE-GP) or an ℓ^2 penalty on the decoder (RAE-ℓ^2) [18].

Other methods aim to better estimate the lower bound by using importance weighting (IWAE) [13], by using linear normalizing flows (VAE LinNF) [24], inverse autoregressive flows (VAE-IAF) [21] or Markov chain Monte Carlo using Hamiltonian importance sampling (HVAE) [15] to better estimate the posterior.

Approaches in the third category encourage disentanglement of the features in the latent space by adding a weight to balance the terms of the loss in Eq. 1 (β-VAE) [19], subsequently improved with a better reconstruction capacity by progressively increasing the KL-divergence term (Disentangled β-VAE) [14], by decomposing the loss to show a total correlation term (β-TC VAE) [16], or by encouraging the distribution of the latent variable $q(\mathbf{z})$ to be factorial (FactorVAE) [20].

Finally, other methods change the distance computed between the distributions by adding the mutual information between $\mathbf{x}$ and $\mathbf{z}$ as regularization (InfoVAE) [29], using another divergence term in the loss such as the maximum mean discrepancy in the Wasserstein autoencoder (WAE) [26] or a discriminator to differentiate a prior's sample from a posterior's sample in the adversarial autoencoder (Adv. AE) [23], or by changing the reconstruction metric for another similarity metric such as the multi-scale structural similarity (MS-SSIM VAE) [25], or for the prediction of a discriminator on the output of the VAE (VAEGAN) [22].

All of these models, described in more detail in Appendix D, perform well in computer vision, as shown by Chadebec et al. [41] who compared 19 of them on classic computer vision datasets (MNIST, CIFAR10 and CELEBA) on five tasks: image reconstruction, image generation, classification, clustering and interpolation. In the proposed benchmark, they will be compared with the autoencoder (AE) and VAE [4] in the context of medical imaging.

2.3. Selection and evaluation of the models

When evaluating unsupervised anomaly detection approaches, two aspects are usually assessed: their ability to reconstruct images of high quality and their ability to detect anomalies. The first aspect can only be fully assessed when reconstructing images of healthy subjects. Commonly used metrics are the mean-squared error (MSE), the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [44]. These paired metrics are computed between the input and reconstructed images [45]. To assess the second aspect, most studies [3, 10, 12, 30, 34, 46, 47, 48, 49, 50, 51, 52, 53] rely on ground truth anomaly masks, making such evaluation similar to that of supervised segmentation methods using metrics such as the DSC or the AUPCR [3]. To go further, Xia et al. [10] defined two new metrics: an identity metric to measure whether the image reconstructed corresponds to the same subject as the image given in input, and a healthiness metric to measure whether the reconstructed image appears healthy. However, these

metrics use the a priori information of the anomaly mask, which limits their application. In situations when the ground truth anomaly mask is unavailable, the pseudo-healthy reconstructions can be evaluated using a classifier trained to differentiate pathological and healthy images [40], or using an anomaly score derived from the reconstruction error (supposed to be low for healthy images and high for pathological images) [31, 52].

In our context, the detection of hypometabolism in brain FDG PET images, no ground truth anomaly masks are usually available. We thus previously proposed a simulation framework [42, 54] that consists in simulating the effect of a disease on images of healthy subjects by reducing the PET uptake within areas of the brain associated with different dementias [40] defined using a mask M. After locally reducing the intensity of the image by a certain percentage within M, a Gaussian smoothing is applied to generate a realistic result and diffuse anomalies. This approach effectively replicates realistic regional hypometabolism and provides pairs of diseased images with the original healthy scan that is used as the target ground truth for the pseudo-healthy reconstruction. Hypometabolism can be simulated in different parts of the brain, which allows assessing the models' ability to generalize to anomalies caused by different dementia subtypes. Furthermore, for a given dementia subtype, different severity degrees can be simulated by adjusting the intensity reduction. This allows investigating the sensitivity of the UAD approaches to both subtle and severe anomalies. All the details about the masks used and the simulation pipeline are in [42, 54].

In [42], we also proposed a metric, which relies on the simulation framework, to evaluate whether a model is able to reconstruct images that are looking healthy. This healthiness score $\mathcal{H}$ is defined as

$$\mathcal{H} = \frac{\mu_M}{\mu_{\bar{M}}} \ , \tag{2}$$

with μ_M the average uptake within the mask M used to simulate the anomaly and $\mu_{\bar{M}}$ the average uptake of voxels in the brain excluding the mask M. The healthiness score compares the average uptake in the region in which we simulate the disease and the other regions of the brain. It is supposed to be around 1 for images of healthy subjects, lower than 1 for simulated images and expected to be around 1 again for the pseudo-healthy reconstructions.

As we consider that to accurately detect anomalies a model should reconstruct pseudo-healthy images of high quality, we use the pairwise performance measures as a first step in our evaluation. We especially rely on the SSIM, rather than the MSE or PSNR, as it is a perceptual metric that appears more informative than a pixel-wise difference, and because it is a different metric than the optimization criterion, which is MSE for all the models except for the MS-SSIM VAE [25]. In particular, the SSIM is used as selection criterion when searching for the best hyper-parameters' configurations and selecting the best trained models, and is combined with the MSE when searching for the best encoder-decoder architecture. We use the simulation framework with the healthiness metric in a second step to push further the evaluation of the trained models being compared.

3. Materials

As in [42], FDG PET scans used in this study were obtained from the ADNI database [55, 56]. The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment and early AD.

We selected FDG PET images co-registered, averaged and uniformized to a resolution of 8 mm full width at half maximum to reduce the variability due to the use of different scanners. We only used PET images for which a T1-weighted (T1w) MR image was available at the same session for preprocessing purposes. The images were then processed using Clinica's [57] `pet-linear` pipeline: they were registered using a rigid transformation to the corresponding T1w MRI of the same session, and then affinely registered to the MNI ICBM 2009c Nonlinear Symmetric template [58, 59] using the transformation computed with the `t1-linear` pipeline. They were then normalized in intensity using the average PET uptake in a region comprising cerebellum and pons, and cropped. In the end, the dimension of the PET scan is $169 \times 208 \times 179$ with 1 mm isotropic voxels. To filter out potential PET images not well registered to the MNI template, we used the quality control pipeline of ClinicaDL [60].

A total of 3511 FDG PET scans from 1600 participants were downloaded from the ADNI database. This includes 554 cognitively normal (CN) subjects (1010 images) that we selected since UAD models are trained only on images from healthy subjects. We know that physiological changes can appear several years before the first clinical symptoms, so to ensure that images really correspond to a healthy brain, we

kept only scans from subjects that are CN for at least three years after the session considered. All the details about data selection are in Appendix A. Our final dataset comprises 739 images from 378 CN subjects.

We split our dataset of 378 CN subjects into training and test sets at the subject's level to avoid any form of data leakage [61], stratifying by sex and age to reduce biases. Only baseline sessions were kept in the test set to avoid biased results. The test set, comprising 60 CN subjects (60 images), is used to assess whether the healthy images are reconstructed as healthy. We denote it as "Test CN". We then performed a six-fold cross-validation on the training data to estimate the variance due to data splitting [62]. 53 subjects (53 images) belong to the validation sets to monitor the training and 265 subjects (between 510 and 538 images depending on the fold) are used to train our models. The training split and test set statistics are summarized in Table 1.

Table 1: Summary of participant demographics at baseline for the different training/validation splits and test sets considered. Note that split s corresponds to using fold s from the 6-fold cross-validation as validation set and the other folds as training set.

		Set	# subjects (%F)	# images	Avg age (± SD)	Age range
Training / validation	split 0	train	265 (52.8%)	536	74.9 ± 6.1	55.8, 95.0
		validation	53 (41.5%)	53	72.8 ± 5.5	62.3, 85.3
	split 1	train	265 (53.5%)	533	74.7 ± 6.1	55.8, 95.0
		validation	53 (37.7%)	53	73.4 ± 5.6	59.9, 88.9
	split 2	train	265 (49.8%)	537	74.6 ± 5.9	55.8, 95.0
		validation	53 (56.6%)	53	74.4 ± 6.6	63.8, 93.6
	split 3	train	265 (48.6%)	538	74.7 ± 6.0	55.8, 95.0
		validation	53 (62.2%)	53	73.9 ± 6.2	61.2, 86.2
	split 4	train	265 (49.1%)	511	75.0 ± 5.9	59.7, 95.0
		validation	53 (60.4%)	53	72.4 ± 6.5	55.8, 84.7
	split 5	train	265 (51.6%)	510	74.6 ± 5.9	55.8, 93.6
		validation	53 (47.2%)	53	73.1 ± 6.6	59.7, 92.8
Test		CN test	60 (63.3%)	60	73.5 ± 6.6	59.8, 85.8
		AD test	353 (41.1%)	353	75.3 ± 7.6	55.1, 90.3

We also use the 60 images from the CN test set to build new test sets by using our simulation method [42]. We simulate AD with nine intensity levels from 5% to 70% hypometabolism and simulate five dementia subtypes at 30% hypometabolism: AD, behavioral variant frontotemporal dementia (bvFTD), logopenic variant primary progressive aphasia (lvPPA), semantic variant PPA (svPPA) and posterior cortical atrophy (PCA), which results in a total of 14 simulated test sets. These test sets are denoted using the dementia simulated and the hypometabolism intensity. For instance, "Test AD 30" corresponds to images simulating AD with a 30% hypometabolism.

In addition to the images of CN subjects, the data downloaded from the ADNI dataset include FDG PET images of AD patients. After applying the same data selection procedure as the one used for the CN subjects (explained in Appendix A), we keep 353 images at baseline from 353 AD patients for testing purposes.

4. Model selection

We aim to compare 20 AE and VAE-based models. For the comparison to be meaningful, we must find the best architectures and parameters for each model. We decided to use the same encoder-decoder architecture for all of the models as it would have been too long to find an optimal architecture for each model, and as we believe it makes the comparison fairer. The architecture was obtained using a random search on the vanilla VAE. We then attempted to find optimal hyper-parameters for each model through either a random search or a grid search. Once the best parameters for each model were found, we trained them on all the six splits of the cross-validation and we selected the best. The best trained models were finally evaluated using the simulation framework and metric presented in Section 2.3. The procedure is summarized in Figure 1. The random search and evaluation procedures are implemented in ClinicaDL [60] while the VAE-based models are implemented in Pythae [41], which are both open-source tools (clinicadl.readthedocs.io, pythae.readthedocs.io).

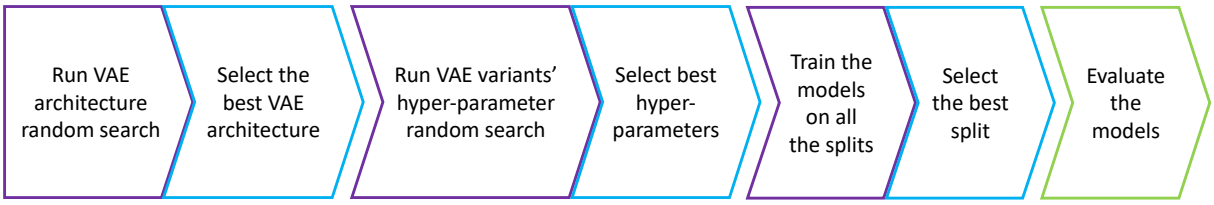
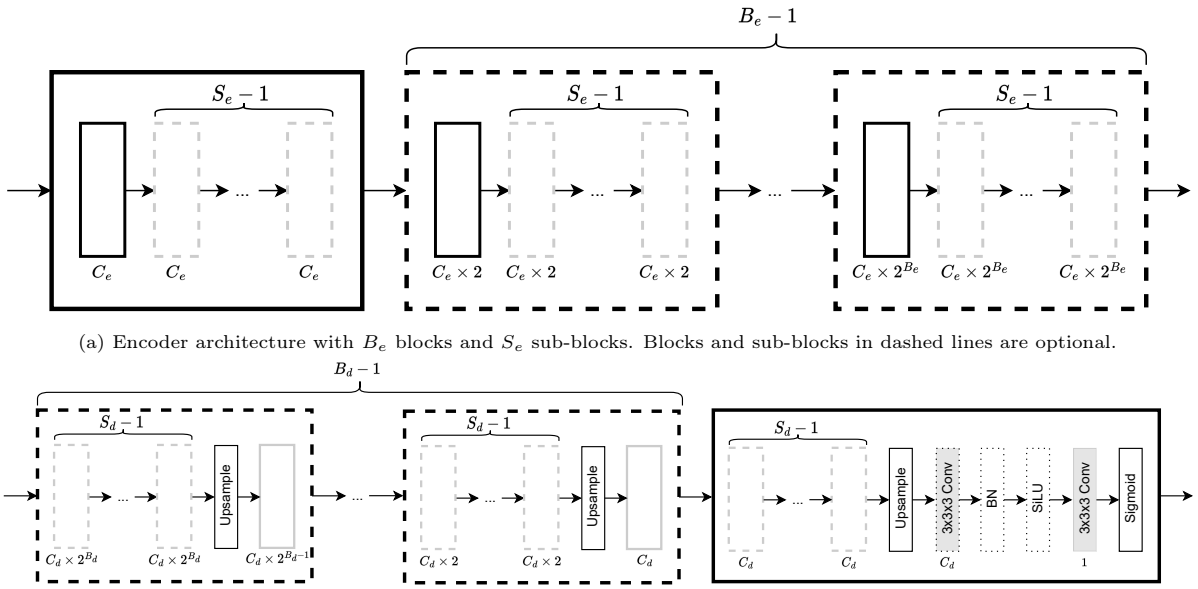


Figure 1: Diagram summarizing the benchmark steps. We represent steps performed on training sets in purple (random search and training), the selections on the validation sets are represented in blue and the final evaluation on test sets is in green.

All of the models were trained for 200 epochs on a HPC with Nvidia Tesla V100 GPUs that have 32 GB of dedicated memory. The choice for the batch size and the learning rate will be discussed further in this section. We used the same environment to train all of the models.

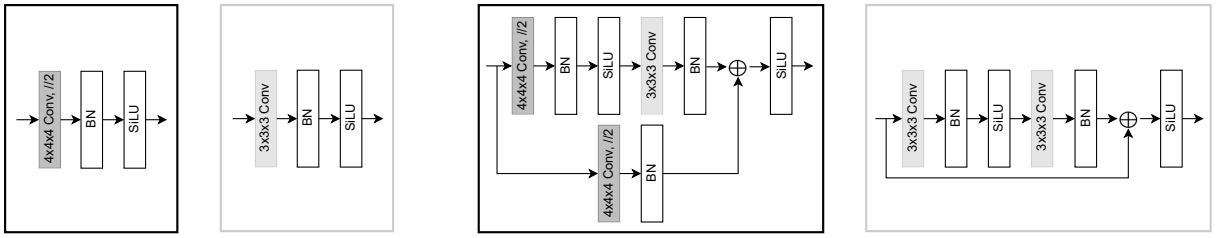
4.1. Selection of the encoder-decoder architecture

The training parameters and the encoder-decoder architecture were selected with a random search for the vanilla VAE. We trained the models on a random selection of three splits as a trade-off between reducing the variance due to data selection and the computational time required to train the models. We then selected the models based on the average SSIM and MSE, computed within the full image field of view, on the validation sets.



(a) Encoder architecture with B_e blocks and S_e sub-blocks. Blocks and sub-blocks in dashed lines are optional.

(b) Decoder architecture with B_d blocks and S_d sub-blocks. Blocks and sub-blocks in dashed lines are optional, and layers in dotted lines are only included if `last_conv = True`.



(c) Sub-block architecture. Left: convolutional sub-blocks. Right: residual sub-blocks. Black sub-blocks: image size is divided by 2 along each dimension by using 4x4x4 convolutions with stride 2 (layers in darker gray), used as the first sub-block of each encoder block. Gray sub-blocks: image size remains the same by using 3x3x3 convolutions with stride 1 (layers in light gray). BN: batch normalization, SiLU: swish activation function

Figure 2: Encoder-decoder modular architecture. The number of convolution kernels in each sub-block is indicated under the sub-block (e.g. C_e).

We defined a modular architecture for the encoder and decoder which is shown in Figure 2. The encoder (shown in Figure 2a) is composed of a number B_e of blocks, each containing a number S_e of

Table 2: Hyper-parameters included in our encoder-decoder VAE architecture random search

Hyper-parameter	Label	Search space	Selected value
Number of encoder blocks	B_e	{4, 5, 6}	5
Number of sub-blocks per encoder block	S_e	{1, 2, 3}	1
Number of channels for the first encoder sub-block	C_e	{16, 32}	16
Number of decoder blocks	B_d	{4, 5, 6}	5
Number of decoder sub-blocks	S_d	{1, 2, 3}	1
Number of channels for the last decoder sub-block	C_d	{16, 32}	16
Latent space size		{256, 512, 1024}	256
Learning rate		$\{10^{-3}, 10^{-4}, 10^{-5}\}$	10^{-4}
Block type		{`conv`, `res`}	`conv`
Added convolution in last decoder block	`last_conv`	{`True`, `False`}	`False`

sub-blocks. Similarly, the decoder is composed of a number B_d of blocks, each containing a number S_d of sub-blocks (Figure 2b). For a chosen architecture, the sub-blocks can either all be convolutional or all be residual (see Figure 2c). In both cases, the convolution layers are followed by a batch normalization and we use a swish activation function as suggested in [63].

In the encoder, the number of channels is doubled by the first convolution in each block. At the same time, the size of the image is divided by 2 along each dimension by using a 3D convolution with kernel size $(4, 4, 4)$, stride $(2, 2, 2)$ and padding $(1, 1, 1)$. The following sub-blocks are optional and their convolution operations have kernel size $(3, 3, 3)$, stride $(1, 1, 1)$ and padding $(1, 1, 1)$. In the decoder, the last sub-block of each block is preceded by an upsampling layer, to be roughly symmetrical with the encoder. Convolution operations in the decoder have kernel size $(3, 3, 3)$, stride $(1, 1, 1)$ and padding $(1, 1, 1)$. This architecture was inspired from ResNet models [64] and VGG models [65].

The parameters of this modular architecture (summarized in Table 2) are therefore the following: the latent space size, the number of blocks in the encoder B_e, the number of blocks in the decoder B_d, the number of sub-blocks per encoder block S_e, the number of sub-blocks per decoder blocks S_d, the number of channels for the first encoder block C_e, the number of channels for the last decoder block C_d, and the layer type (convolution or residual). We implement a random search to explore this parameter space, and choose possible values for each parameter based on previous experiments, intermediate results (as we launched the random search in successive batches) and intuition. We decided to set the batch size of the data loader to 8. Even though this is a constraint for configurations that would require more memory, this choice allows flexibility; in scenarios where certain VAE variants require more memory, we can reduce the batch size while maintaining a reasonable number of images per batch (e.g., 6 or 4). Details of all the parameters tested and their impact are discussed in Appendix B.

After comparing around 200 configurations, the encoder architecture selected is composed of five blocks, each with one sub-block, each containing a convolutional layer, a batch normalization and a swish activation function. These blocks are followed by a flatten and a fully connected layer. The latent space has size 256. The decoder is symmetrical, it is composed of a fully connected layer followed by five blocks, each with one sub-block, each composed of an upsampling layer, a convolutional layer, a batch normalization and a swish activation. This model has 16 channels after the first encoder block and before the last decoder block. This is shown in Figure 3 and detailed in Appendix C.

4.2. Selection of the models' hyper-parameters

Once we found an encoder-decoder architecture that gave good performance, we used it for the AE and 18 VAE variants presented in Section 2.2. However, all of these variants, except the SVAE [17], have supplementary hyper-parameters that may have significant impact on the models' performance. We therefore searched for the best configuration of hyper-parameters for each model in the context of 3D brain FDG PET reconstruction by launching either a random search (when we searched for more than one hyper-parameter) or grid search (when there is only one hyper-parameter). Similarly to the architecture search, we train each configuration on three splits. We then selected the best set of hyper-parameters for each VAE-based model using the best average SSIM on the validation set as criterion.

As there were many models to optimize, we limited the number of random searches to $N \times 10$ with N the number of parameters to search. For instance, when there was only one hyper-parameter to tune, we

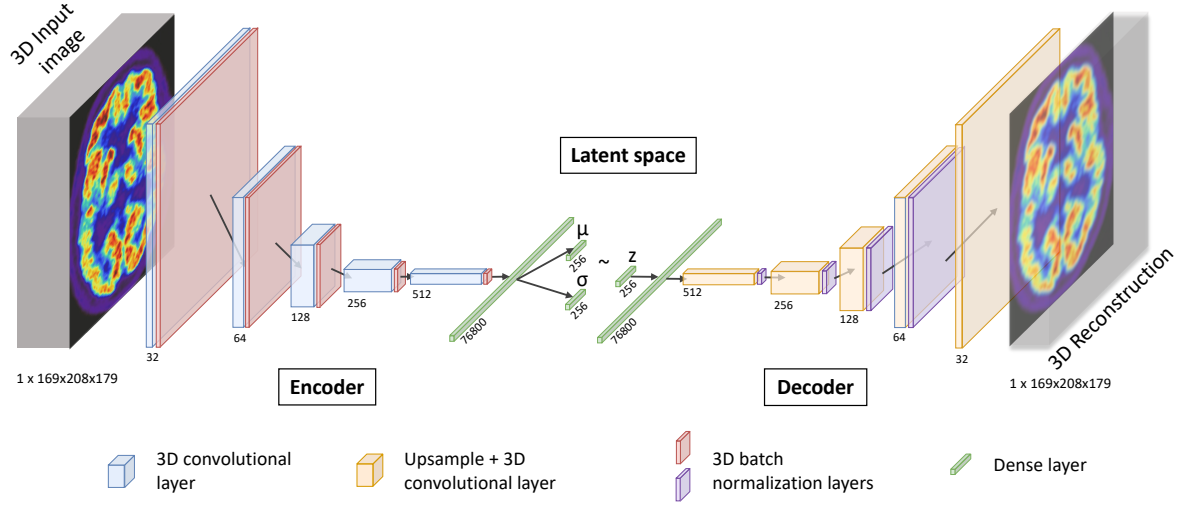


Figure 3: Diagram of the selected VAE architecture

launched a grid search with maximum 10 different values for that parameter, if there were two parameters, we trained a maximum of 20 models and so on. This may not be the fairest decision as it does not allow exploring the same percentage of the parameter space depending on N (as it scales to the power N and not linearly), but it accounts for the fact that a model with more parameters may be more tedious to tune. Moreover, we carefully chose a range of values to test for each hyper-parameter of each model based on the original implementation papers, our prior knowledge of the models, and the work done by Chadebec et al. [41]. Note that some of the hyper-parameters were excluded from our search when an optimal value was provided in the literature, which allowed reducing the number of configurations trained.

Following the vanilla VAE training, the different configurations were trained by default with a batch size of 8 and a learning rate of 10^{-4} on 200 epochs. When some set of hyper-parameters led to memory errors, we gradually reduced the batch size to 6, 4 or 2, and when they lead to errors in the computation of the loss, we reduced the learning rate to 10^{-5}. In spite of this, combinations leading to errors were removed, further reducing the size of the hyper-parameter space.

The details relating to the different VAE-based models, their hyper-parameters, the random search, the trained configurations and the results are provided in Appendix D. A summary is proposed in Table 3.

We report in Table 4 reconstruction metrics for the 18 VAE variants with the best configuration of hyper-parameters that we tested. Out of all of the models, only three did not perform well on the validation set (highlighted in gray): the VAMP [27] with an average SSIM of 0.702, the MS-SSIM VAE [25] with an average SSIM of 0.472 and the SVAE with a very low average SSIM of 0.151. We found it quite surprising that the MS-SSIM VAE [25] performed so poorly in terms of average SSIM since it optimizes a perceptual metric related to the SSIM, namely the multi-scale SSIM. These results could potentially be explained by the fact that the MS-SSIM computation in 3D is very costly, meaning that the only kernel size that allowed training in a reasonable amount of time was 2, potentially leading to a poor estimation of the metric, especially since the kernel size suggested in the MS-SSIM original implementation is 11 [66]. In the end, only three models with three different combinations of parameters were trained successfully, possibly explaining why we did not find a configuration giving acceptable reconstruction. Finally, the SVAE did not train with a high dimensional latent space (hundred and above) due to the computation of the Bessel function in the loss. Since this model does not have any hyper-parameter to tune, we decided to launch a grid search to find the best latent space size (within the set $\{8, 16, 32\}$). The reduction of the latent space size may explain why the reconstruction is not satisfying, as we know that low latent dimensions lead to poorer reconstructions. Moreover, the SVAE seems to be better suited for hyperspherical data distributions, which is not the case in our application. For the following experiments, we decided not to consider the VAMP [27], MS-SSIM VAE [25] and SVAE [17].

4.3. Selection of the best trained models

Once the best parameters were selected through the random search, all 17 models (AE, VAE and the 15 remaining VAE-based models) were trained on the six splits of the cross-validation. We kept the same training parameters as for the random search: the models were trained on 200 epochs with a learning

Table 3: Summary of the hyper-parameters optimized and selected thanks to the random search for each VAE variant. The hyper-parameters are detailed in Appendix D.

Models	Hyper-parameters	Search space	Selected value
Adv. AE [23]	adv. loss scale	{0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99}	0.9
β-TC VAE [16]	β	{0.001, 0.005, 0.01, 0.05, 0.1, 1, 2, 5, 10}	2
	α	{1, 3}	1
	γ	{1, 3}	1
β-VAE [19]	β	{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 2, 5, 10, 100}	10
Dis. β-VAE [14]	β	{0.01, 0.1, 1, 5, 10}	10
	C	{5, 25, 50}	50
	warm-up epochs	{100, 1000}	1000
FactorVAE [20]	γ	{2, 5, 10, 15, 20, 30, 40, 50, 100, 200}	40
HVAE [15]	n lf	{1, 2, 10, 15, 20}	10
	eps lf	{0.00001, 0.0001, 0.001, 0.01}	0.00001
	β_0	{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}	0.8
InfoVAE [29]	kernel choice	{rbf, imq}	rbf
	α	{0.0, 0.2, 0.4, 0.6, 0.8, 1.0}	1
	λ	{0.01, 0.1, 1, 10, 100}	0.1
	kernel bandwidth	{0.01, 0.1, 0.5, 1, 5, 10, 100}	0.1
IWAE [13]	n samples	{2, 3, 4, 5, 6, 8, 10, 12, 15, 20}	6
MS-SSIM VAE [25]	β	{0.01, 0.1, 1, 10, 100}	-
	window size	{2, 3, 5, 11}	-
RAE-ℓ^2 [18]	embedding weight	{0.00001, 0.0001, 0.001, 0.01, 0.1, 1}	0.0001
	reg. weight	{0.00001, 0.0001, 0.001, 0.01, 0.1, 1}	1
RAE-GP [18]	embedding weight	{0.00001, 0.0001, 0.001, 0.01, 0.1, 1}	0.01
	reg. weight	{0.00001, 0.0001, 0.001, 0.01, 0.1, 1}	0.0001
SVAE [17]	latent space size	{8, 16, 32}	-
VAEGAN [22]	adv. loss scale	{0.3, 0.5, 0.7, 0.9}	0.5
	reconstruction layer	{1, 2, 3, 4}	1
VAE-IAF [21]	n made blocks	{2, 4, 6, 8}	4
	n hidden in made	{2, 3, 4, 5}	4
	hidden size	{64, 128, 256}	128
VAE LinNF [24]	flows	{10P, 10R, 5P, 5R, 5P5R, 5R5P, 5PR, 5RP, 2PR, 2RP}	10R
VAMP [27]	number components	{10, 20, 30, 40, 50}	-
	linear scheduling steps	{0, 20, 40}	-
VQVAE [28]	quantization loss factor	{0.25, 0.5, 0.75, 0.9, 1, 1.5, 2, 4}	2
	n embeddings	{128, 256, 512, 1024}	512
WAE [26]	kernel choice	{rbf, imq}	rbf
	reg. weight	{0.01, 0.1, 0.5, 1, 5, 10, 100}	0.1
	kernel bandwidth	{0.01, 0.1, 0.5, 1, 5, 10, 100}	5

Table 4: Reconstruction metrics obtained for the best configuration of each VAE variant on the validation sets (mean ± std computed over the three splits randomly selected)

Models	SSIM ↑	MSE ($\times 10^{-3}$) ↓	PSNR ↑
β-TC VAE [16]	0.870 ± 0.002	1.901 ± 0.123	27.41 ± 0.27
β-VAE [19]	0.868 ± 0.003	1.995 ± 0.067	27.17 ± 0.14
Dis. β-VAE [14]	0.874 ± 0.006	2.004 ± 0.153	27.18 ± 0.30
FactorVAE [20]	0.876 ± 0.003	1.895 ± 0.084	27.47 ± 0.14
HVAE [15]	0.873 ± 0.007	1.862 ± 0.068	27.48 ± 0.14
InfoVAE [29]	0.877 ± 0.006	1.813 ± 0.075	27.63 ± 0.13
IWAE [13]	0.865 ± 0.007	2.087 ± 0.146	27.02 ± 0.24
MS-SSIM VAE [25]	0.472 ± 0.034	70.174 ± 5.660	11.61 ± 0.36
RAE-GP [18]	0.880 ± 0.006	1.715 ± 0.105	27.84 ± 0.26
RAE-ℓ^2 [18]	0.884 ± 0.005	1.815 ± 0.049	27.61 ± 0.11
SVAE [17]	0.151 ± 0.001	632.694 ± 5.106	1.99 ± 0.03
VAEGAN [22]	0.860 ± 0.013	2.241 ± 0.193	26.64 ± 0.38
VAE-IAF [21]	0.823 ± 0.005	2.272 ± 0.057	26.65 ± 0.08
VAE LinNF [24]	0.871 ± 0.001	1.855 ± 0.125	27.54 ± 0.21
VAMP [27]	0.702 ± 0.097	5.581 ± 0.874	22.73 ± 0.72
VQVAE [28]	0.881 ± 0.003	1.805 ± 0.032	27.62 ± 0.07
WAE [26]	0.881 ± 0.005	1.862 ± 0.075	27.54 ± 0.08

rate of 10^{-4} and a batch size of 8. There were a few exceptions: the VAE-IAF [21] was trained with a learning rate of 10^{-5} to avoid errors during training. The RAE-GP [18] was trained with a batch size of 6, the VAEGAN [22] with a batch size of 4 and the IWAE [13] with a batch size of 2 because of the high memory usage of these models.

We then selected the best fold for each model using the average SSIM on the validation sets. The performance of all 17 models on the six splits are presented in Appendix E, with the best split of each model highlighted in bold. We can notice that the splits 2 and 3 are over-represented among the selected models. This can be explained by the fact that the cross-validation is not stratified and the distributions of age and sex between training and validation sets for split 2 and 3 are more similar than for the other splits (Table 1).

5. Results obtained for the best models on the test sets

Once all the models with a correct reconstruction were trained and the best model was selected (optimal set of parameters among those tested and best split), we could evaluate each model using the procedure defined in Section 2.3. Pseudo-healthy images were reconstructed for each of the 15 test sets (the test set with the images of healthy subjects and the 14 test sets with simulated images) in order to measure the performance of the models both qualitatively by visualizing the pseudo-healthy reconstructions, and quantitatively by computing the reconstruction metrics and the healthiness score.

5.1. Quantitative evaluation of the pseudo-healthy reconstructions from images of control subjects

We first assessed whether the different models could correctly reconstruct images of healthy subjects from the test set by computing the SSIM, MSE and PSNR between the input and the pseudo-healthy reconstruction. Results are reported in Table 5. We observe that the reconstruction metrics of all but two models are in the same order of magnitude, with an SSIM on average between 0.873 (VAE [4]) and 0.887 (RAE-GP [18]), an MSE on average between 1.6×10^{-3} for the RAE-GP [18] and 1.859×10^{-3} for the IWAE [13], and a PSNR on average between 26.7 (VAEGAN [22]) and 28.1 (RAE-GP [18]). This shows that the RAE-GP [18] has the best reconstruction capacity. On the other hand, the VAEGAN [22] and the VAE-IAF [21] perform the worst, with respectively an average SSIM of 0.866 and 0.837, and an average MSE of 2.195×10^{-3} and 2.099×10^{-3}, which is even worse than the vanilla VAE and the AE.

Table 5: Reconstruction metrics obtained for images from Test CN (mean ± std computed over images from the test set)

Models	SSIM ↑	MSE ($\times 10^{-3}$) ↓	PSNR ↑
AE	0.882 ± 0.026	1.649 ± 0.613	28.00 ± 1.10
Adv. AE [23]	0.882 ± 0.028	1.707 ± 0.610	27.83 ± 1.06
β-TC VAE [16]	0.878 ± 0.025	1.720 ± 0.565	27.79 ± 1.05
β-VAE [19]	0.876 ± 0.027	1.846 ± 0.638	27.49 ± 1.04
Dis. β-VAE [14]	0.880 ± 0.023	1.841 ± 0.634	27.50 ± 1.06
FactorVAE [20]	0.879 ± 0.026	1.651 ± 0.584	27.98 ± 1.06
HVAE [15]	0.882 ± 0.024	1.809 ± 0.635	27.59 ± 1.09
INFOVAE [29]	0.883 ± 0.024	1.704 ± 0.594	27.84 ± 1.04
IWAE [13]	0.876 ± 0.026	1.859 ± 0.564	27.44 ± 1.03
RAE-GP [18]	0.887 ± 0.023	1.605 ± 0.671	28.14 ± 1.13
RAE-ℓ^2 [18]	0.882 ± 0.024	1.631 ± 0.531	28.02 ± 1.02
VAE [4]	0.873 ± 0.028	1.736 ± 0.566	27.75 ± 1.02
VAEGAN [22]	0.866 ± 0.027	2.195 ± 0.641	26.72 ± 1.04
VAE-IAF [21]	0.837 ± 0.027	2.099 ± 0.720	26.92 ± 1.00
VAE LinNF [24]	0.881 ± 0.023	1.807 ± 0.610	27.58 ± 1.05
VQVAE [28]	0.884 ± 0.026	1.649 ± 0.593	27.99 ± 1.10
WAE [26]	0.883 ± 0.026	1.651 ± 0.618	27.99 ± 1.10

5.2. Quantitative evaluation of the pseudo-healthy reconstructions from images with simulated dementia

The first evaluation step with simulated data is to compute reconstruction metrics between the pseudo-healthy reconstructions obtained from these simulated data and the ground truth images used to simulate hypometabolic images, which are the targets. These results are reported in Table 6. For all the models, the reconstructions are slightly worse than for images reconstructed from the ground truth itself (Table 5) with an average SSIM between 0.854 (VAEGAN [22]) and 0.878 (RAE-GP [18]), an average MSE between $1.997 \times 10^{-}3$ for the RAE-ℓ^2 [18] and 2.650×10^{-3} for the VAEGAN [22], and an average PSNR between 25.88 (VAEGAN [22]) and 27.12 (RAE-ℓ^2 [18]). The only exception is the VAE-IAF [21] for which the SSIM increases from 0.837 on average to 0.842. However the reconstruction metrics are still quite high, meaning that the reconstructions from simulated hypometabolic images are similar to their target.

5.3. Qualitative evaluation of the pseudo-healthy reconstructions

Examples of pseudo-healthy reconstructions obtained from the original image of a control subject and images with simulated dementia are displayed in Figure 4. We first observe that all the models are able to reconstruct the input image of a healthy subject. We can recognize the shape of the brain, the areas with high metabolism (gray matter) and the others with a lower metabolism (white matter, ventricles). The VAE-IAF [21] reconstruction has an artifact in the precuneus, which appears as a spherical hypermetabolism. This probably explains why the average SSIM is lower for the VAE-IAF [21] than for other models. We can also see that the VAEGAN [22] tends to reconstruct the image with a higher average intensity as shown by the fact that the difference map is mostly negative (meaning that the reconstruction's voxel values are superior to the input's voxel values).

When reconstructing images with different degrees of simulated AD, we observe that all the models are able to reconstruct images that are visibly healthy by correcting the hypometabolism simulated. On the difference maps, we can recognize the mask used for the simulation as an anomaly, meaning that the model is able to reconstruct pseudo-healthy images. From this qualitative analysis, the models that seem to perform the best in terms of anomaly detection are the VAE-IAF [21] (excluding the fact that it reconstructs an artifact), the β-VAE [19], the disentangled β-VAE [14] and the HVAE [15], at least for images with low (AD 15) and medium (AD 30) severity. It is indeed possible to better distinguish the abnormal area in both hemispheres on the difference maps and the reconstruction errors do not hide the anomaly.

Additional examples of pseudo-healthy reconstructions obtained for different subjects and different simulated dementias are displayed in Appendix F.

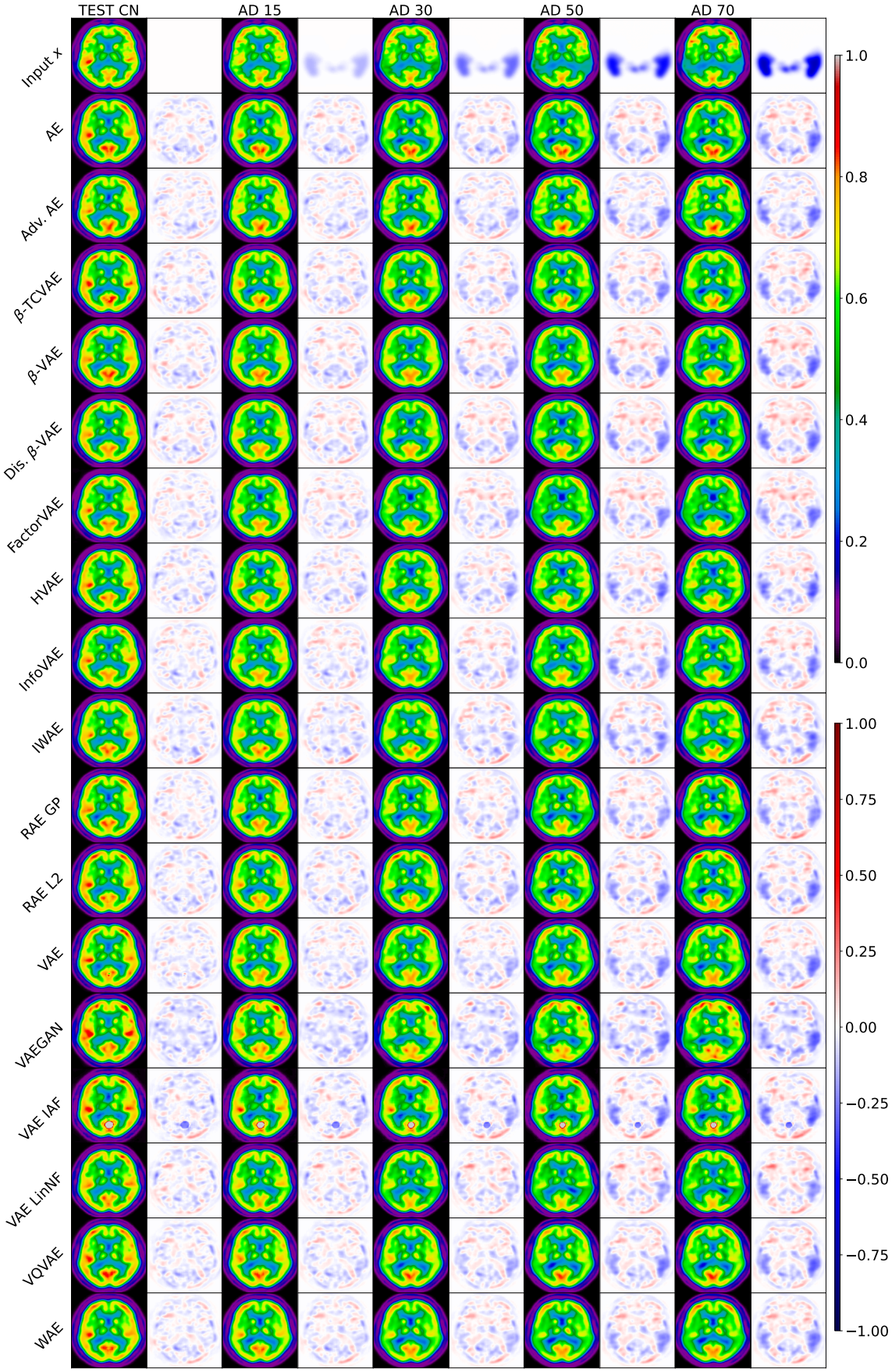


Figure 4: Examples of reconstructions obtained with the different VAE variants from the original image of a cognitively normal subject (images of the first column, Test CN) and from the same subject with AD simulated at different intensity degrees (AD 15, AD 30, AD 50 and AD 70). The first row shows the input image in odd columns and the mask of the simulated disease in even columns when the input is a simulated image. All the other rows are the pseudo-healthy reconstructions of the models in odd columns and the difference between the pseudo-healthy reconstruction and the input in even columns.

Table 6: Reconstruction metrics obtained between pseudo-healthy reconstructions obtained from the simulated images of Test AD 30 and the ground truth images (mean ± std computed over images from the test set)

Models	SSIM ↑	MSE ($\times 10^{-3}$) ↓	PSNR ↑
AE	0.876 ± 0.025	2.067 ± 0.643	26.99 ± 1.03
Adv. AE [23]	0.878 ± 0.027	2.021 ± 0.620	27.07 ± 0.99
β-TC VAE [16]	0.871 ± 0.024	2.094 ± 0.605	26.92 ± 1.02
β-VAE [19]	0.873 ± 0.026	2.107 ± 0.678	26.90 ± 1.00
Dis. β-VAE [14]	0.874 ± 0.023	2.158 ± 0.667	26.80 ± 1.02
FactorVAE [20]	0.873 ± 0.024	2.157 ± 0.603	26.78 ± 0.94
HVAE [15]	0.876 ± 0.023	2.071 ± 0.646	26.98 ± 1.04
INFOVAE [29]	0.878 ± 0.022	2.044 ± 0.595	27.02 ± 0.97
IWAE [13]	0.864 ± 0.027	2.265 ± 0.571	26.55 ± 0.91
RAE-GP [18]	0.878 ± 0.022	2.118 ± 0.690	26.88 ± 0.99
RAE-ℓ^2 [18]	0.877 ± 0.023	1.997 ± 0.564	27.12 ± 0.99
VAE [4]	0.870 ± 0.027	2.075 ± 0.589	26.95 ± 0.95
VAEGAN [22]	0.854 ± 0.027	2.650 ± 0.662	25.88 ± 0.97
VAE-IAF [21]	0.842 ± 0.025	2.322 ± 0.735	26.47 ± 0.97
VAE LinNF [24]	0.876 ± 0.022	2.179 ± 0.614	26.74 ± 0.97
VQVAE [28]	0.878 ± 0.025	2.089 ± 0.596	26.92 ± 0.97
WAE [26]	0.877 ± 0.025	2.087 ± 0.650	26.95 ± 1.04

5.4. Quantitative evaluation with the healthiness metric

After qualitatively analyzing the pseudo-healthy reconstructions, we computed the healthiness score defined in Section 2.3 for the different simulated test sets.

Figure 5 displays the distribution of the healthiness score for the ground truth (i.e., the images of healthy subjects), the images simulating AD with 30% hypometabolism (AD 30) and the reconstructions obtained for the different models from the AD 30 images. As expected, the healthiness of the ground truth is between 1.0 and 1.08, and it drops to between 0.83 and 0.90 when simulating AD with a hypometabolism intensity of 30%. Studying the healthiness of the pseudo-healthy reconstruction for each model, we first observe that all the models are able to reconstruct images that are healthier than the simulated input as the healthiness of the reconstructions (around 1) is superior to the healthiness of the simulated images they were reconstructed from (around 0.87). We can observe that three models seem to perform slightly better than the others, namely the β-VAE [19], the disentangled β-VAE [14] and the VAE-IAF [21] with a healthiness between 0.97 and 1.04 for the first two and 0.96 and 1.03 for the third. On the other hand, the VAEGAN [22] appears to be the model with the worst performance (with a healthiness between 0.93 and 1.0), followed by the FactorVAE [20] and the RAE-GP [18] (which have a healthiness score between 0.94 and 1.01).

These results are consistent with the qualitative analysis, as we observed that the β-VAE [19], disentangled β-VAE [14] and VAE-IAF [21] seemed to better highlight the simulated anomalies, while the VAEGAN's [22] poor reconstructions tended to hide the anomalies.

We also analyzed the impact of the severity of the simulated disease on the healthiness metric. Figure 6 displays the evolution of the healthiness for all the models with increasing severity of simulated AD from 5% to 70%. We notice that all the models reconstruct images that are decreasingly healthy according to this metric when increasing the severity of the simulated disease. As in the previous experiment, the β-VAE [19] and disentangled β-VAE perform the best for high hypometabolism, followed by the VAE-IAF [21]. The VAEGAN [22] and RAE-GP [18] have the worst performance. However, the healthiness of the reconstruction remains above the one of simulated data, which means that all the models can reconstruct pseudo-healthy images.

Figure 7 displays for various dementia subtypes (PCA, bvFTD, lvPPA, svPPA and nfvPPA simulated at 30%) the distribution of the healthiness computed for the ground truth, the simulated images and the images reconstructed with all the models. All the models have very similar performance with a healthiness between 0.95 and 1 when simulating PCA, between 0.9 and 1.1 for bvFTD, between 0.96 and 1.6 for lvPPA, between 0.68 and 0.87 for svPPA, and between 1.0 and 1.1 for nfvPPA. As for AD, the VAEGAN's [22] performance is slightly lower than that of the other models, and the β-VAE [19] and disentangled

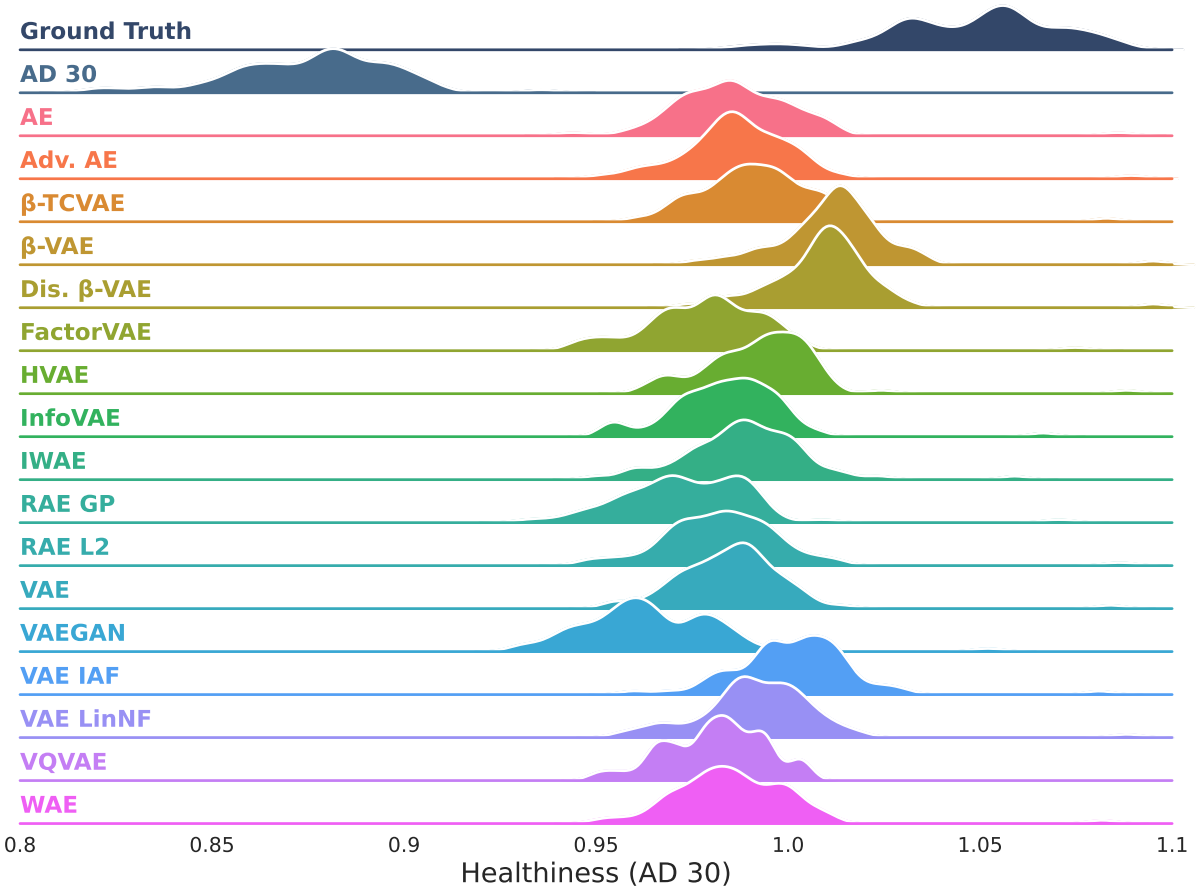


Figure 5: Ridgeline plot showing the distribution of the healthiness metric for images from Test AD 30. The first row corresponds to the healthiness of the ground truth, the second row to the healthiness of the images simulating AD with 30% hypometabolism used as input, and the remaining rows to the healthiness of the pseudo-healthy reconstructions obtained with the VAE models.

β-VAE [14] seem to perform slightly better than the average. We notice that the healthiness of the ground truth, derived from the images of CN subjects, depend on the simulated dementia, given the different masks used for computation. For example, in the case of svPPA, the ground truth's healthiness tends to be lower than that of AD (falling between 0.67 and 0.92). This difference comes from the mask's location in the temporal pole for svPPA, where FDG uptake is naturally lower even in healthy images, in contrast to other regions [67].

To push further the comparison of the models, we jointly analyzed their performance in terms of reconstruction accuracy and healthiness. Figure 8 displays a joint density plot of the SSIM and healthiness metric computed for pseudo-healthy reconstructions obtained from images simulating AD at 30% of hypometabolism. We carefully selected four models that we compare to the VAE: the VAEGAN [22] that performs worse than most models, both in terms of reconstruction accuracy and healthiness, the RAE-GP [18] that has a good reconstruction but a low healthiness performance, the VAE-IAF [21] that has the worst reconstruction accuracy and but a good healthiness, and finally the β-VAE [19] that has both good reconstruction and healthiness performance. This analysis confirms that, among the selected models, the β-VAE is the one that performs the best, and the VAEGAN is the one performing the worst.

5.5. Qualitative analysis of the pseudo-healthy reconstructions obtained from real AD patients

Even though no ground truth is available, it is important to analyze the behavior of the VAE models on data from real patients, here with AD. Figure 9 displays examples of pseudo-healthy reconstructions obtained from the image of an AD patient. This patient presents a typical hypometabolism in the parietal and temporal lobes, which is detected by all the models. However, we also observe for all the models what appears as hypermetabolism in the frontal lobe, which is not typical of AD and probably results from

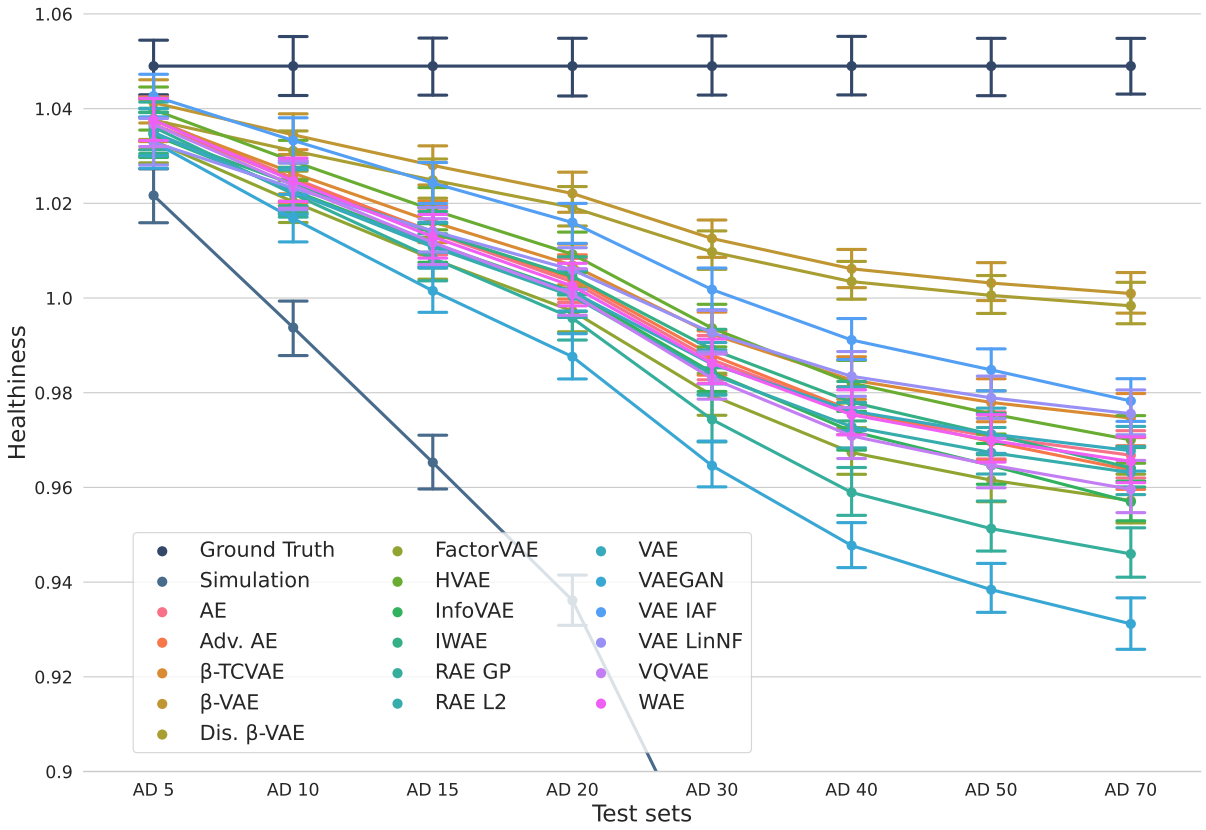


Figure 6: Healthiness metric depending on the severity of the anomalies simulated. The healthiness of the ground truth, which is constant, is displayed as reference. The healthiness of the images simulating AD rapidly drops with the hypometabolism increasing from 5% to 70%. The other curves correspond to the healthiness of the reconstructions obtained with the VAE models. Each dot represents the mean value of the healthiness and the error bar represents the standard deviation.

reconstruction inaccuracies as this tendency was also visible for the CN subject displayed in Figure 4, better seen in Figure F.11.

6. Discussion

This benchmark assessed the ability of 20 VAE models to reconstruct pseudo-healthy 3D brain FDG PET images for anomaly detection. We first searched for the best encoder-decoder architecture for the vanilla VAE. We then optimized the hyper-parameters of all the VAE-based models. After discarding the models with low reconstruction performance, we trained the 17 remaining ones on all the splits of the cross-validation to the select the best split for each model. Finally, we compared the trained models using conventional reconstruction metrics, as well as the simulation framework [42, 54] paired with the healthiness metric [42] we previously proposed.

6.1. Model selection

We performed an extensive random search to define the optimal encoder-decoder architecture for the vanilla VAE. 200 models were trained for a total of approximately 5000 GPU hours. The architecture we obtained is similar to what we could implement following examples and guidelines found in the literature with the objective to obtain a small model that allows fitting 3D high resolution images in the GPU memory: the encoder and decoder are symmetric, they are composed of five blocks, each containing only one 3D convolution layer, a batch normalization and a swish activation [63]. This architecture is for instance very similar to the one we tested in our previous work [42]. Having a small encoder and decoder proves especially advantageous in this benchmark for models with heightened memory-requirements, like the VAEGAN [22] (due to its extra discriminator network), the IWAE [13] (since it uses several samples from the latent space), and the VAE-IAF [21] (since it has extra layers for the auto-regressive flows in the latent space). This architecture was used for all the VAE-based models. Whilst optimizing the

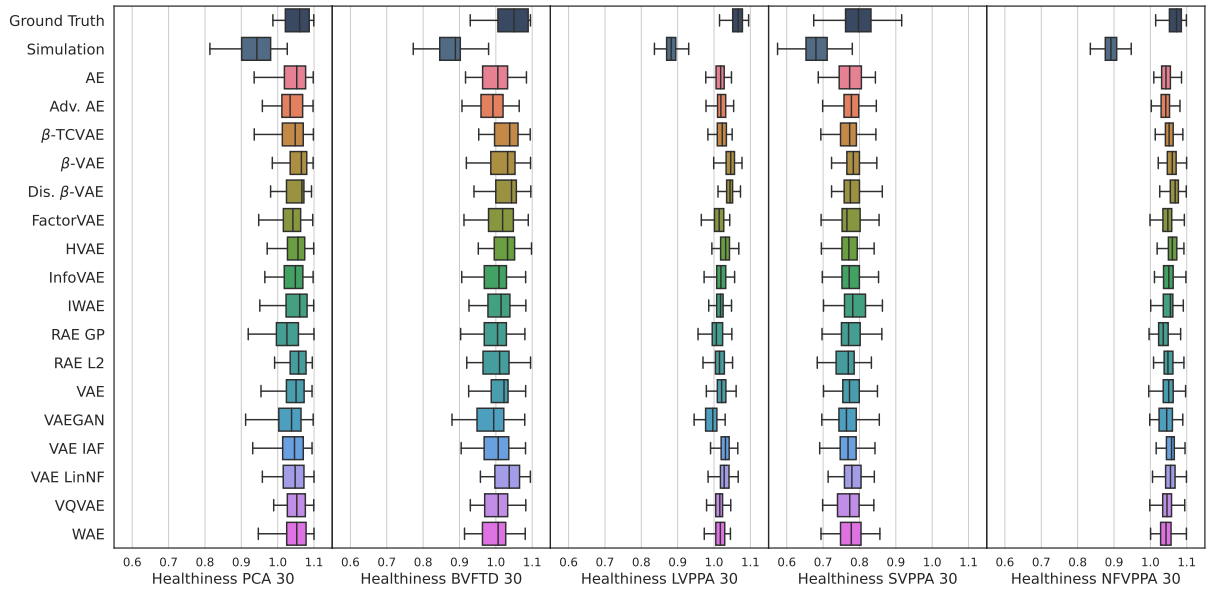


Figure 7: Distribution of the healthiness metric depending on the dementia simulated at 30% hypometabolism: PCA, bvFTD, lvPPA, svPPA and nfvPPA. Each boxplot displays the median, the lower and upper quartiles and the minimum and maximum of the healthiness. The first box (top row) shows the healthiness of the ground truth, the second one the healthiness of the simulated images used as input and the remaining ones the healthiness of the pseudo-healthy reconstructions obtained with the VAE models.

architecture for the vanilla VAE may give this model an advantage, it was not conceivable for us, given our computational resources, to optimize the encoder-decoder architecture separately for all of the models.

To optimize the hyper-parameters of each VAE variant, 324 models were trained for a total of approximately 18,000 hours of GPU use. At this stage we removed three models from the study, as they led to poor reconstructions in comparison with the others: the SVAE [17], the MS-SSIM VAE [25] and the VAMP [27]. For each remaining model, it was possible to find a set of hyper-parameters that led to good reconstruction performance.

After training the models with the selected hyper-parameters on the six splits of the cross-validation, we selected the best split for each of them. We observed that splits 2 and 3 gave the best results for 13 models out of 17 (Table E.28). This can be explained by the fact that the cross-validation was not stratified and so the validation set may not be representative of the training population for some of the splits (Table 1). This may have biased the selection of the hyper-parameters since some models were not trained on splits 2 and 3 when randomly selecting three folds out of six, which may result in underestimated performance for these configurations. However, it would have been too long to train all the configurations on all the splits; and we appraise that we still found a satisfying combination of parameters with respect to the reconstruction metrics, even though it may not be the best one.

All the selection steps were based on the validation sets , potentially leading to over-fitting on these validation sets. Performing a 6-fold cross-validation and randomly selecting the splits reduced this risk.

6.2. Model evaluation

To evaluate the different models, we applied the evaluation procedure presented in [42]. This evaluation consists in two main steps: first measuring the reconstruction performance of the models using reconstruction metrics for images of healthy subjects, and then using simulated data in order to evaluate the ability of the models to reconstruct pseudo-healthy images (i.e. whether the reconstructions appear healthy).

In terms of reconstruction metrics, for the images of healthy subjects (Table 5), all the trained models led to similar performance, with the nine best models having an average SSIM above 0.88, seven models having an average SSIM between 0.86 and 0.88, and only one having an average SSIM below 0.86. All the models were able to reconstruct realistic brain images as shown in Figure 4. Many models performed better than the vanilla VAE according to both the MSE and the SSIM, but not substantially. The reconstruction metrics computed between the reconstructions obtained from simulated images and the ground truth (Table 6) show that the reconstructed images are quite similar to their healthy target (i.e., the original

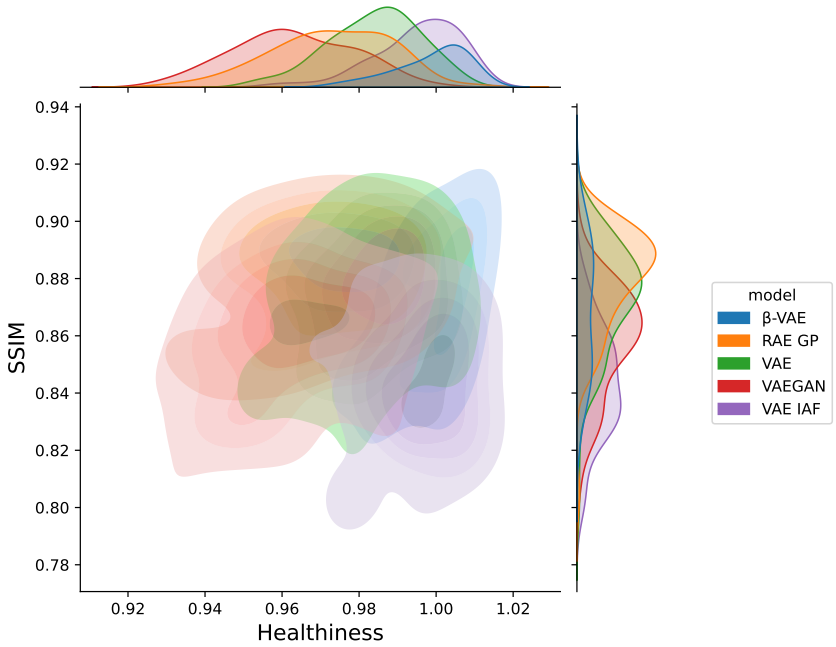


Figure 8: Joint density plot of the healthiness metric (x axis) and SSIM (y axis) computed for pseudo-healthy reconstructions obtained from images from Test AD 30. A good model should appear on the top right part of the graph (high SSIM and healthiness close to 1).

images used to simulate hypometabolic images), indicating that the reconstruction capacity of the models is not affected when using images with anomalies as input.

In terms of healthiness, on images simulating AD, it appears that the β-VAE [19], the disentangled β-VAE [14] and the VAE-IAF [21] performed better than the other VAEs, whereas the VAEGAN [22] and the RAE-GP [18] gave the worst results (Figure 5). Interestingly, the healthiness distribution of the ground truth images is multi-modal, and so is logically the distribution of simulated images. This is also the case of the healthiness distributions of the reconstruction for most of the models, especially for the FactorVAE [20] and the VQVAE [28] for which we can properly recognize the shape of the distribution. However, the reconstructions of the two best performing models, the β-VAE [19] and the disentangled β-VAE [14], have a uni-modal healthiness distribution, potentially explaining why they are not the best performing models in terms of reconstruction metrics. This may be explained by the fact that, in both cases, we set $\beta = 10$ ($\gg 1$), giving more weight to the KL term than the reconstruction term in the loss.

As illustrated in Figure 7, most of the models were able to reconstruct images of healthy appearance also for dementia subtypes different from AD. As for AD, the best performing models were the β-VAE [19] and the disentangled β-VAE [14]. The VAE LinNF [24] and the HVAE [15] also seem to have a higher healthiness than the other models. On the other hand, the VAEGAN [22], the RAE-GP [18] and the FactorVAE [20] were the models with the lowest performance. We could further see that the healthiness metric was not optimal for all the dementia subtypes. For instance, for PCA and svPPA, the healthiness of simulated images was not substantially different from that of the ground truth. Nevertheless, we observed that the healthiness of the reconstruction was close to that of the ground truth and higher than that of simulated data, which is sufficient to assess the healthiness of reconstructed images.

In general, we observed that all the models were able to reconstruct images with a healthiness substantially above the healthiness of simulated images, regardless of the kind of simulated anomalies, and almost equal to the healthiness of the ground truth, indicating that the reconstructions are indeed healthy looking.

To push further the model comparison, we jointly analyzed reconstruction and healthiness metrics (Figure 8). The RAE-GP [18] was the model with the best reconstruction, but was ranked among the worst in terms of healthiness. Even though the reconstructions look healthy when compared to the simulated input, it means that the RAE-GP [18] did not learn the healthy image distribution as well as other models, but rather learned to reconstruct the input as is. On the contrary, the VAE-IAF [21] was the model with the worst reconstruction, but was among the best in terms of healthiness. This can be explained by the presence of a reconstruction artifact that impacts the reconstruction score. The β-VAE [19] was the best model in terms of healthiness and was average in terms of reconstruction, and the VAEGAN [22]

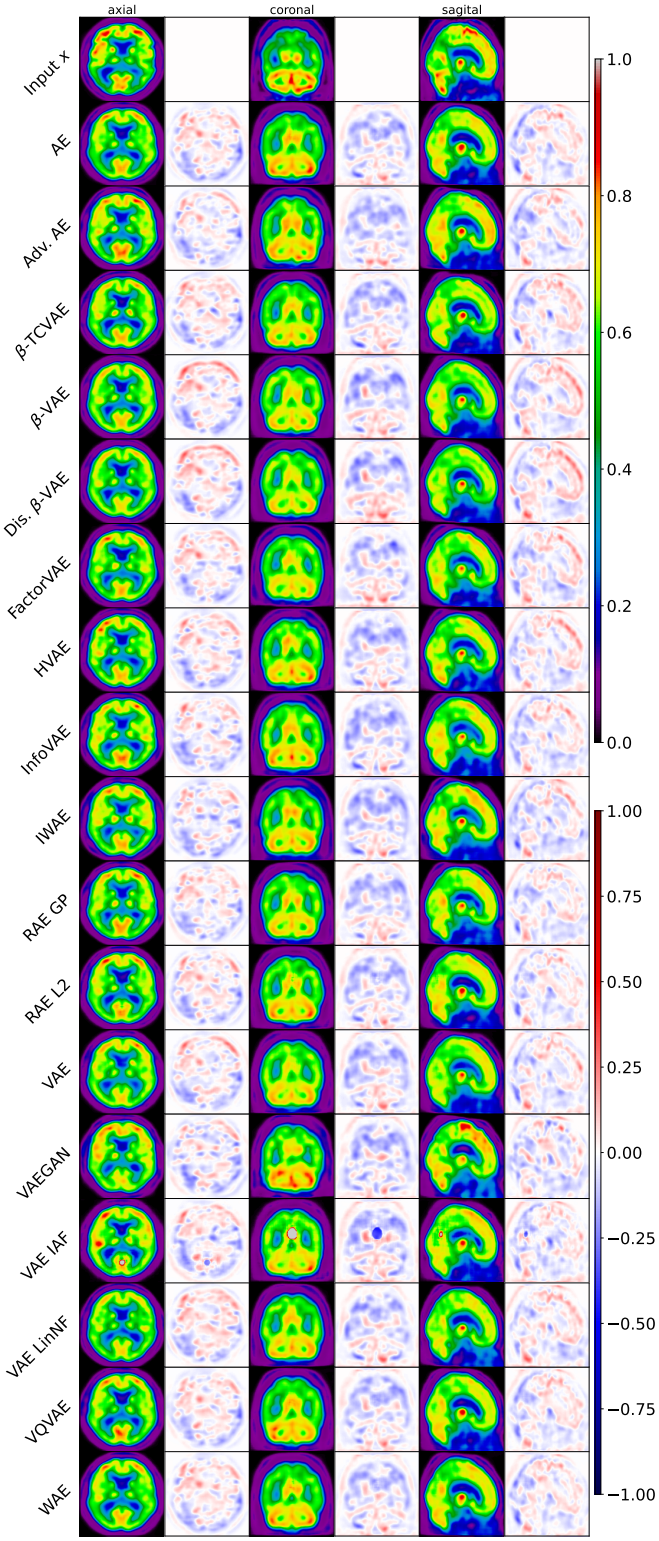


Figure 9: Example of reconstructions obtained from the different VAE variants on an AD patient (on axial, coronal and sagital slices). The first row shows the input image in odd columns. The rows below are the pseudo-healthy reconstruction of the models in odd columns, and the difference between the pseudo-healthy reconstruction and the input in even columns.

under-performed both in terms of reconstruction and healthiness.

A surprising result highlighted by the benchmark is that the simple AE performed well in comparison with more complex models, especially according to the healthiness for simulated data. Although it was expected that this model would be able to reconstruct images of healthy subjects, there was no certainty that the AE would be able to reconstruct healthy looking images from simulated images, especially when simulating severe hypometabolism (50% and more). It would be interesting to assess the performance of this model when given real images from AD patients.

In our previous study [43], we compared a subset of the models that we present here on down-sampled 3D brain FDG PET. Another major difference with the present work is that we had trained the models with default hyper-parameters' values. We observe that some of the models that performed poorly in this previous study, such as the VAEGAN [22] and the VAE LinNF [24], performed much better after searching for optimal hyper-parameters, whereas the VAMP [27] and the MS-SSIM VAE [25] still perform poorly even after hyper-parameters tuning. Even though not surprising, this highlights the benefit and need of optimizing each model, even though this step does not guarantee reaching good performance.

6.3. Limitations and perspectives

The main limitation of our work is the absence of ground truth masks for the anomalies we aim to detect. However, this benchmark proved the utility of the simulation-based evaluation framework we previously proposed [42, 54], which allowed evaluating the pseudo-healthy images reconstructed by the models using pairs of abnormal and healthy images for the same subjects, for different dementia subtypes and severity degrees. The evaluation framework also introduced the healthiness metric that automatically quantifies whether a reconstruction is pseudo-healthy. This framework is a first evaluation step that does not require the involvement of a clinician: it would indeed be impossible to ask a clinician to rate the reconstructions of 20 different models. However, a limitation is that we do not really evaluate how well each model is able to detect anomalies using these pseudo-healthy reconstructions. A solution would be to use the anomaly score proposed in [42] or abnormality maps using Z-scores [67]. A comprehensive evaluation would ultimately require using real images with real anomalies and having the results reviewed by clinicians.

The current evaluation is limited to the quality of the reconstructions and their degree of healthiness, and does not directly assess how well each model learned the healthy distribution. An interesting work would be to compare the latent distributions of the trained models to assess whether the posterior learned by the different models is the same for images from healthy and diseased subjects. This could be done using the simulation framework [42] and comparing the latent representations of both the original and simulated images. It would help us to understand the performance difference between the various VAEs, and may give us some ideas to improve them.

The models were compared on a single modality, FDG PET. It would be further interesting to test these models on structural MRI, which have different characteristics such as sharp structures. This would also allow us to compare the performance of these VAE variants with other approaches in the literature, as many have been developed to detect lesions in structural MRI. Similarly, it would be interesting to include other VAE models that performed well in computer vision such as Hierarchical VAEs [63], or compare VAEs to other generative models such as GANs and diffusion models.

6.4. Reproducibility

In order to make this study as reproducible as possible [68, 69], we tried to follow the guidelines of the MICCAI reproducibility checklist [2]:

- the publicly available dataset and final cohort we work with is mainly described in Section 3 with details of the preprocessing and data selection steps presented in Appendix A. We provide a summary of participant demographics for the train, validation and test splits in Table 1;
- the architecture choices for the VAE and the impact of those choices are detailed in Section 4.1 and Appendix B;
- the VAE variants are described in Appendix D with the range of hyper-parameters considered for each of them;

[2] https://miccai2021.org/files/downloads/MICCAI2021-Reproducibility-Checklist.pdf

- the training protocol and the method to tune and select hyper-parameters are described in Section 4.2 and Appendix D;
- we also provide a clear definition of the specific evaluation metrics and statistics used to report results in Section 2.3.

Moreover, most of the code that we used is available in ClinicaDL [60], an open-source software that is developed to enable reproducible deep learning studies in neuroimaging. Pipelines are available to perform the following steps:

- selecting subjects from a neuroimaging dataset,
- rigorously separating data into independent training and testing sets,
- rigorously splitting the training set using a cross-validation,
- launching random searches to optimize architecture and hyper-parameters,
- easily training VAE-based models on neuroimages,
- constructing new test sets by generating simulated data using the method described in [42],
- reconstructing pseudo-healthy images from trained models for the tests sets and computing the reconstruction metrics used in evaluation.

All the VAE-based models are implemented in Pythae [41], an open-source Python library that aims at unifying the implementation of VAE-based models, and facilitating benchmarks. Moreover, all the preprocessing pipelines are available in Clinica [57], an open-source software for reproducible processing of neuroimaging datasets. Clinica has been used to:

- curate and organize the ADNI dataset following a community standard, namely the brain imaging data structure (BIDS) [70],
- perform linear registration and intensity normalization of the FDG PET scans (`pet-linear` pipeline).

Finally, all the code for random searches, model training and evaluation is available in the following repository: `https://github.com/ravih18/UAD_VAE_benchmark`. This repository includes dependencies and software versions used.

7. Conclusion

In summary, we presented a benchmark of twenty VAE-based models for the unsupervised detection of dementia related anomalies in 3D brain FDG PET. The aim was to introduce the use of recent VAE variants with medical imaging data of high dimension and compare their performance. We proposed a random search method to find the optimal architecture for the vanilla VAE, as well as a random search method to tune the hyper-parameters of the implemented models.

We observed that 17 of the 20 models had a good reconstruction quality. Using our previously proposed evaluation framework [42], we showed that the 17 models were able to reconstruct pseudo-healthy images when fed with simulated abnormal images. By simulating AD with varying intensity and dementia other than AD, we also showed that these models were able to generalize to anomalies of different shapes, localizations and intensities. If no model clearly outperformed the others, the β-VAE [19] and disentangled β-VAE [14] slightly outperformed the other models, while remaining easy to tune and not being noticeably computationally costly.

Even if it is recognized that VAEs generate blurry images, all these experiments showed that most of the models were able to reconstruct good quality pseudo-healthy 3D FDG PET. The VAE variants showed similar performance and did not systematically outperform the vanilla VAE (or even the simple AE).

Finally, we can conclude that most VAEs are well suited for pseudo-healthy reconstruction of brain FDG PET images.

8. Acknowledgment

The research leading to these results has received funding from the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute) and reference ANR-10-IAIHU-06 (Agence Nationale de la Recherche-10-IA Institut Hospitalo-Universitaire-6).

This work was granted access to the HPC resources of IDRIS under the allocation AD011011648 made by GENCI (Grand Equipement National de Calcul Intensif).

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

[1] Fernando, T., Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Deep learning for medical anomaly detection – a survey. ACM Computing Surveys **54**(7) (2021)

[2] Chen, X., Konukoglu, E. In: Unsupervised abnormality detection in medical images with deep generative methods. Elsevier (2022) 303–324

[3] Baur, C., Denner, S., Wiestler, B., Navab, N., Albarqouni, S.: Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study. Medical Image Analysis **69** (2021) 101952

[4] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (ICLR). (2014)

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. Volume 27. (2014)

[6] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33** (2020) 6840–6851

[7] Graham, M.S., Pinaya, W.H.L., Wright, P., Tudosiu, P.D., Mah, Y.H., Teo, J.T., Jäger, H.R., Werring, D., Nachev, P., Ourselin, S., et al.: Unsupervised 3d out-of-distribution detection with latent diffusion models. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer (2023) 446–456

[8] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Information Processing in Medical Imaging. LNCS (2017)

[9] Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical Image Analysis **54** (2019)

[10] Xia, T., Chartsias, A., Tsaftaris, S.A.: Pseudo-healthy synthesis with pathology disentanglement and adversarial learning. Medical Image Analysis **64** (2020) 101719

[11] Shi, R., Sheng, C., Jin, S., Zhang, Q., Zhang, S., Zhang, L., Ding, C., Wang, L., Wang, L., Han, Y., et al.: Generative adversarial network constrained multiple loss autoencoder: A deep learning-based individual atrophy detection for alzheimer's disease and mild cognitive impairment. Human brain mapping **44**(3) (2023) 1129–1146

[12] Bercea, C.I., Wiestler, B., Rueckert, D., Schnabel, J.A.: Reversing the abnormal: Pseudo-healthy generative networks for anomaly detection. arXiv preprint arXiv:2303.08452 (2023)

[13] Burda, Y., Grosse, R.B., Salakhutdinov, R.: Importance weighted autoencoders. In: International Conference on Learning Representations (ICLR). (2016)

[14] Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β-vae. arXiv preprint arXiv:1804.03599 (2018)

[15] Caterini, A.L., Doucet, A., Sejdinovic, D.: Hamiltonian Variational Auto-Encoder. In: Advances in Neural Information Processing Systems. Volume 31., Curran Associates, Inc. (2018)

[16] Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. Advances in Neural Information Processing Systems **31** (2018)

[17] Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., Tomczak, J.M.: Hyperspherical variational auto-encoders. arXiv:1804.00891 (2018)

[18] Ghosh, P., Sajjadi, M.S., Vergari, A., Black, M., Schölkopf, B.: From variational to deterministic autoencoders. arXiv:1903.12436 (2019)

[19] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (ICLR). (2017)

[20] Kim, H., Mnih, A.: Disentangling by factorising. In: International conference on machine learning (ICML), PMLR (2018) 2649–2658

[21] Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. Advances in Neural Information Processing Systems **29** (2016)

[22] Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning (ICML), PMLR (2016) 1558–1566

[23] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv:1511.05644 (2015)

[24] Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International conference on machine learning (ICML), PMLR (2015) 1530–1538

[25] Snell, J., Ridgeway, K., Liao, R., Roads, B.D., Mozer, M.C., Zemel, R.S.: Learning to generate images with perceptual similarity metrics. In: ICIP, IEEE (2017) 4277–4281

[26] Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders. In: International Conference on Learning Representations (ICLR). (2018)

[27] Tomczak, J., Welling, M.: VAE with a VampPrior. In: International Conference on Artificial Intelligence and Statistics, PMLR (2018) 1214–1223

[28] Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in Neural Information Processing Systems **30** (2017)

[29] Zhao, S., Song, J., Ermon, S.: Infovae: Balancing learning and inference in variational autoencoders. In: Proc AAAI conference on artificial intelligence. Volume 33. (2019) 5885–5892

[30] Chen, X., Konukoglu, E.: Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. In: MIDL. (2018)

[31] Choi, H., Ha, S., Kang, H., Lee, H., Lee, D.S.: Deep learning only by normal brain PET identify unheralded brain anomalies. EBioMedicine **43** (2019) 447–453

[32] Mostapha, M., Prieto, J., Murphy, V., Girault, J., Foster, M., Rumple, A., Blocher, J., Lin, W., Elison, J., Gilmore, J., et al.: Semi-supervised vae-gan for out-of-sample detection applied to mri quality control. In: International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Springer (2019) 127–136

[33] Uzunova, H., Schultz, S., Handels, H., Ehrhardt, J.: Unsupervised pathology detection in medical images using conditional variational autoencoders. IJCARS **14** (2019) 451–461

[34] Zimmerer, D., Isensee, F., Petersen, J., Kohl, S., Maier-Hein, K.: Unsupervised anomaly localization using variational auto-encoders. In: International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Springer (2019) 289–297

[35] Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4, Springer (2019) 161–169

[36] Chen, X., You, S., Tezcan, K.C., Konukoglu, E.: Unsupervised lesion detection via image restoration with a normative prior. Medical image analysis **64** (2020) 101713

[37] Chételat, G., Arbizu, J., Barthel, H., Garibotto, V., Law, I., Morbelli, S., van de Giessen, E., Agosta, F., Barkhof, F., Brooks, D.J., et al.: Amyloid-PET and 18F-FDG-PET in the diagnostic investigation of Alzheimer's disease and other dementias. The Lancet Neurology **19**(11) (2020) 951–962

[38] Herholz, K.: FDG PET and differential diagnosis of dementia. Alzheimer Disease and Associated Disorders **9**(1) (1995) 6–16

[39] Jack, C.R., Bennett, D.A., Blennow, K., Carrillo, M.C., Feldman, H.H., Frisoni, G.B., Hampel, H., Jagust, W.J., Johnson, K.A., Knopman, D.S., Petersen, R.C., Scheltens, P., Sperling, R.A., Dubois, B.: A/T/N: An unbiased descriptive classification scheme for Alzheimer disease biomarkers. Neurology **87**(5) (2016) 539–547

[40] Burgos, N., Cardoso, M.J., Samper-González, J., Habert, M.O., Durrleman, S., Ourselin, S., Colliot, O.: Anomaly detection for the individual analysis of brain PET images. J Med Imag **8**(2) (2021) 024003

[41] Chadebec, C., Vincent, L.J., Allassonniere, S.: Pythae: Unifying generative autoencoders in python - a benchmarking use case. In: Thirty-sixth Conference on NeurIPS Datasets and Benchmarks Track. (2022)

[42] Hassanaly, R., Brianceau, C., Solal, M., Colliot, O., Burgos, N.: Evaluation of pseudo-healthy image reconstruction for anomaly detection with deep generative models: Application to brain FDG PET. Machine Learning for Biomedical Imaging **2** (2024) 611–656

[43] Hassanaly, R., Brianceau, C., Colliot, O., Burgos, N.: Unsupervised anomaly detection in 3D brain FDG PET: A benchmark of 17 vae-based approaches. In: Deep Generative Models workshop at the 26th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). (2023)

[44] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE T Image Process **13**(4) (2004) 600–612

[45] Nečasová, T., Burgos, N., Svoboda, D.: Validation and evaluation metrics for medical and biomedical image synthesis. In Burgos, N., Svoboda, D., eds.: Biomedical Image Synthesis and Simulation. The MICCAI Society book Series. Academic Press (2022) 573–600

[46] Xia, T., Chartsias, A., Tsaftaris, S.A.: Adversarial pseudo healthy synthesis needs pathology factorization. In: International Conference on Medical Imaging with Deep Learning, PMLR (2019) 512–526

[47] Bercea, C.I., Wiestler, B., Rueckert, D., Schnabel, J.A.: Generalizing unsupervised anomaly detection: Towards unbiased pathology screening. In: Medical Imaging with Deep Learning. (2023)

[48] Lüth, C.T., Zimmerer, D., Koehler, G., Jaeger, P.F., Isensee, F., Petersen, J., Maier-Hein, K.H.: Cradl: Contrastive representations for unsupervised anomaly detection and localization. In: Bildverarbeitung für die Medizin 2023. (2023)

[49] Wargnier-Dauchelle, V., Grenier, T., Durand-Dubief, F., Cotton, F., Sdika, M.: A weakly supervised gradient attribution constraint for interpretable classification and anomaly detection. IEEE Transactions on Medical Imaging (2023)

[50] Pinaya, W.H., Tudosiu, P.D., Gray, R., Rees, G., Nachev, P., Ourselin, S., Cardoso, M.J.: Unsupervised brain imaging 3d anomaly detection and segmentation with transformers. Medical Image Analysis **79** (2022) 102475

[51] Chatterjee, S., Sciarra, A., Dünnwald, M., Tummala, P., Agrawal, S.K., Jauhari, A., Kalra, A., Oeltze-Jafra, S., Speck, O., Nürnberger, A.: Strega: Unsupervised anomaly detection in brain mris using a compact context-encoding variational autoencoder. Computers in Biology and Medicine **149** (2022) 106093

[52] Luo, G., Xie, W., Gao, R., Zheng, T., Chen, L., Sun, H.: Unsupervised anomaly detection in brain mri: Learning abstract distribution from massive healthy brains. Computers in Biology and Medicine **154** (2023) 106610

[53] Bengs, M., Behrendt, F., Laves, M.H., Krüger, J., Opfer, R., Schlaefer, A.: Unsupervised anomaly detection in 3d brain mri using deep learning with multi-task brain age prediction. In: Medical Imaging 2022: Computer-Aided Diagnosis. Volume 12033., SPIE (2022) 291–295

[54] Hassanaly, R., Bottani, S., Sauty, B., Colliot, O., Burgos, N.: Simulation based evaluation framework for deep learning unsupervised anomaly detection on brain fdg-pet. In: Medical Imaging 2023: Image Processing. Volume 12464., SPIE (2023) 511–518

[55] Jagust, W.J., Bandy, D., Chen, K., Foster, N.L., Landau, S.M., Mathis, C.A., Price, J.C., Reiman, E.M., Skovronsky, D., Koeppe, R.A.: The Alzheimer's Disease Neuroimaging Initiative positron emission tomography core. Alzheimer's & Dementia **6**(3) (2010) 221–229

[56] Jagust, W.J., Landau, S.M., Koeppe, R.A., Reiman, E.M., Chen, K., Mathis, C.A., Price, J.C., Foster, N.L., Wang, A.Y.: The Alzheimer's Disease Neuroimaging Initiative 2 PET Core: 2015. Alzheimer's & Dementia **11**(7) (2015) 757–771

[57] Routier, A., Burgos, N., Díaz, M., Bacci, M., Bottani, S., El-Rifai, O., Fontanella, S., Gori, P., Guillon, J., Guyot, A., Hassanaly, R., Jacquemont, T., Lu, P., Marcoux, A., Moreau, T., Samper-González, J., Teichmann, M., Thibeau-Sutre, E., Vaillant, G., Wen, J., Wild, A., Habert, M.O., Durrleman, S., Colliot, O.: Clinica: An open-source software platform for reproducible clinical neuroscience studies. Front Neuroinform **15** (2021)

[58] Fonov, V., Evans, A., McKinstry, R., Almli, C., Collins, D.: Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. NeuroImage **47** (2009) S102

[59] Fonov, V., Evans, A.C., Botteron, K., Almli, C.R., McKinstry, R.C., Collins, D.L.: Unbiased average age-appropriate atlases for pediatric studies. NeuroImage **54**(1) (2011) 313–327

[60] Thibeau-Sutre, E., Díaz, M., Hassanaly, R., Routier, A., Dormont, D., Colliot, O., Burgos, N.: ClinicaDL: An open-source deep learning software for reproducible neuroimaging processing. Computer Methods and Programs in Biomedicine **220** (2022)

[61] Wen, J., Thibeau-Sutre, E., Diaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., Dormont, D., Durrleman, S., Burgos, N., Colliot, O.: Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation. Medical Image Analysis **63** (2020) 101694

[62] Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi Sepahvand, N., Raff, E., Madan, K., Voleti, V., Ebrahimi Kahou, S., Michalski, V., Arbel, T., Pal, C., Varoquaux, G., Vincent, P.: Accounting for variance in machine learning benchmarks. Proceedings of Machine Learning and Systems **3** (2021)

[63] Vahdat, A., Kautz, J.: Nvae: A deep hierarchical variational autoencoder. Advances in Neural Information Processing Systems **33** (2020) 19667–19679

[64] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

[65] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[66] Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Volume 2., Ieee (2003) 1398–1402

[67] Solal, M., Hassanaly, R., Burgos, N.: Leveraging healthy population variability in deep learning unsupervised anomaly detection in brain FDG PET. In: SPIE Medical Imaging, San Diego (California), United States (2024)

[68] Colliot, O., Thibeau-Sutre, E., Burgos, N.: Reproducibility in Machine Learning for Medical Imaging. In Colliot, O., ed.: Machine Learning for Brain Disorders. Neuromethods. Springer US, New York, NY (2023) 631–653

[69] Colliot, O., Thibeau-Sutre, E., Brianceau, C., Burgos, N.: Reproducibility in medical image computing: What is it and how is it assessed? Open Review 3fIXW9mFfn (2024)

[70] Gorgolewski, K.J., Auer, T., Calhoun, V.D., Craddock, R.C., Das, S., Duff, E.P., Flandin, G., Ghosh, S.S., Glatard, T., Halchenko, Y.O., et al.: The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Scientific data **3**(1) (2016) 1–9

[71] Neal, R.M.: Hamiltonian importance sampling. In: talk presented at the Banff International Research Station (BIRS) workshop on Mathematical Issues in Molecular Dynamics. (2005)

[72] Germain, M., Gregor, K., Murray, I., Larochelle, H.: Made: Masked autoencoder for distribution estimation. In: International conference on machine learning (ICML), PMLR (2015) 881–889

Appendix A. Data selection

The ADNI database comprises at least 3511 FDG PET scans from 1600 participants. This includes 1010 images of 554 cognitively normal (CN) subjects.To ensure that images really correspond to a healthy brain, as we know that physiological changes can appear several years before the first clinical symptoms, we kept only scans from subjects that are CN for at least three years after the session considered. We discarded 78 subjects (129 images) for whom diagnosis progresses to AD, 72 subjects (72 images) for whom there is a unique session (which is not enough to assess the reliability of the CN label) and 21 subjects (49 images) for whom there are multiple conversions or regressions. We finally kept 383 stable CN subjects (760 images).

The ADNI database also includes 560 AD patients (791 images). We removed 2 patients (2 images) with unstable AD diagnosis, 3 patients (3 images) of regressive AD, 189 patients (189 images) for whom there is a unique session, and 4 subjects that were already in the training set. We kept the 362 baseline sessions of the remaining AD patients for testing purposes and discarded all the other images.

In addition, we ran quality control and discarded in total 30 images [60]: 18 images from CN subjects and 9 images from AD patients after `t1-linear` quality control, and 3 images from CN subjects after `pet-linear` quality control. In the end, we have 378 CN subjects (739 images) and 353 AD patients (353 images).

Appendix B. Details of the encoder-decoder architecture selection procedure

In previous works [42, 54], we set the latent space size to 128 as a trade-off between performance and resources but observed that a larger latent space would lead to better reconstructions. We therefore decided to try sizes from the set $\{256, 512, 1024\}$. For choosing the number of blocks for the encoder and decoder, B_e and B_d, we initially tried the integer range from 3 to 7. This parameter influences the size of the last feature map before the fully connected layer, and therefore the number of parameters in that layer. We noticed that having an encoder with 3 blocks leads to a very large number of parameters in the fully connected layer (around 750,000 if $C_e = 16$ or double if we double C_e), whereas an encoder with 7 blocks would lead to very small feature maps ($1 \times 1 \times 1$). We therefore reduced the range, and chose values between 4 and 6. We kept the number of sub-blocks in the encoder and decoder S_e and S_d relatively small to restrain the number of parameters of our model whilst still testing deep architectures. We chose the number of channels C_e and C_d based on previous experiments and decided to set it to either 16 or 32. We also added the possibility to add a convolution layer in our last decoder block (shown by dotted lines in Figure 2). We also included the learning rate and the optimizer as parameters in our random search. We first performed experiments where the learning rate was chosen from $\{10^{-5}, 10^{-4}, 10^{-3}\}$, but setting it to 10^{-3} led to errors in the computation of the loss so we pursued our search with only 10^{-5} and 10^{-4} as options. The optimizer could be either Adam or Adamax, following the suggestions from [63]. The parameters included in our random search are summarized in Table 2.

After attempting to train 200 models, a pattern emerged and we could select and test an additional architecture following our intuition. These results are summarized in Table B.7 and Table B.8. Certain parameters, such as the learning rate, the latent space size, and the number of channels C_e and C_d were easy to choose as a clear relation with the reconstruction metrics could be observed, allowing us to choose to set the learning rate to 10^{-4}, the latent space size to 256 and $D_e = C_d = 16$. We particularly struggled to train models with residual sub-blocks due to memory constraints, and those that were able to train did not give very good results so we focused our efforts on models with convolutional sub-blocks. The optimizer did not seem to make a large difference so we chose Adam. For the remaining parameters, we observed that there was no need for a very deep architecture (or large number of sub-blocks within blocks), whereas a large number of blocks was beneficial in terms of memory as it induced a smaller number of parameters before the fully connected layer. After analyzing these results and noticing these patterns, we decided to test an extra model which we designed to be symmetrical (for the sake of simplicity) and as light as possible in terms of memory (128 MB instead of 286 MB), as we knew that some of the models that we would train later with this architecture are much more memory greedy. We selected random splits by drawing 3 cards from a deck of 6 cards to train our model and found that this model performed similarly to the best performing models from our random search (equivalent SSIM and best MSE). We therefore decided to select this architecture as it was simpler (because symmetrical) and smaller in terms of memory and number of parameters.

Table B.7: Results of the random search on the VAE architecture: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected). The selected configuration is highlighted in gray.

Id	block type	C_e	B_e	S_e	latent size	B_d	S_d	C_d	last conv	learning rate	SSIM ↑	MSE ↓
1	conv	16	5	2	256	5	2	16	True	0.0001	0.866 ± 0.006	2.158 ± 0.043
2	conv	16	5	2	256	4	3	32	False	0.0001	0.864 ± 0.008	2.221 ± 0.111
3	conv	16	5	3	1024	4	3	16	False	0.0001	0.861 ± 0.017	2.202 ± 0.082
4	conv	32	5	1	256	5	1	32	False	0.00001	0.861 ± 0.004	2.081 ± 0.021
5	conv	32	5	1	512	5	2	32	False	0.0001	0.857 ± 0.001	2.153 ± 0.056
6	conv	16	5	1	256	5	1	16	False	0.0001	0.856 ± 0.004	1.919 ± 0.088
7	conv	32	5	3	512	5	1	16	True	0.0001	0.852 ± 0.002	2.138 ± 0.101
8	conv	32	5	1	512	4	1	32	True	0.0001	0.851 ± 0.006	2.303 ± 0.187
9	conv	16	5	2	1024	4	1	16	True	0.0001	0.850 ± 0.007	2.572 ± 0.145
10	conv	16	5	2	1024	5	1	16	True	0.0001	0.840 ± 0.024	2.014 ± 0.083

Table B.8: Results of the random search on the VAE architecture: ranking according to the MSE of the 10 best configurations (mean ± std over the three folds randomly selected). The selected configuration is highlighted in gray. The Id corresponds to the rank of the model when sorting them according to the SSIM (Table B.7).

Id	block type	C_e	B_e	S_e	latent size	B_d	S_d	C_d	last conv	learning rate	SSIM ↑	MSE ↓
6	conv	16	5	1	256	5	1	16	False	0.0001	0.856 ± 0.004	1.919 ± 0.088
10	conv	16	5	2	1024	5	1	16	True	0.0001	0.840 ± 0.024	2.014 ± 0.083
14	conv	32	5	1	1024	4	3	16	False	0.00001	0.826 ± 0.013	2.062 ± 0.065
4	conv	32	5	1	256	5	1	32	False	0.00001	0.861 ± 0.004	2.081 ± 0.021
11	conv	16	5	1	256	4	1	32	True	0.0001	0.800 ± 0.049	2.081 ± 0.107
13	conv	32	6	1	256	6	2	32	False	0.00001	0.832 ± 0.009	2.091 ± 0.113
12	conv	16	5	3	512	4	2	16	True	0.0001	0.818 ± 0.032	2.111 ± 0.069
7	conv	32	5	3	512	5	1	16	True	0.0001	0.853 ± 0.002	2.128 ± 0.101
5	conv	32	5	1	512	5	2	32	False	0.0001	0.857 ± 0.001	2.153 ± 0.056
1	conv	16	5	2	256	5	2	16	True	0.0001	0.866 ± 0.006	2.158 ± 0.043

Appendix C. Detailed description of the final encoder-decoder architecture

Table C.9: Summary of the layer parameters in the final VAE architecture

	Layer	Input shape	Output shape	Kernel	Stride	Padding	BN	Activation
Encoder	Conv3D	(1, 169, 208, 179)	(16, 84, 104, 89)	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)	True	SiLU
	Conv3D	(16, 84, 104, 89)	(32, 42, 52, 44)	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)	True	SiLU
	Conv3D	(32, 42, 52, 44)	(64, 21, 26, 22)	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)	True	SiLU
	Conv3D	(64, 21, 26, 22)	(128, 10, 13, 11)	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)	True	SiLU
	Conv3D	(128, 10, 13, 11)	(256, 5, 6, 5)	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)	True	SiLU
	Flatten	(256, 5, 6, 5)	(38400)	-	-	-	-	-
	FC × 2	(38400)	(256) × 2	-	-	-	False	SiLU
Decoder	FC	(256)	(38400)	-	-	-	False	ReLU
	Unflatten	(38400)	(256, 5, 6, 5)	-	-	-	-	-
	Upsample	(256, 5, 6, 5)	(256, 10, 13, 11)	-	-	-	-	-
	Conv3D	(256, 10, 13, 11)	(128, 10, 13, 11)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	True	SiLU
	Upsample	(128, 10, 13, 11)	(128, 21, 26, 22)	-	-	-	-	-
	Conv3D	(128, 21, 26, 22)	(64, 21, 26, 22)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	True	SiLU
	Upsample	(64, 21, 26, 22)	(64, 42, 52, 44)	-	-	-	-	-
	Conv3D	(64, 42, 52, 44)	(32, 42, 52, 44)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	True	SiLU
	Upsample	(32, 42, 52, 44)	(32, 84, 104, 89)	-	-	-	-	-
	Conv3D	(32, 84, 104, 89)	(16, 84, 104, 89)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	True	SiLU
	Upsample	(16, 84, 104, 89)	(16, 169, 208, 179)	-	-	-	-	-
	Conv3D	(16, 169, 208, 179)	(1, 169, 208, 179)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	False	Sigmoid

BN: batch normalization, FC: fully connected, SiLU: swish activation function

Appendix D. Description of the VAE variants and of their hyper-parameter selection procedure

This section describes all the VAE variants. Hyper-parameters were chosen following implementations and recommendations from the original papers and the benchmark previously done by Chadebec et al. [41]. The results of the random searches are reported for each of the models. For each model, we selected the configuration with the best average SSIM on the validation folds.

Appendix D.1. Adversarial Autoencoder

The adversarial autoencoder [23] is a probabilistic autoencoder model that uses the GAN framework to perform variational inference in the latent space. It uses a discriminator network to differentiate a prior's sample from a posterior's sample as a form of regularization. Its objective and training are quite similar to that of a VAE

$$\mathcal{L}_{\text{Adv. AE}} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \alpha\mathcal{L}_{\text{GAN}} \ ,$$

where

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{\tilde{z}\sim p_z(z)}\left[\log(1 - D(\tilde{z})\right] + \mathbb{E}_{x\sim p_\theta}\left[\mathbb{E}_{z\sim q_\phi(z|x)}[\log D(z)]\right] \ .$$

We set the discriminator to be the same as in [41], that is, a multilayer perceptron with a single hidden layer with 256 units and ReLU activation. We performed a grid search of 10 configurations for

$$\alpha \in \{0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99\} \ .$$

The results are reported in Table D.10.

Table D.10: Results of the random search on the hyper-parameters of the Adv. AE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

adversarial loss scale	SSIM ↑	MSE ($\times 10^{-3}$) ↓
0.9	0.873 ± 0.005	1.770 ± 0.083
0.01	0.872 ± 0.003	1.771 ± 0.144
0.1	0.869 ± 0.005	1.846 ± 0.115
0.5	0.869 ± 0.015	1.811 ± 0.094
0.75	0.869 ± 0.006	1.784 ± 0.142
0.25	0.866 ± 0.002	1.841 ± 0.155
0.99	0.865 ± 0.014	1.863 ± 0.094
0.05	0.863 ± 0.009	1.779 ± 0.103
0.001	0.863 ± 0.007	1.860 ± 0.075
0.95	0.856 ± 0.001	1.814 ± 0.118

Appendix D.2. β-TC VAE

The β-TCVAE, or Total Correlation VAE [16], is an extension of the β-VAE [19], which aims at further isolating sources of disentanglement by rewriting the ELBO in the following way:

$$\mathcal{L}_{\beta\text{-TCVAE}} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathcal{L}_{\text{reg}} \quad ,$$

where

$$\mathcal{L}_{\text{reg}} = \alpha \mathcal{D}_{\text{KL}}\left[q_\phi(z,x)||q_\phi(z)p_\theta(x)\right] + \beta \mathcal{D}_{\text{KL}}\left[q_\phi(z)||\prod_j q_\phi(z_j)\right] + \gamma \sum_j \mathcal{D}_{\text{KL}}\left[q_\phi(z_j)||p_z(z_j)\right] \quad .$$

The regularization term is therefore the sum of the mutual information between x and z, the total correlation, which models the dependence between dimensions of the latent vector, and the dimension-wise KL divergence, which prevents each dimension of the latent variable from diverging too far from its prior.

Following the authors' suggestion, we set $\alpha = \gamma = 1$ for most of the models and performed a grid search for parameter $\beta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 1, 2, 5, 10\}$. We also tried the configurations $(\beta, \alpha, \gamma) = (1, 1, 3)$ and $(\beta, \alpha, \gamma) = (1, 3, 1)$, which made a total of 12 configurations. The results are reported in Table D.11.

Table D.11: Results of the random search on the hyper-parameters of the β-TC VAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

β	α	γ	SSIM ↑	MSE ($\times 10^{-3}$) ↓
2	1	1	0.870 ± 0.002	1.901 ± 0.123
0.05	1	1	0.866 ± 0.002	1.953 ± 0.082
1	1	3	0.866 ± 0.003	1.993 ± 0.077
5	1	1	0.864 ± 0.004	1.923 ± 0.072
0.005	1	1	0.864 ± 0.009	1.871 ± 0.113
0.001	1	1	0.863 ± 0.005	1.903 ± 0.138
1	3	1	0.862 ± 0.010	1.810 ± 0.034
10	1	1	0.862 ± 0.008	1.969 ± 0.095
0.01	1	1	0.860 ± 0.010	1.917 ± 0.096
0.1	1	1	0.855 ± 0.010	1.864 ± 0.100

Appendix D.3. β-VAE

The β-VAE [19] was introduced to encourage the disentanglement of features in the latent space by adding a weight β in front of the KL term to adjust the balance between reconstruction and regularization. The objective is:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \beta \mathcal{D}_{\text{KL}}\left[q_\phi(z|x)||p_z(z)\right] \quad ,$$

where setting $\beta > 1$ leads to stronger disentanglement whereas using a smaller β can favor better reconstruction abilities.

We performed a grid search of 10 configurations for $\beta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 2, 5, 10, 100\}$. The results are reported in Table D.12.

Table D.12: Results of the random search on the hyper-parameters of the β-VAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

β	SSIM ↑	MSE ($\times 10^{-3}$) ↓
10	0.868 ± 0.003	1.995 ± 0.067
0.005	0.868 ± 0.006	1.785 ± 0.142
0.01	0.867 ± 0.006	1.755 ± 0.122
0.001	0.866 ± 0.005	1.825 ± 0.072
0.05	0.866 ± 0.008	1.859 ± 0.090
2	0.863 ± 0.009	1.9 ± 0.072
0.1	0.863 ± 0.011	1.894 ± 0.103
0.5	0.858 ± 0.011	1.835 ± 0.117
5	0.856 ± 0.011	1.969 ± 0.095
100	0.816 ± 0.008	3.716 ± 0.292

Appendix D.4. Disentangled β-VAE

The disentangled β-VAE[14] introduces a way to progressively increase the latent encoding capacity to improve the reconstruction accuracy in comparison with the β-VAE [19]. The objective becomes

$$\mathcal{L}_{\text{disentangled } \beta\text{-VAE}} = \mathcal{L}_{rec} - \beta|\mathcal{D}_{KL}(q_\phi(z|x)||p(z)) - C| \ ,$$

with C the value of the KL divergence term we would like to approach.

We performed a random search on the three parameters: $\beta \in \{10^{-2}, 10^{-1}, 1, 5, 10\}$, $C \in \{5, 25, 50\}$ and the number of epochs (warm-up epochs) during which the KL divergence in the ELBO will increase from 0 to C, which can be 100 or 1000. We trained a total of 20 configurations (out of 60 possible combinations), and the results of the random search are given in the Table D.13.

Table D.13: Results of the random search on the hyper-parameters of the Dis. β-VAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

β	C	warm-up epoch	SSIM ↑	MSE ($\times 10^{-3}$) ↓
10	50	1000	0.874 ± 0.006	2.004 ± 0.153
0.1	25	1000	0.873 ± 0.002	1.821 ± 0.056
1	50	100	0.873 ± 0.007	1.852 ± 0.092
0.01	5	1000	0.871 ± 0.004	1.755 ± 0.055
0.1	50	100	0.871 ± 0.009	1.869 ± 0.073
0.01	25	1000	0.870 ± 0.003	1.753 ± 0.074
10	5	1000	0.870 ± 0.003	2.053 ± 0.110
0.1	5	1000	0.869 ± 0.014	1.815 ± 0.036
1	5	100	0.869 ± 0.008	1.879 ± 0.064
5	25	1000	0.867 ± 0.002	2.009 ± 0.068

Appendix D.5. Factor VAE

Kim et al [20] proposed a new metric for disentanglement that encourages the latent representation to be factorial, and independent across each dimension of the latent space. The loss function is the following:

$$\mathcal{L}_{FactorVAE} = \mathcal{L}_{VAE} - \gamma\mathcal{D}_{KL}(q_\phi(z)||\bar{q_\phi}(z))) \ ,$$

with $\bar{q_\phi}(z) := \prod_{j=1}^{d} q_\phi(z_j)$ for a model with a latent space of dimension d.

We performed a grid search of 10 configurations to find the optimal $\gamma \in \{2, 5, 10, 15, 20, 30, 40, 50, 100, 200\}$. The results are reported in Table D.14.

Table D.14: Results of the random search on the hyper-parameters of the FactorVAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

γ	SSIM ↑	MSE ($\times 10^{-3}$) ↓
40	0.876 ± 0.003	1.895 ± 0.084
100	0.875 ± 0.007	1.827 ± 0.092
15	0.874 ± 0.004	1.872 ± 0.048
20	0.869 ± 0.010	1.875 ± 0.090
50	0.866 ± 0.011	1.850 ± 0.070
200	0.864 ± 0.008	1.820 ± 0.032
10	0.864 ± 0.011	1.859 ± 0.086
30	0.864 ± 0.020	1.805 ± 0.096
5	0.862 ± 0.019	1.890 ± 0.075
2	0.852 ± 0.016	1.901 ± 0.081

Appendix D.6. Hamiltonian VAE

Caterini et al. introduced a new method to obtain a low variance unbiased estimation of the ELBO using Markov chain Monte Carlo with Hamiltonian importance sampling [71] and by proposing a method to select optimal reverse kernels, building the Hamiltonian VAE [15] with the following loss:

$$\mathcal{L}_{HVAE} = \mathbb{E}_{z_0 \sim q^0_{\theta,\phi}(.,.)} \left[\log p_\theta(x, z_K) - \frac{1}{2} \rho_K^T \rho_K - \log q^0_{\theta,\phi}(z_0) \right] + \frac{l}{2}$$

where $(z_0, \rho_0) = \mathcal{H}_{\theta,\phi}(z_0, \gamma_0/\sqrt{\beta_0})$, $\mathcal{H}$ is the Hamiltonian importance sampling [71], β_0 is the inverse temperature and $\gamma_0 \sim \mathcal{N}(\cdot|0, I)$.

There are three hyper-parameters that we randomly searched for: the number of step in the leapfrog $n_{lf} \in \{1, 2, 10, 15, 20\}$, the leapfrog step size $\epsilon_{lf} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ and $\beta_0 \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ the tempering factor in the Hamiltonian Monte Carlo Sampler. We trained 20 configurations out of 220 possible combinations. The results are reported in Table D.15. Note that some configurations were really long to train, sometimes exceeding the time limit of the HPC used to train the models.

Table D.15: Results of the random search on the hyper-parameters of the HVAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

n_{lf}	ϵ_{lf}	β_0	SSIM ↑	MSE ($\times 10^{-3}$) ↓
10	0.00001	0.8	0.873 ± 0.007	1.862 ± 0.068
2	0.00001	0.7	0.870 ± 0.002	1.905 ± 0.079
2	0.001	0.2	0.870 ± 0.004	1.847 ± 0.082
1	0.001	0.5	0.869 ± 0.008	1.853 ± 0.101
15	0.00001	0.2	0.868 ± 0.009	1.854 ± 0.075
1	0.001	0.7	0.865 ± 0.004	1.890 ± 0.102
2	0.01	1	0.865 ± 0.005	1.911 ± 0.066
15	0.001	0.4	0.865 ± 0.008	1.805 ± 0.024
15	0.001	0.1	0.864 ± 0.009	1.908 ± 0.084
10	0.0001	0.9	0.863 ± 0.003	1.882 ± 0.104

Appendix D.7. Info VAE MMD

To improve both the generative model and the amortized inference distribution, Zhao et al. [29] proposed to add the mutual information between z and x in the objective function of the VAE. To be optimized, the loss is rewritten as follows:

$$\mathcal{L}_{InfoVAE} = \mathbb{E}_{p\mathcal{D}(x)} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] - (1-\alpha) \mathbb{E}_{p\mathcal{D}(x)} \mathcal{D}_{KL}(q_\phi(z|x)||p(z)) - (\alpha + \lambda - 1) \mathcal{D}(q_\phi(z)||p(z))$$

with $\mathcal{D}$ the maximum mean discrepancy (MMD).

We performed a random search of the following parameters: $\alpha \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, $\lambda \in \{0.01, 0.1, 1, 10, 100\}$, the choice of the kernel for the MMD $\in \{\text{rbf}, \text{imq}\}$ (rbf: radial basis function, imq: inverse multi-quadratic) and the kernel bandwidth $\in \{0.01, 0.1, 0.5, 1, 5, 10, 100\}$. We trained 30 configurations out of 420 possible combinations. The results are reported in Table D.16.

Table D.16: Results of the random search on the hyper-parameters of the InfoVAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

kernel choice	α	λ	kernel bandwidth	SSIM ↑	MSE ($\times 10^{-3}$) ↓
rbf	1	0.1	0.1	0.877 ± 0.006	1.813 ± 0.075
rbf	0.4	1	0.5	0.875 ± 0.006	1.804 ± 0.052
rbf	1	0.1	0.5	0.874 ± 0.003	1.770 ± 0.077
rbf	0.00001	100	1	0.873 ± 0.002	1.852 ± 0.095
rbf	0.00001	10	0.5	0.873 ± 0.004	1.846 ± 0.094
imq	0.4	100	0.1	0.872 ± 0.008	1.866 ± 0.045
imq	1	10	1	0.872 ± 0.006	1.830 ± 0.068
rbf	0.6	0.01	5	0.871 ± 0.007	1.832 ± 0.088
rbf	1	100	0.01	0.870 ± 0.004	1.768 ± 0.079
imq	0.2	0.01	0.01	0.870 ± 0.005	1.830 ± 0.107

Appendix D.8. IWAE

Instead of relying on a single sample for estimating the posterior, the IWAE [13] utilizes importance weights during the sampling process in the latent space on multiple samples (Monte Carlo estimator), assigning higher weights to more probable samples. This provides a new ELBO that becomes tighter when the number of samples increases. The loss is the following:

$$\mathcal{L}_{IWAE} = \mathbb{E}_{z_1,\dots,z_k \sim q_\phi(z|x)} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p_\theta(x, z_i)}{q_\phi(z_i|x)} \right]$$

with $k \in \{2, 3, 4, 5, 6, 8, 10, 12, 15, 20\}$ the number of samples to use in the Monte Carlo estimator.

When k grows, the IWAE becomes very memory greedy and time consuming during training, especially with 3D images. We had to reduce the batch size to 2, and, in spite of this, the model would crash because of memory when setting $k > 6$. The results are reported in Table D.16.

Table D.17: Results of the random search on the hyper-parameters of the IWAE: ranking according to the SSIM of the 3 best configurations (mean ± std over the three folds randomly selected).

number of samples	SSIM ↑	MSE ($\times 10^{-3}$) ↓
6	0.865 ± 0.007	2.087 ± 0.146
3	0.861 ± 0.002	2.048 ± 0.064
4	0.854 ± 0.013	2.178 ± 0.201

Appendix D.9. MS-SSIM VAE

Snell et al. [25] proposed an extension of the VAE, called the expected loss VAE, where the pixel-wise reconstruction loss can be replaced by any deterministic reconstruction loss. For this, the probabilistic decoder p_θ is replaced by a deterministic equivalent f_θ so that the reconstruction $\hat{x}$ of x given $z \sim q_\phi(z|x)$ is given by $\hat{x} = f_\theta(x)$ and the reconstruction loss is given by $\Delta(x, \hat{x})$. The objective becomes

$$\mathcal{L}_{\text{EL-VAE}} = \mathbb{E}_{q_\phi(z|x)} \left[\Delta(x, \hat{x}) \right] - \beta \mathcal{D}_{\text{KL}} \left[q_\phi(z|x) || p_z(z) \right].$$

Following the authors' suggestion we use the MS-SSIM, or multi-scale structural similarity, as our reconstruction loss.

We performed a random search on β and the window size used in the computations of the MS-SSIM, where β is sampled from $\{0.01, 0.1, 1, 10, 100\}$ and the window size is sampled from $\{2, 3, 5, 11\}$. We trained 10 configurations out of 20 possible combinations. The results are reported in Table D.18.

The training time was too long for configurations with a window size different from 2, explaining why Table D.18 contains only five configuration with a window size of 2.

Table D.18: Results of the random search on the hyper-parameters of the MS-SSIM VAE: ranking according to the SSIM of the 5 best configurations (mean ± std over the three folds randomly selected).

β	window size	SSIM ↑	MSE ($\times 10^{-3}$) ↓
100	2	0.472 ± 0.034	70.174 ± 5.660
1	2	0.453 ± 0.050	75.443 ± 11.098
1	2	0.448 ± 0.018	74.110 ± 1.158
1	2	0.445 ± 0.050	76.354 ± 8.802
0.01	2	0.393 ± 0.039	84.579 ± 7.183

Appendix D.10. Regularized auto-encoder

Ghosh et al. [18] claimed that the probabilistic sampling in VAE is equivalent to a noise injection to the decoder, acting as a stochastic regularization of the latent space. The authors proposed a new approach that consists in replacing the random noise injection by a deterministic regularization in the decoder. The training objective becomes

$$\mathcal{L}_{RAE} = \|x - \hat{x}\|_2^2 + \beta.\mathcal{L}_Z^{RAE} + \lambda.\mathcal{L}_{REG} \ ,$$

with $\mathcal{L}_{REG}$ the regularization term for the decoder and $\mathcal{L}_Z^{RAE} = 1/2\|z\|_2^2$ a constraint on the latent space. The authors suggested two different regularizations for the decoder:

- the first option is a $\mathcal{L}_2$ norm on the weights of the decoder $\mathcal{L}_{REG} = \|\theta\|_2^2$, giving the RAE-$\ell^2$ model;
- another choice is to apply a gradient penalty on the discriminator $\mathcal{L}_{REG} = \|\nabla D_\theta(E_\phi(x))\|_2^2$, giving the RAE-GP model.

We performed a random search on both λ and β, that are both sampled from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. We trained 20 configurations for both the RAE-ℓ^2 and the RAE-GP out of 36 possible combinations for each model. The results are respectively reported in Tables D.19 and D.20.

Table D.19: Results of the random search on the hyper-parameters of the RAE-ℓ^2: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

embedding weight	reg weight	SSIM ↑	MSE ($\times 10^{-3}$) ↓
0.0001	1	0.884 ± 0.005	1.815 ± 0.049
0.0001	0.001	0.883 ± 0.002	1.765 ± 0.070
0.0001	1	0.879 ± 0.008	1.848 ± 0.059
0.0001	0.01	0.879 ± 0.009	1.857 ± 0.064
0.00001	0.01	0.879 ± 0.007	1.868 ± 0.055
0.1	0.001	0.878 ± 0.007	1.814 ± 0.052
0.00001	0.01	0.878 ± 0.006	1.785 ± 0.076
0.1	0.0001	0.878 ± 0.007	1.853 ± 0.077
0.00001	0.1	0.878 ± 0.007	1.783 ± 0.107
0.00001	0.01	0.877 ± 0.005	1.831 ± 0.121

Table D.20: Results of the random search on the hyper-parameters of the RAE-GP: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

embedding weight	reg weight	SSIM ↑	MSE ($\times 10^{-3}$) ↓
0.01	0.0001	0.880 ± 0.006	1.715 ± 0.105
0.0001	0.0001	0.877 ± 0.008	1.744 ± 0.056
0.1	0.00001	0.877 ± 0.009	1.820 ± 0.093
1	0.001	0.867 ± 0.003	1.756 ± 0.063
0.1	0.01	0.861 ± 0.011	1.828 ± 0.031
0.1	0.1	0.845 ± 0.012	1.750 ± 0.107
0.0001	0.1	0.842 ± 0.010	1.769 ± 0.079
0.1	0.1	0.839 ± 0.008	1.799 ± 0.101
0.00001	1	0.825 ± 0.013	1.906 ± 0.135
0.1	1	0.808 ± 0.004	1.924 ± 0.145

Appendix D.11. Hyperspherical VAE

The hyperspherical VAE [17] uses a von Mises-Fisher (vMF) distribution as prior leading to a hyperspherical latent space. This model has the advantage of not having additional hyper-parameters compared to a standard VAE but only works with a small latent space as large values lead to errors when computing the modified Bessel function involved in the probability density function of the vMF distribution. Therefore, we performed a grid search on three different smaller latent space sizes: $8, 16, 32$. The results are reported in Table D.21.

Table D.21: Results of the random search on the hyper-parameters of the SVAE: ranking according to the SSIM of the 3 best configurations (mean ± std over the three folds randomly selected).

latent space size	SSIM ↑	MSE ($\times 10^{-3}$) ↓
16	0.151 ± 0.001	632.694 ± 5.106
32	0.150 ± 0.002	640.791 ± 4.328
8	0.083 ± 0.028	189.998 ± 68.394

Appendix D.12. VAE-GAN

In the VAE-GAN [22], a discriminator is trained on the output of a VAE to enhance the VAE's reconstruction abilities. The idea is to use the learned feature representations from intermediate layers of the GAN discriminator as a basis for the VAE reconstruction objective, assuming that the discriminator can capture high-level structures relevant to the data distribution. Overall, this allows replacing voxel-wise similarity between input and output by feature-wise similarity. For $z \sim p_z(z)$ and $\hat{x} \sim D_\theta(z)$, the objective is given by

$$\mathcal{L}_{\text{VAE-GAN}} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log \mathcal{N}(D_l(x)|D_l(\hat{x}), \mathbf{I})\right] - \mathcal{D}_{\text{KL}}\left[q_\phi(z|x)||p_z(z)\right] - \log\left(\frac{D(x)}{1 - D(D_\theta(z))}\right) \ ,$$

where D denotes the discriminator, D_l the hidden representation of the l-th layer of the discriminator, and D_θ the decoder. We also added a hyper-parameter α to the decoder's loss, such that high values of α encourage better reconstruction with respect to the features learned at the layer l of the discriminator.

We set the discriminator to be a neural network with 4 convolutions and 2 fully connected layers, with batch normalization and ReLU activation. We set the margin to 0.4 and the equilibrium to 0.68 as in the original paper. We performed a random search for $\alpha \in \{0.3, 0.5, 0.7, 0.9\}$ and $l \in \{1, 2, 3, 4\}$. This model is particularly long to train and, due to memory constraints, we reduced the batch size to 4 instead of 8 for these models. We trained 10 different configurations out of 16 possible combinations. The results are reported in Table D.22. We note that there is a strong correlation between the chosen reconstruction layer in the decoder and the quality of the reconstruction.

Table D.22: Results of the random search on the hyper-parameters of the VAEGAN: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

adversarial loss scale	reconstruction layer	SSIM ↑	MSE ($\times 10^{-3}$) ↓
0.5	1	0.860 ± 0.013	2.241 ± 0.193
0.1	1	0.851 ± 0.015	2.671 ± 0.090
0.1	1	0.849 ± 0.006	2.547 ± 0.356
0.3	2	0.799 ± 0.050	3.705 ± 0.559
0.3	2	0.780 ± 0.101	3.968 ± 0.642
0.9	3	0.714 ± 0.148	9.727 ± 4.423
0.7	3	0.692 ± 0.101	9.832 ± 1.185
0.8	3	0.572 ± 0.061	10.336 ± 5.712
0.9	4	0.560 ± 0.113	24.463 ± 6.516

Appendix D.13. VAE with inverse auto-regressive flows

The VAE with inverse auto-regressive flows [21] incorporates a series of inverse auto-regressive flows in the encoder, enhancing the flexibility of the learned posterior distribution, and scaling well to high-dimensional latent spaces. We use masked autoencoder for distribution estimation (MADE) [72] as normalizing flow, as suggested in [21] and implemented in Pythae [41].

We performed a random search on the following parameters: the number of MADE blocks $\in \{2, 3, 4, 5, 6, 8\}$, the number of hidden layers in the MADE blocks $\in \{2, 3, 4, 5\}$ and the size of the hidden layers $\in \{64, 128, 256\}$. We trained 30 different configurations out of 72 possible combinations. However, we noticed that the performance was really poor when the number of MADE blocks was odd, reducing the possible values for this parameter to even numbers. The results are reported in Table D.23.

Table D.23: Results of the random search on the hyper-parameters of the VAE-IAF: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

n MADE blocks	n hidden in MADE	hidden size	SSIM ↑	MSE ($\times 10^{-3}$) ↓
4	4	128	0.823 ± 0.005	2.272 ± 0.057
4	5	256	0.823 ± 0.008	2.248 ± 0.063
2	5	256	0.823 ± 0.007	2.220 ± 0.071
8	4	128	0.822 ± 0.005	2.282 ± 0.092
6	5	128	0.820 ± 0.004	2.259 ± 0.148
6	5	64	0.820 ± 0.003	2.403 ± 0.107
4	3	128	0.819 ± 0.008	2.260 ± 0.055
4	2	64	0.818 ± 0.015	2.331 ± 0.133
4	5	64	0.817 ± 0.006	2.359 ± 0.114
8	5	64	0.816 ± 0.005	2.546 ± 0.028

Appendix D.14. VAE with linear normalizing flows

The VAE with linear normalizing flows [24] enables a better approximation of the posterior distribution $q_\phi(z|x)$ using a series of linear normalizing flows, which are invertible transformations. To get the latent vector z_K, z_0 is sampled from $q_\phi(z|x)$ and passes through K linear normalizing flows f_k such that $z_K = f_K \circ ... \circ f_2 \circ f_1(z_0)$. These flows enable the model to capture complex distributions in the latent space. The authors suggest to use a succession of linear flows, and more precisely planar or radial flows, because it is computationally efficient to compute their Jacobian, as needed to compute the loss

$$\mathcal{L}_{VAElinNF} = \mathcal{L}_{rec} + \log q_\phi(z_0) - \log q(z_K) - \sum_{k=1}^{K} \log |\det \frac{\partial f_k}{\partial z}| \ .$$

We tried 10 different configurations of flows $\in \{10P, 10R, 5P, 5R, 5P5R, 5R5P, 5PR, 5RP, 2PR, 2RP\}$, with P designating a planar flow and R a radial flow. The results are reported in Table D.24. We note that configurations with radial flows clearly outperform configurations with planar flows.

Table D.24: Results of the random search on the hyper-parameters of the VAE LinNF: ranking according to the SSIM of the nine best configurations (mean ± std over the three folds randomly selected). P designate a planar flow, R designate a radial flow.

flows	SSIM ↑	MSE ($\times 10^{-3}$) ↓
10R	0.871 ± 0.001	1.855 ± 0.125
5R	0.860 ± 0.008	1.897 ± 0.066
2PR	0.827 ± 0.005	2.262 ± 0.135
5PR	0.737 ± 0.058	3.218 ± 0.205
5P5R	0.720 ± 0.095	3.148 ± 0.763
5RP	0.716 ± 0.112	3.829 ± 0.942
5R5P	0.708 ± 0.080	4.652 ± 1.518
5P	0.679 ± 0.098	4.897 ± 1.878
10P	0.570 ± 0.064	8.082 ± 4.095

Appendix D.15. VAE with VampPrior

The VAE with a "Variational Mixture of Posteriors" prior, or VampPrior [27], aims to replace the simple normal prior with a mixture of distributions (e.g. mixture of Gaussians), allowing capturing more complex data distributions. We optimize the following loss:

$$\mathcal{L}_{VAMP} = \mathcal{L}_{rec} - (\log p_\lambda(z) - \log q_\phi(z|x))$$

with $p_\lambda(z) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(z|u_k)$, K being the number of components, and u_k being the "pseudo-input" learned through back-propagation.

We performed a random search on the number components $K \in \{10, 20, 30, 40, 50\}$ and the number of linear scheduling steps $\in \{0, 20, 40\}$. The results are reported in Table D.25.

Table D.25: Results of the random search on the hyper-parameters of the VAMP: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

number components	linear scheduling steps	SSIM ↑	MSE ($\times 10^{-3}$) ↓
20	40	0.702 ± 0.097	5.581 ± 0.874
10	20	0.686 ± 0.019	7.231 ± 5.478
10	20	0.678 ± 0.025	5.841 ± 0.902
20	20	0.633 ± 0.007	3.640 ± 0.025
20	0	0.631 ± 0.003	3.569 ± 0.108
20	20	0.628 ± 0.002	3.586 ± 0.120
30	20	0.625 ± 0.005	3.892 ± 0.150
30	0	0.622 ± 0.001	3.965 ± 0.091
40	40	0.621 ± 0.005	4.151 ± 0.226
40	0	0.620 ± 0.004	4.074 ± 0.169

Appendix D.16. Vector-quantized VAE

Van Den Oord et al. [28] suggested using discrete (rather than continuous) latent representations and having a learned (rather than static) prior. The latent space is structured as an $\mathbb{R}^{K\times D}$ vector space. We denote $\mathcal{E} = \{e_1, e_2, ..., e_K\}$ where $e_i \in \mathbb{R}^D$ for $i \in \{1, 2, ..., K\}$. We say that K is the size of the latent embedding space, and D is the dimension of the embedding vectors.

For an embedding size d, the input x is passed through the encoder to obtain the output $z_e(x) \in \mathbb{R}^{d\times D}$, which is then passed through the discretisation bottleneck to map it to an element of $z_q(x) \in \mathcal{E}^d$ such that $(z_q(x))_j = e_k$ where $k = \arg\min_l ||z_e(x) - e_l||_2$ for $j \in \{1, 2, ..., d\}$. As the argmin operation lacks differentiability, learning of the embeddings and regularisation of the latent space is managed by integrating the stopgradient operator sg into the training objective:

$$\mathcal{L}_{\text{VQVAE}}(x) = \log p(x|z_q(x)) + ||\text{sg}[z_e(x)] - e||_2^2 + \beta||z_e(x) - \text{sg}[e]||_2^2 \ .$$

As suggested by the authors, we replaced the term $||\text{sg}[z_e(x)] - e||_2^2$ in the loss by the exponential moving average (EMA) update with a decay of 0.99. We then considered two hyper-parameters in our random search: the size of the latent embedding space $K \in \{128, 256, 512, 1024\}$ and the regularization weight $\beta \in \{0.25, 0.5, 0.75, 0.9, 1, 1.5, 2, 4\}$. The results are reported in Table D.26.

Table D.26: Results of the random search on the hyper-parameters of the VQVAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

commitment loss factor	quantization loss factor	num embeddings	use EMA	decay	SSIM ↑	MSE ($\times 10^{-3}$) ↓
0.25	2	512	True	0.99	0.881 ± 0.003	1.805 ± 0.032
0.25	0.25	1024	True	0.99	0.880 ± 0.009	1.866 ± 0.064
0.25	0.5	256	True	0.99	0.879 ± 0.005	1.797 ± 0.037
0.25	0.25	512	True	0.99	0.877 ± 0.007	1.836 ± 0.093
0.25	4	1024	True	0.99	0.876 ± 0.005	1.854 ± 0.065
0.25	0.75	256	True	0.99	0.874 ± 0.011	1.896 ± 0.065
0.25	0.9	512	True	0.99	0.870 ± 0.004	1.927 ± 0.023
0.25	1.5	256	True	0.99	0.870 ± 0.011	1.856 ± 0.100
0.25	4	1024	True	0.99	0.870 ± 0.011	1.854 ± 0.056
0.25	1.5	1024	True	0.99	0.868 ± 0.014	1.827 ± 0.084

Appendix D.17. Wasserstein auto-encoder

The Wasserstein auto-encoder [26] introduces the use of a penalized form of the Wasserstein distance to measure the dissimilarity between the model's generated distribution and the true data distribution. This leads to more stable training, mitigating mode collapse and improving the model's ability to generate diverse and realistic samples.

$$\mathcal{L}_{WAE} = \mathcal{L}_{rec} + \lambda \mathcal{D}_Z(p_z(z), q_\phi(z)) \ ,$$

with $\mathcal{D}_Z$ an arbitrary divergence. Different divergences $\mathcal{D}_Z$ are suggested by the authors, we here use the maximum mean discrepancy (MMD).

We performed a random search on three parameters: the kernel choice $\in \{\text{rbf}, \text{imq}\}$, the regularization weight $\lambda \in \{0.01, 0.1, 0.5, 1, 5, 10, 100\}$ and the kernel bandwidth $\in \{0.01, 0.1, 0.5, 1, 5, 10, 100\}$. The results are reported in Table D.27.

Table D.27: Results of the random search on the hyper-parameters of the WAE: ranking according to the SSIM of the 10 best configurations (mean ± std over the three folds randomly selected).

kernel choice	regularization weight	kernel bandwidth	SSIM ↑	MSE ($\times 10^{-3}$) ↓
rbf	0.1	5	0.881 ± 0.005	1.862 ± 0.075
rbf	0.5	0.1	0.880 ± 0.002	1.835 ± 0.096
rbf	0.5	0.5	0.879 ± 0.004	1.798 ± 0.035
rbf	0.01	0.1	0.879 ± 0.006	1.866 ± 0.061
imq	5	1	0.878 ± 0.003	1.838 ± 0.073
rbf	10	5	0.877 ± 0.005	1.838 ± 0.090
imq	1	1	0.876 ± 0.006	1.835 ± 0.097
imq	1	0.01	0.876 ± 0.007	1.882 ± 0.067
imq	5	100	0.874 ± 0.002	1.894 ± 0.070
imq	100	100	0.874 ± 0.016	1.865 ± 0.069

Appendix E. Model training

Table E.28: SSIM obtained on each validation set of the 6-fold cross-validation for the different trained models (mean ± std over the images from the validation set). The best split of each VAE variant is highlighted in bold.

Models	Split 0	Split 1	Split 2	Split 3	Split 4	Split 5
Adv. AE [23]	0.865 ± 0.032	0.860 ± 0.026	0.876 ± 0.036	$\mathbf{0.876 \pm 0.024}$	0.875 ± 0.030	0.859 ± 0.040
AE	0.866 ± 0.034	0.852 ± 0.032	$\mathbf{0.868 \pm 0.034}$	0.881 ± 0.024	0.871 ± 0.029	0.862 ± 0.037
β-TC VAE [16]	0.858 ± 0.036	0.869 ± 0.024	0.869 ± 0.030	$\mathbf{0.876 \pm 0.024}$	0.860 ± 0.035	0.871 ± 0.037
β-VAE [19]	0.870 ± 0.029	0.869 ± 0.024	$\mathbf{0.874 \pm 0.030}$	0.872 ± 0.027	0.865 ± 0.032	0.864 ± 0.036
Dis. β-VAE [14]	0.870 ± 0.029	0.868 ± 0.023	$\mathbf{0.879 \pm 0.027}$	0.868 ± 0.026	0.865 ± 0.033	0.865 ± 0.036
FactorVAE [20]	0.863 ± 0.033	0.849 ± 0.024	$\mathbf{0.874 \pm 0.033}$	0.872 ± 0.028	0.873 ± 0.027	0.861 ± 0.040
HVAE [15]	0.840 ± 0.040	0.869 ± 0.025	0.864 ± 0.031	0.867 ± 0.027	$\mathbf{0.878 \pm 0.027}$	0.858 ± 0.038
InfoVAE [29]	0.870 ± 0.030	0.871 ± 0.025	0.874 ± 0.031	0.873 ± 0.024	$\mathbf{0.876 \pm 0.027}$	0.870 ± 0.036
IWAE [13]	0.861 ± 0.036	0.860 ± 0.031	0.867 ± 0.033	0.868 ± 0.028	$\mathbf{0.875 \pm 0.026}$	0.861 ± 0.046
RAE-GP [18]	0.878 ± 0.030	0.872 ± 0.028	$\mathbf{0.884 \pm 0.029}$	0.884 ± 0.029	0.880 ± 0.026	0.873 ± 0.033
RAE-ℓ^2 [18]	0.857 ± 0.037	0.873 ± 0.025	0.850 ± 0.040	$\mathbf{0.882 \pm 0.023}$	0.868 ± 0.031	0.863 ± 0.038
VAE [4]	0.866 ± 0.030	0.868 ± 0.024	$\mathbf{0.869 \pm 0.030}$	0.865 ± 0.029	0.851 ± 0.041	0.868 ± 0.037
VAEGAN [22]	0.804 ± 0.044	0.863 ± 0.025	0.846 ± 0.038	$\mathbf{0.866 \pm 0.026}$	0.855 ± 0.038	0.858 ± 0.035
VAE-IAF [21]	0.827 ± 0.037	0.818 ± 0.032	$\mathbf{0.829 \pm 0.034}$	0.828 ± 0.027	0.828 ± 0.034	0.823 ± 0.037
VAE LinNF [24]	0.860 ± 0.033	$\mathbf{0.870 \pm 0.024}$	0.865 ± 0.035	0.852 ± 0.032	0.858 ± 0.039	0.870 ± 0.036
VQVAE [28]	0.857 ± 0.036	0.852 ± 0.027	0.869 ± 0.031	$\mathbf{0.883 \pm 0.025}$	0.868 ± 0.031	0.864 ± 0.037
WAE [26]	0.870 ± 0.032	0.871 ± 0.024	0.871 ± 0.033	$\mathbf{0.882 \pm 0.024}$	0.869 ± 0.034	0.862 ± 0.038

Appendix F. Reconstructions

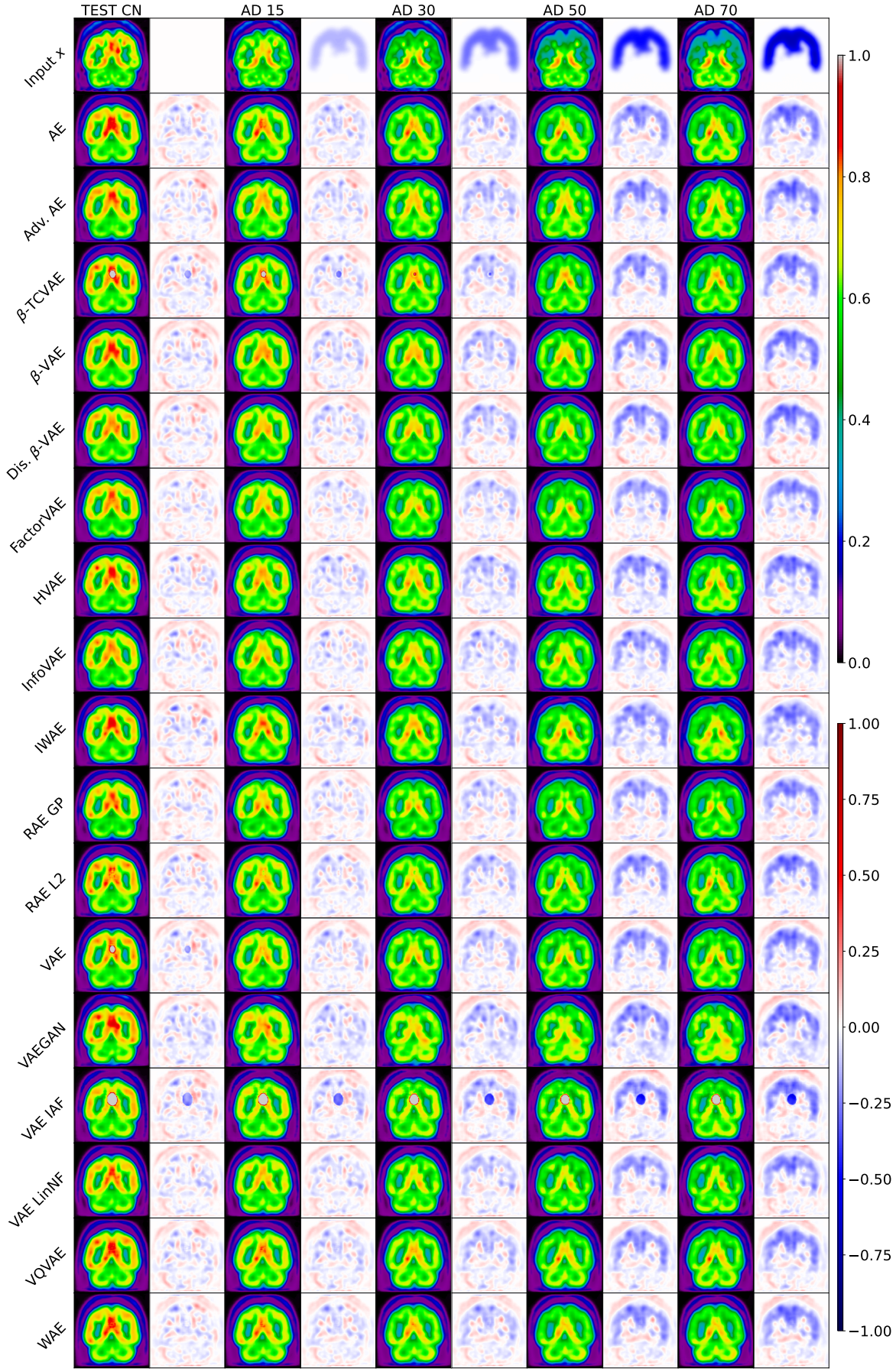


Figure F.10: Examples of reconstructions (coronal slices) obtained with the different VAE variants from the original image of a cognitively normal subject (images of the first column, Test CN) and from the same subject with AD simulated at different intensity degrees (AD 15, AD 30, AD 50 and AD 70). The first row shows the input image in odd columns and the mask of the simulated disease in even columns when the input is a simulated image. All the other rows are the pseudo-healthy reconstructions of the models in odd columns and the difference between the pseudo-healthy reconstruction and the input in even columns.

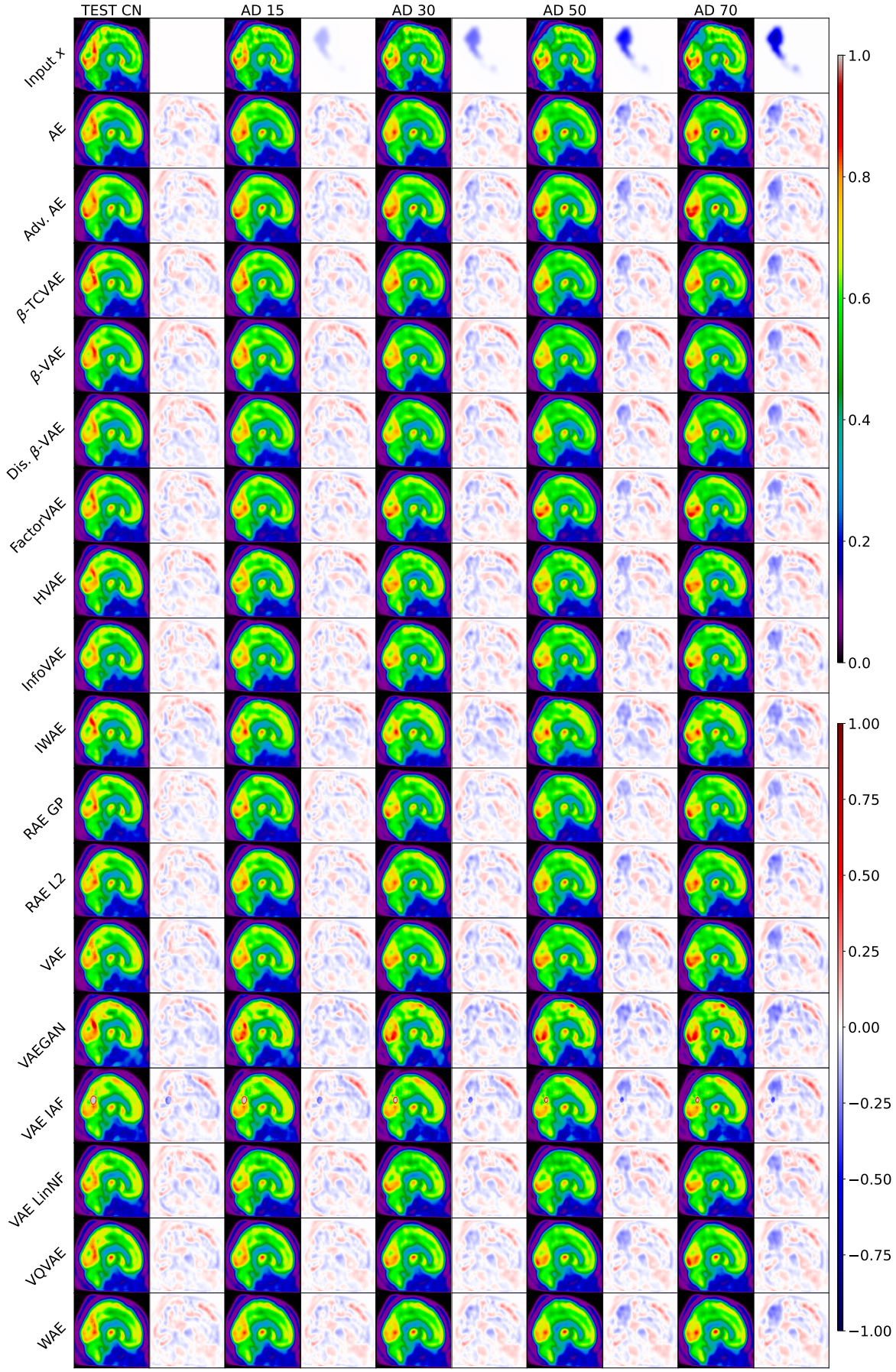


Figure F.11: Examples of reconstructions (sagittal slices) obtained with the different VAE variants from the original image of a cognitively normal subject (images of the first column, Test CN) and from the same subject with AD simulated at different intensity degrees (AD 15, AD 30, AD 50 and AD 70). The first row shows the input image in odd columns and the mask of the simulated disease in even columns when the input is a simulated image. All the other rows are the pseudo-healthy reconstructions of the models in odd columns and the difference between the pseudo-healthy reconstruction and the input in even columns.

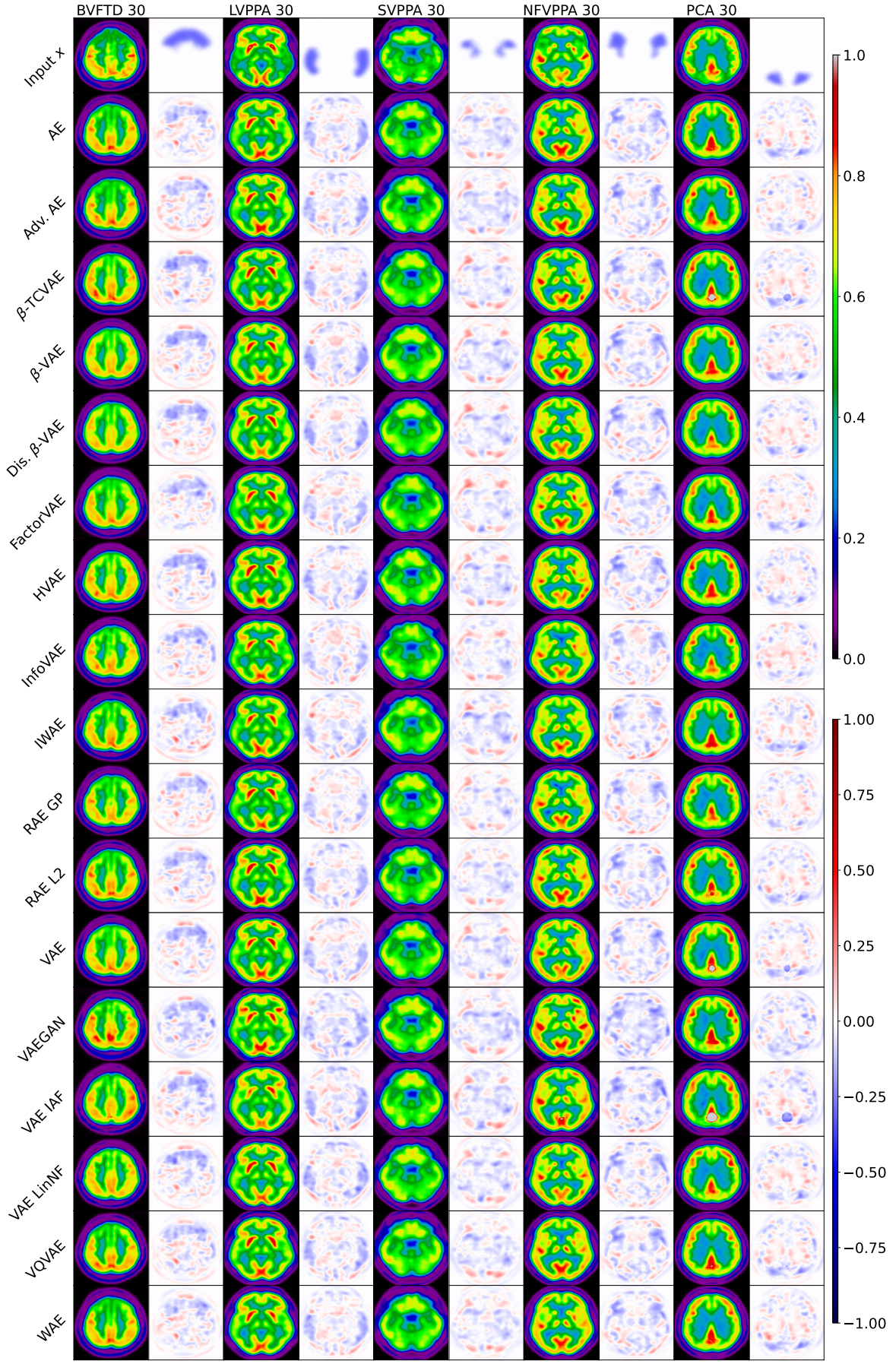


Figure F.12: Examples of reconstructions (axial slices) obtained with the different VAE variants from the same subject with different dementia subtypes simulated at 30% intensity degree (bvFTD 30, lvPPA 30, svPPA 30, nfvPPA 30 and PCA 30). The first row shows the input image in odd columns and the mask of the simulated disease in even columns. All the other rows are the pseudo-healthy reconstructions of the models in odd columns and the difference between the pseudo-healthy reconstruction and the input in even columns.