Mixture of Inference Networks for VAE-based Audio-visual Speech Enhancement

Mostafa Sadeghi and Xavier Alameda-Pineda, *Senior Member, IEEE*

***Abstract*—In this paper, we are interested in unsupervised (unknown noise) speech enhancement, where the probability distribution of clean speech spectrogram is simulated via a latent variable generative model, also called the decoder. Recently, variational autoencoders (VAEs) have gained much popularity as probabilistic generative models. In VAEs, the posterior of the latent variables is computationally intractable, and it is approximated by a so-called encoder network. Motivated by the fact that visual data, i.e. lip images of the speaker, provide helpful and complementary information about speech, some audio-visual VAE architectures have been recently proposed. The initialization of the latent variables at test time is crucial as the overall inference problem is non-convex. This is usually done by using the output of the encoder where the noisy audio and clean visual data are given as input. Current audio-visual models do not provide an effective initialization because the two modalities are tightly coupled (concatenated) in the associated architectures. To overcome this issue, we introduce the mixture of inference networks variational autoencoder (MIN-VAE), inspired by mixture models. Two encoder networks input, respectively, audio and visual data, and the posterior of the latent variables is modeled as a mixture of two Gaussian distributions output from each encoder network. The mixture variable is also latent, and therefore the inference of learning the optimal balance between the audio and visual inference network is unsupervised as well. By training a shared decoder, the overall network learns to adaptively fuse the two modalities. Moreover, at test time, the visual encoder, which takes (clean) visual data, is used for initialization. A variational inference approach is derived to train the proposed generative model. Thanks to the novel inference procedure and the robust initialization, the proposed audio-visual VAE exhibits superior performance on speech enhancement than using the standard audio-only as well as audio-visual counterparts.**

***Index Terms*—Audio-visual speech enhancement, generative models, variational auto-encoder, mixture model.**

I. Introduction

SPEECH enhancement, or removing background noise from noisy speech [1], [2], is a classic yet very important problem in signal processing and machine learning. Traditional solutions to this problem are based on spectral subtraction [3] and Wiener filtering [4], targeting noise and/or speech power spectral density (PSD) estimation in the short-time Fourier transform (STFT) domain. The recent impressive performance of deep neural networks (DNNs) in computer vision and machine learning has paved the way to revisit the speech enhancement problem. DNNs have been widely utilized in this regard, where a neural network is trained to map a noisy speech spectrogram to its clean version, or to a time frequency (TF) mask [5]–[7]. This is usually done in a supervised way, using a huge dataset of noise and clean speech signals for training. As such, the performance of a supervised speech enhancement technique often degrades when dealing with an unknown type of noise.

Unsupervised techniques provide another procedure for speech enhancement that does not use noise signals for training. A popular unsupervised method is based on nonnegative matrix factorization (NMF) [8]–[10] for modeling the PSD of speech signals [11], which decomposes PSD as a product of two non-negative low-rank matrices (a dictionary of basis spectra and the corresponding activations). An NMF-based speech enhancement method consists of first learning a set of basis spectra for clean speech spectrograms at training phase, prior to speech enhancement [9], [12], [13]. Then, by decomposing the noisy spectrogram as the sum of clean speech and noise spectrograms, the corresponding clean speech activations as well as the NMF parameters of noise are estimated. While being computationally efficient, this modeling and enhancement framework cannot properly explain complicated structure of speech spectrogram due to the limited representational power dictated by the two low-rank matrices. A deep autoencoder (DAE) has been employed in [14] to model clean speech and noise spectrograms. A DAE is pre-trained for clean speech spectrograms, while an extra DAE for noise spectrogram is trained at the enhancement stage using the noisy spectrogram. The corresponding inference problem is under-determined, and the authors proposed to constrain the unknown speech using a pre-trained NMF model. As such, this DAE-based method might encounter the same shortcomings as those of the NMF-based speech enhancement [15].

Deep latent variable models offer a more sophisticated and efficient modeling framework than NMF and DAE, gaining much interest over the past few years [15]–[21]. The first and main step is to train a generative model for clean speech spectrogram using a variational auto-encoder (VAE) [22], [23]. VAE provides an efficient way to estimate the parameters of a non-linear generative model, also called the decoder. This is done by approximating the intractable posterior distribution of the latent variables using a Gaussian distribution parametrized by a neural network, called the inference (encoder) network. The encoder and decoder are jointly trained to maximize a variational lower bound on the marginal data log-likelihood. At test time, the trained generative model is combined with a noise model, e.g. NMF. The unknown noise parameters and clean speech are then estimated from the observed noisy

Mostafa Sadeghi is with the Multispeech team at Inria Nancy - Grand Est, France. Xavier Alameda-Pineda is with the Perception team at Inria Grenoble Rhône-Alpes and Université Grenoble Alpes, France.

Xavier Alameda-Pineda acknowledges the ANR ML3RI (ANR-19-CE33-0008-01), the ANR-3IA MIAI (ANR-19-P3IA-0003) and the H2020 SPRING project (GA 871245).

speech. Being independent of the noise type at training, these methods show better generalization than the supervised approaches [15], [16].

Motivated by the fact that the visual information, when associated with audio information, often helps improve the performance in various tasks [24]–[26], and in particular the quality of speech enhancement [27]–[29], an audio-visual latent variable generative model has recently been proposed in [30]. Within this model, the visual features corresponding to the lips region of the speaker are also fed to the encoder and decoder networks of the VAE. The effectiveness and superior performance of the audio-visual VAE (AV-VAE) compared to the audio-only VAE (A-VAE), as well as the supervised deep learning based method of [29] has been experimentally verified in [30]. To deal with noisy visual data at test time, e.g. non-frontal or occluded lips images, a robust method has been proposed in [31], which was later improved and extended in [32]. In the proposed approach, a mixture of trained A-VAE and AV-VAE is used as the clean speech model during speech enhancement. Because of that, the deteriorating effects associated with missing/noisy visual information are avoided as the algorithm switches from AV-VAE to A-VAE in these cases [31]. Besides AV-VAE, a video-only VAE (V-VAE) has also been introduced in [30], where the posterior parameters of the latent variables, that is, the encoder parameters, are trained using only visual information. As such, the latent variables governing the generative process of clean speech spectrogram are inferred from visual data only. V-VAE has been shown to yield much better speech enhancement performance than A-VAE when the (acoustic) noise level is high [30].

In the speech enhancement phase, because of the non-linear generative model, the posterior of the latent variables does not admit a closed-form expression. Two approaches are often used to get around this problem. The first solution is based on the Markov Chain Monte Carlo (MCMC) method [33], in which a sampling technique, e.g. the Metropolis-Hastings algorithm [33], is used to sample from the posterior [15], [16]. The obtained samples are then used to approximate the expectations using a Monte-Carlo average. The second approach makes use of optimization techniques to find the maximum a posteriori estimation of the latent variables [20]. In either case, the initialization plays an important role, as the associated problems are highly non-convex. In practice, the trained encoder is used to initialize the latent variables by giving the noisy speech spectrogram as the input and taking the posterior mean at the output. This can partly explain why V-VAE performs better than A-AVE at high noise levels. In fact, the latent variable initialization in V-VAE is based on visual features, whereas in A-VAE, it is based on the noisy speech. As a result, V-VAE provides a better initialization, because it uses noise-free data (visual features) [30].

The original contribution of this paper is to optimally exploit the complementarity of A-VAE and V-VAE, without systematic recourse to simultaneously using audio and visual features, i.e. via simple concatenation (tight fusion) as done in AV-VAE. Indeed, we aim to bridge the performance gap between A-VAE and V-VAE by designing a mixture of audio and visual inference (encoder) networks, called mixture of inference networks VAE (MIN-VAE). The inputs to audio and visual encoders are speech spectrogram frames and the corresponding visual features, respectively, thus training MIN-VAE to select the best combination of the the audio and visual information. A variational inference approach is proposed to train the mixture of the two encoders jointly with a shared decoder (generative) network. This way, the decoder reconstructs the input audio data using the optimal combination of the audio and visual latent samples. At test time, the latent variables are initialized using the visual encoder, thus providing a robust initialization. Our experiments show that MIN-VAE yields much better performance than previous methods, i.e. A-VAE, V-VAE, and AV-VAE.

It should be noted that there are some fundamental differences between our proposed MIN-VAE and the mixture model introduced in [31]. While the purpose of our work is to combine an A-VAE with a V-VAE to take advantage of the both in terms of robust initialization and an improved generative model, the work in [31] addresses robust audio-visual speech enhancement. We achieve our goal by proposing a VAE architecture with a single decoder but a mixture of audio- and visual-based encoders. A new inference method is also derived to train the proposed VAE. In [31], the robustness is achieved by considering a mixture of an A-VAE's decoder and an AV-VAE's decoder at test phase. Both the decoders have been trained separately (using standard A-VAE and AV-VAE), and no particular VAE architecture is trained. In contrast to our present work, the architecture proposed in [31] does not provide robustness to latent initialization as it uses a VAE architecture where the audio and visual modalities are tightly fused.

The rest of the paper is organized as follows. In Section II, we review clean speech modeling using already proposed VAE architectures. Next, Section III introduces our proposed MIN-VAE modeling and the associated speech enhancement strategy. Experimental results are then presented in Section IV.

II. VAE-based Speech Modeling

In this section, audio-only, visual-only and audio-visual clean speech modeling based on VAE is reviewed. Roughly speaking, this consists in defining a latent variable generative model for each time frame of clean speech spectrogram. A parametric Gaussian distribution is used to define the conditional distribution of a spectrogram time frame given its associated latent variable (and, depending on the choice, the visual feature vector). The parameters of the distribution are modeled by DNNs. Assuming a standard Gaussian prior distribution for the latent variables, the model (DNN) parameters are then learned from a collection of clean training data using variational inference. To do so, the posterior distribution of the latent variables is approximated by a Gaussian distribution parametrized by a DNN, called the encoder network. In what follows, three VAE-based modeling frameworks are reviewed.

A. Audio-only VAE

Let $\mathbf{s}_n \in \mathbb{C}^F$ denote the vector of speech STFT coefficients at time frame n, for $n \in \{0, ..., N-1\}$, which is assumed

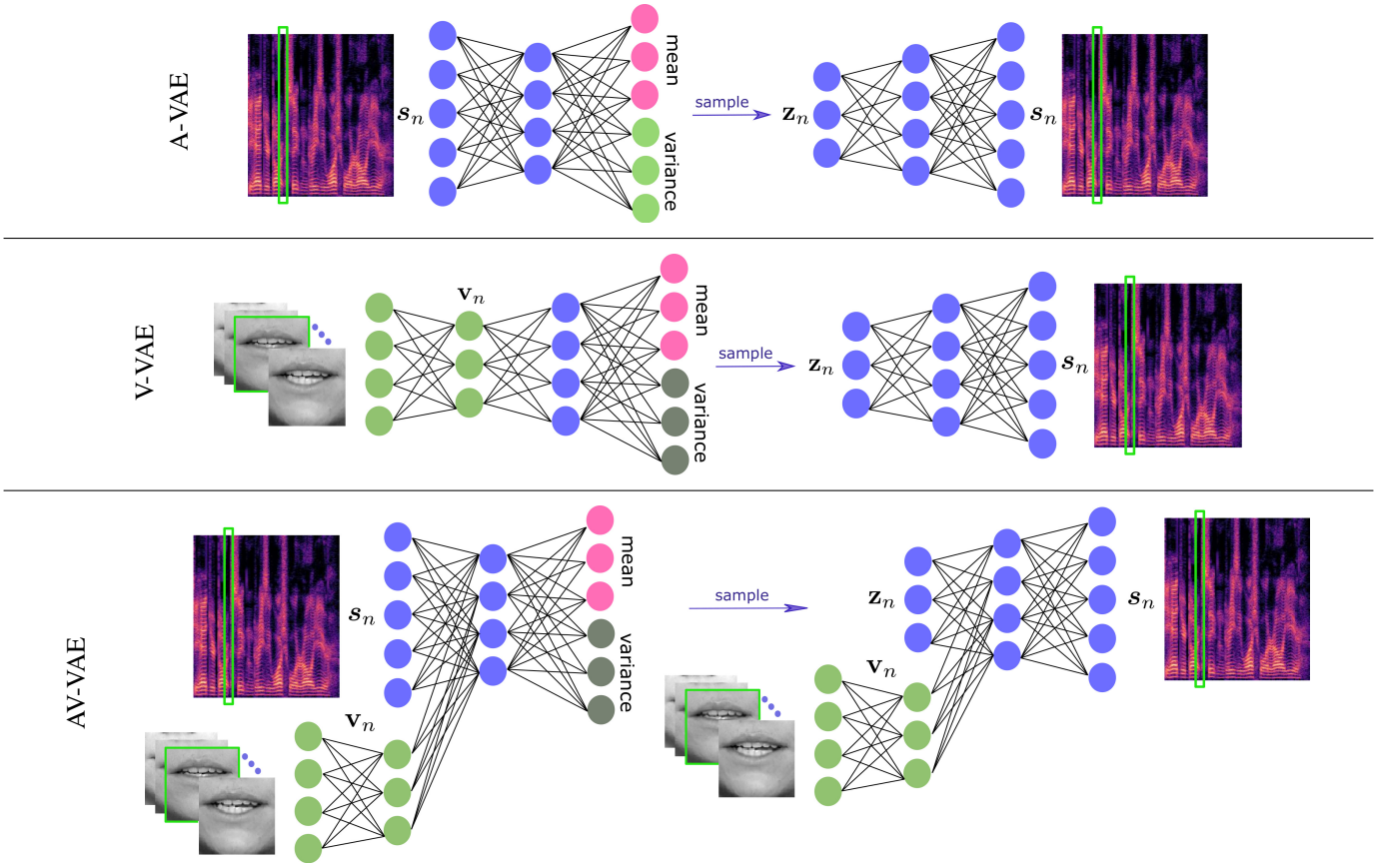


Fig. 1: Architectures for (top) the Audio-only VAE (A-VAE) proposed in [16], (middle) the Video-only VAE (V-VAE) proposed in [30] and (bottom) the Audio-Visual VAE (AV-VAE) proposed in [30].

to be generated according to the following latent variable model [15], [16]:

$$\mathbf{s}_n|\mathbf{z}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}\Big(\boldsymbol{\sigma}_s(\mathbf{z}_n)\Big)\Big), \quad (1)$$

$$\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (2)$$

where $\mathbf{z}_n \in \mathbb{R}^L$, with $L \ll F$, is a latent random variable, $\mathcal{N}_c(\mathbf{0}, \boldsymbol{\Sigma})$ denotes a zero-mean complex proper Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}$, and $\mathcal{N}(\mathbf{0}, \mathbf{I})$ stands for a zero-mean Gaussian distribution with an identity covariance matrix. Moreover, $\boldsymbol{\sigma}_s(.) : \mathbb{R}^L \mapsto \mathbb{R}_+^F$ is modeled with a neural network parameterized by $\boldsymbol{\theta}$, which is called the *decoder*.

In order to estimate the set of parameters, $\boldsymbol{\theta}$, the VAE formalism proposes to approximate the intractable posterior distribution $p(\mathbf{z}_n|\mathbf{s}_n)$ by a variational distribution parametrized by a neural network, called the *inference (encoder) network* [23]. This variational distribution writes:

$$q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a) = \mathcal{N}\Big(\boldsymbol{\mu}_z^a(\mathbf{s}_n), \mathrm{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big), \quad (3)$$

where, $\boldsymbol{\mu}_z^a(.) : \mathbb{R}_+^F \mapsto \mathbb{R}^L$ and $\boldsymbol{\sigma}_z^a(.) : \mathbb{R}_+^F \mapsto \mathbb{R}_+^L$ are neural networks, with parameters denoted $\boldsymbol{\phi}_a$, taking $\tilde{\mathbf{s}}_n \triangleq (|s_{0n}|^2 \dots |s_{F-1\,n}|^2)^\top$ as input. Given a sequence of STFT speech time frames $\mathbf{s} = \{\mathbf{s}_n\}_{n=0}^{N_{tr}-1}$ as training data, with $\mathbf{z} = \{\mathbf{z}_n\}_{n=0}^{N_{tr}-1}$ being the associated latent variables, the parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}_a\}$ are then estimated by maximizing a lower bound on the data log-likelihood $\log p(\mathbf{s}; \boldsymbol{\theta})$. Note that,

$$\log p(\mathbf{s}; \boldsymbol{\theta}) = \log \int p(\mathbf{s}|\mathbf{z}; \boldsymbol{\theta}) p(\mathbf{z}) d\mathbf{z}$$
$$\geq \mathbb{E}_{q(\mathbf{z}|\mathbf{s};\boldsymbol{\phi}_a)}\Big[\log \frac{p(\mathbf{s}|\mathbf{z}; \boldsymbol{\theta}) p(\mathbf{z})}{q(\mathbf{z}|\mathbf{s}; \boldsymbol{\phi}_a)}\Big] \triangleq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}_a) \quad (4)$$

where, the Jensen's inequality has been used, as it is classically done, see [23]. The function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}_a)$ is called the evidence lower bound (ELBO) [23], because it provides a lower bound on $\log p(\mathbf{s}; \boldsymbol{\theta})$. The ELBO can be decomposed as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}_a) = \mathbb{E}_{q(\mathbf{z}|\mathbf{s};\boldsymbol{\phi}_a)}\Big[\ln p(\mathbf{s}|\mathbf{z}; \boldsymbol{\theta})\Big] -$$
$$\mathcal{D}_{KL}\Big(q(\mathbf{z}|\mathbf{s}; \boldsymbol{\phi}_a) \parallel p(\mathbf{z})\Big), \quad (5)$$

where, $\mathcal{D}_{KL}(q \parallel p)$ denotes the Kullback-Leibler (KL) divergence between q and p. The first term in the right-hand side of (5) evaluates the reconstruction quality of the decoder, and the second one is a regularization term encouraging the variational posterior to remain close to the prior. As the expectation in (5) is computationally intractable, it is usually approximated by a single sample drawn from $q(\mathbf{z}|\mathbf{s}; \boldsymbol{\phi}_a)$ [23]. Employing a so-called re-parameterization trick, the set of parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}_a\}$ is estimated by a stochastic gradient ascent algorithm [23]. Since all the parameters are inferred using only audio data, the

above model is called A-VAE [30]. The associated architecture is shown in Fig. 1 (top).

B. Visual-only VAE

A visual VAE (V-VAE) is proposed in [30], assuming the same generative model as in (1) and (2). The difference with the A-VAE is that, here, the posterior $p(\mathbf{z}_n|\mathbf{s}_n)$ is approximated using visual-data only:

$$q(\mathbf{z}_n|\mathbf{v}_n, \boldsymbol{\phi}_v) = \mathcal{N}\Big(\boldsymbol{\mu}_z^v(\mathbf{v}_n), \text{diag}\Big(\boldsymbol{\sigma}_z^v(\mathbf{v}_n)\Big)\Big), \tag{6}$$

where, $\mathbf{v}_n \in \mathbb{R}^M$ is an embedding for the image of the speaker lips at frame n, and $\boldsymbol{\mu}_z^v(.) : \mathbb{R}^M \mapsto \mathbb{R}^L$ and $\boldsymbol{\sigma}_z^v(.) : \mathbb{R}^M \mapsto \mathbb{R}_+^L$ denote neural networks with parameters $\boldsymbol{\phi}_v$. Hence, V-VAE attempts to reconstruct clean speech using latent variables inferred from the lips images. The set of parameters, $\{\boldsymbol{\theta}, \boldsymbol{\phi}_v\}$, is estimated in the same way as A-VAE. Figure 1 (middle) depicts the architecture of a V-VAE.

C. Audio-Visual VAE

An audio-visual VAE, called AV-VAE, is also presented in [30] for speech modeling. The rationale of the AV-VAE is to exploit the complementary between audio and visual modalities. The associated generative model is defined as:

$$\mathbf{s}_n|\mathbf{z}_n; \mathbf{v}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}\Big(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n)\Big)\Big), \tag{7}$$

$$\mathbf{z}_n|\mathbf{v}_n \sim \mathcal{N}\Big(\boldsymbol{\mu}_z^{av}(\mathbf{v}_n), \text{diag}\Big(\boldsymbol{\sigma}_z^{av}(\mathbf{v}_n)\Big)\Big), \tag{8}$$

where, $\boldsymbol{\sigma}_s(.,.) : \mathbb{R}^L \times \mathbb{R}^M \mapsto \mathbb{R}_+^F$ is a neural network taking $(\mathbf{z}_n, \mathbf{v}_n)$ as input. Furthermore, $\boldsymbol{\mu}_z^{av}(.) : \mathbb{R}^M \mapsto \mathbb{R}^L$ and $\boldsymbol{\sigma}_z^{av}(.) : \mathbb{R}^M \mapsto \mathbb{R}_+^L$ are neural networks parameterizing the mean and variance of the prior distribution of $\mathbf{z}_n$ using $\mathbf{v}_n$ as the input. Note that throughout the paper, $\mathbf{v}_n$ is treated as a deterministic piece of information, and so we do not model its generative process. The variational approximation to $p(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n)$ takes a similar form as (3), except that $\mathbf{v}_n$ is also fed to the associated neural network. The architecture of an AV-VAE is shown in Fig. 1 (bottom).

III. The Mixture of Inference Networks VAE

In this section, we aim to devise a framework able to choose the best combination between the auditory and visual modalities in the encoder, as opposed to systematically using both encodings through tight fusion, as done in AV-VAE. To achieve this goal, we propose a probabilistic mixture of an audio and a visual encoder, and name it mixture of inference networks VAE (MIN-VAE). Intuitively, the model learns to infer to which extent the approximate posterior of $\mathbf{z}_n$ should be audio- or visual-based. The overall architecture is depicted in Fig. 2. In the following, we introduce the mathematical formulations associated with the proposed MIN-VAE. The generative model is presented in Subsection III-A, which uses a mixture of two different Gaussian distributions for the prior of the latent variables, as opposed to standard VAE and to the models in the previous section that use a standard Gaussian distribution. This innovative choice is motivated to ease the task of the generative model. Indeed, by modeling two different prior distributions, the decoder network can easily understand if the sample is audio-based or visual-based. Subsection III-B proposes a variational distribution to approximate the intractable posterior of the latent variables. This variational distribution is then used in Subsection III-C to derive the training algorithm for the overall MIN-VAE. Finally, noise modelling for speech enhancement at test time is discussed in Subsection III-D.

A. The Generative Model

We assume that each latent code is generated either from an audio or from a visual prior. We model this with a mixing variable $\alpha_n \in \{0, 1\}$ describing whether the latent code $\mathbf{z}_n$ corresponds to the audio or to the visual prior. Once the latent code is generated from the corresponding prior, the speech frame $\mathbf{s}_n$ follows a complex Gaussian distribution with the variance computed by the decoder. We recall that the variance is a non-linear transformation of the latent code.

Formally, each STFT time frame $\mathbf{s}_n$ is modeled as:

$$\mathbf{s}_n|\mathbf{z}_n; \mathbf{v}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}\Big(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n)\Big)\Big), \tag{9}$$

$$\mathbf{z}_n|\alpha_n \sim \Big[\mathcal{N}(\boldsymbol{\mu}_a, \sigma_a\mathbf{I})\Big]^{\alpha_n} \cdot \Big[\mathcal{N}(\boldsymbol{\mu}_v, \sigma_v\mathbf{I})\Big]^{1-\alpha_n}, \tag{10}$$

$$\alpha_n \sim \pi^{\alpha_n} \times (1-\pi)^{1-\alpha_n}, \tag{11}$$

where the audio and visual priors are parametrized by $(\boldsymbol{\mu}_a, \sigma_a)$ and $(\boldsymbol{\mu}_v, \sigma_v)$ respectively, and α_n is assumed to follow a Bernoulli distribution with parameter π. We propose two versions of this architecture, namely: MIN-VAE-v1 where the decoder (9) takes the same form as (7) and uses explicitly visual information (see Fig. 2), and MIN-VAE-v2 where the decoder (9) takes the same form as (1) and does not use explicitly visual information. In both cases the parameters of the decoder are denoted by $\boldsymbol{\theta}$. The derivations will be done for the general case, that is MIN-VAE-v1.

B. The Posterior Distribution

In order to estimate the parameters of the generative model described above, i.e. $\boldsymbol{\psi} = \{\boldsymbol{\mu}_a, \boldsymbol{\mu}_v, \sigma_a, \sigma_v\}$, $\boldsymbol{\theta}$, and π, we follow a maximum likelihood procedure. To derive it, we need to compute the posterior of the latent variables:

$$p(\mathbf{z}_n, \alpha_n|\mathbf{s}_n; \mathbf{v}_n) = p(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n) \cdot p(\alpha_n|\mathbf{s}_n; \mathbf{v}_n). \tag{12}$$

The individual factors in the right-hand side of the above equation cannot be computed in closed-form, due to the non-linear generative model. As similarly done in VAE, we pursue an amortized inference approach to approximate $p(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n, \alpha_n)$ with a parametric Gaussian distribution defined as follows:

$$q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi}) = \begin{cases} q(\mathbf{z}_n|\mathbf{s}_n; \boldsymbol{\phi}_a) & \alpha_n = 1, \\ q(\mathbf{z}_n|\mathbf{v}_n; \boldsymbol{\phi}_v) & \alpha_n = 0, \end{cases} \tag{13}$$

in which, $\boldsymbol{\phi} = \{\boldsymbol{\phi}_a, \boldsymbol{\phi}_v\}$, and $\boldsymbol{\phi}_a$ and $\boldsymbol{\phi}_v$ denote the parameters of the associated audio and visual inference neural networks, taking the same architectures as those in (3) and (6), respectively. For the posterior of α_n, i.e. $p(\alpha_n|\mathbf{s}_n; \mathbf{v}_n)$, we resort to a variational approximation, denoted $r(\alpha_n)$. Put it all together, we have the following approximate posterior:

$$q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi}) \cdot r(\alpha_n) \approx p(\mathbf{z}_n, \alpha_n|\mathbf{s}_n; \mathbf{v}_n). \tag{14}$$

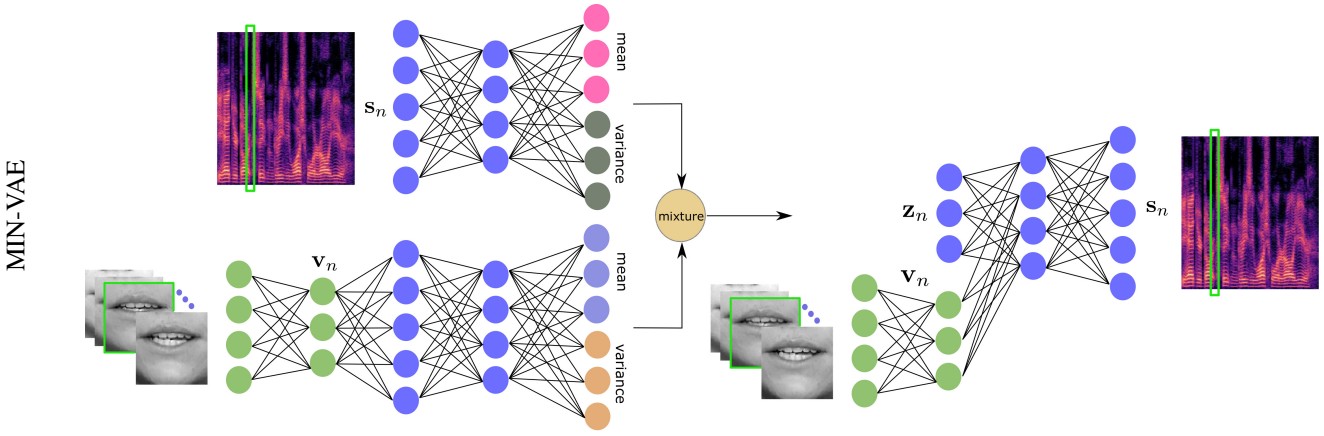


Fig. 2: Architecture of the proposed mixture of inference networks VAE (MIN-VAE). A mixture of an audio- and a visual-based encoder is used to approximate the intractable posterior distribution of the latent variables.

C. Training the MIN-VAE

In order to train the MIN-VAE, we devise an optimization procedure alternating between estimating $\boldsymbol{\Theta} = \{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}, \pi\}$ and updating the variational posterior r. To do so, we first need to give an expression of the exact posterior distribution, so as to write the optimization function, namely the KL-divergence between the approximate variational posterior and the true posterior. To write the exact posterior, we will use the decomposition of the generative model in equations (9) – (11), and recall the definition $\mathbf{s} = \{\mathbf{s}_n\}_{n=1}^{N_{tr}}$, and $\mathbf{z}$, and define $\boldsymbol{\alpha}$ and $\mathbf{v}$ analogously. The full posterior of the latent variables writes:

$$p(\mathbf{z}, \boldsymbol{\alpha}|\mathbf{s}; \mathbf{v}, \boldsymbol{\theta}) = \frac{p(\mathbf{s}, \mathbf{z}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\theta})}{p(\mathbf{s}; \mathbf{v}, \boldsymbol{\theta})} = \frac{p(\mathbf{s}|\mathbf{z}; \mathbf{v}, \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})}{p(\mathbf{s}; \mathbf{v}, \boldsymbol{\theta})}. \quad (15)$$

We then target the KL-divergence between the approximate posterior and the true posterior which reads:

$$\mathcal{D}_{KL}\Big(q(\mathbf{z}|\mathbf{s}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\phi}) r(\boldsymbol{\alpha}) \Big\| p(\mathbf{z}, \boldsymbol{\alpha}|\mathbf{s}; \mathbf{v}, \boldsymbol{\theta})\Big) = \sum_{\boldsymbol{\alpha}} \int_{\mathbb{Z}} q(\mathbf{z}|\mathbf{s}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\phi}) r(\boldsymbol{\alpha}) \log \frac{q(\mathbf{z}|\mathbf{s}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\phi}) r(\boldsymbol{\alpha})}{p(\mathbf{s}|\mathbf{z}; \mathbf{v}, \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z} = -\mathcal{L}(\boldsymbol{\Theta}, r) + \log p(\mathbf{s}; \mathbf{v}, \boldsymbol{\theta}) \geq 0, \quad (16)$$

where

$$\mathcal{L}(\boldsymbol{\Theta}, r) = \sum_{\boldsymbol{\alpha}} \int_{\mathbb{Z}} q(\mathbf{z}|\mathbf{s}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\phi}) r(\boldsymbol{\alpha}) \log \frac{p(\mathbf{s}|\mathbf{z}; \mathbf{v}, \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})}{q(\mathbf{z}|\mathbf{s}, \boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\phi}) r(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}. \quad (17)$$

From (16) we can see that $\log p(\mathbf{s}; \mathbf{v}, \boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\Theta}, r)$. Therefore, instead of maximizing the intractable data log-likelihood $\log p(\mathbf{s}; \mathbf{v}, \boldsymbol{\theta})$, we maximize its lower-bound, i.e. $\mathcal{L}(\boldsymbol{\Theta}, r)$, or equivalently:

$$\boldsymbol{\Theta}^*, r^* = \underset{\boldsymbol{\Theta}, r}{\operatorname{argmin}} -\mathcal{L}(\boldsymbol{\Theta}, r) \quad (18)$$

subject to the constraint that r integrates to one. We solve this problem by alternately optimizing the cost over r and $\boldsymbol{\Theta}$. In the following, the two optimization steps are discussed.

1) Optimizing w.r.t. $r(\boldsymbol{\alpha})$: With $\boldsymbol{\Theta}$ being fixed to its current estimate, solving (18) boils down to:

$$\min_{r} \sum_{\alpha_n} r_n(\alpha_n) \Big[\log \frac{r_n(\alpha_n)}{p(\alpha_n)} + J_n(\alpha_n) \Big], \forall n, \quad (19)$$

meaning that the optimal r is separable on n, where,

$$J_n(\alpha_n) = \int_{\mathbb{Z}} q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi}) \log \frac{q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi})}{p(\mathbf{s}_n|\mathbf{z}_n; \mathbf{v}_n, \boldsymbol{\theta}) p(\mathbf{z}_n|\alpha_n)} \mathrm{d}\mathbf{z}_n = \mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi}) \Big\| p(\mathbf{z}_n|\alpha_n)\Big) - \mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi})} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n; \mathbf{v}_n, \boldsymbol{\theta}) \Big]. \quad (20)$$

Using calculus of variations, we find that $r_n(\alpha_n) \propto p(\alpha_n) \exp\big(-J_n(\alpha_n)\big)$, which is a Bernoulli distribution. To find the associated parameter, we need to compute $J_n(\alpha_n)$. Since the expectation involved in (20) is intractable to compute, we approximate it using a single sample denoted $\mathbf{z}_n^{\alpha_n}$ drawn from $q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi})$, obtaining:

$$\tilde{J}_n(\alpha_n) = \mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n, \alpha_n; \mathbf{v}_n, \boldsymbol{\phi}) \Big\| p(\mathbf{z}_n|\alpha_n)\Big) - \log p(\mathbf{s}_n|\mathbf{z}_n^{\alpha_n}; \mathbf{v}_n, \boldsymbol{\theta}), \quad (21)$$

The parameter of the Bernoulli distribution then takes the following form:

$$\pi_n = g\Big(\tilde{J}_n(\alpha_n = 0) - \tilde{J}_n(\alpha_n = 1) + \log \frac{\pi}{1 - \pi}\Big), \quad (22)$$

where $g(x) = 1/(1 + \exp(-x))$ is the sigmoid function. Computations of the KL divergence terms are provided in Appendix A.

2) Optimizing w.r.t. $\boldsymbol{\Theta}$: With r being fixed to its current estimate, from (18), we can write the optimization over $\boldsymbol{\Theta}$ as:

$$\min_{\boldsymbol{\Theta}} \sum_{\boldsymbol{\alpha}} \int_{\mathbb{Z}} q(\mathbf{z}|\mathbf{s},\boldsymbol{\alpha};\mathbf{v},\boldsymbol{\phi}) r(\boldsymbol{\alpha}) \log \frac{q(\mathbf{z}|\mathbf{s},\boldsymbol{\alpha};\mathbf{v},\boldsymbol{\phi}) r(\boldsymbol{\alpha})}{p(\mathbf{s}|\mathbf{z};\mathbf{v},\boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}$$
$$= \min_{\boldsymbol{\Theta}} \sum_{n=0}^{N_{tr}} \pi_n \Big(\mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\phi}_a) \Big\| p(\mathbf{z}_n|\alpha_n=1) \Big) -$$
$$\mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\phi}_a)} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n;\mathbf{v}_n,\boldsymbol{\theta}) \Big] \Big) +$$
$$(1-\pi_n) \Big(\mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\phi}_v) \Big\| p(\mathbf{z}_n|\alpha_n=0) \Big) -$$
$$\mathbb{E}_{q(\mathbf{z}_n|\mathbf{v}_n;\boldsymbol{\phi}_v)} \Big[\log p(\mathbf{s}_n|\mathbf{z}_n;\mathbf{v}_n,\boldsymbol{\theta}) \Big] \Big) + \mathcal{D}_{KL}\Big(r\left(\alpha_n\right) \parallel p(\alpha_n) \Big). \tag{23}$$

As before, the expectations involved in the above equation are approximated with a single sample drawn from the associated posteriors. After computing the cost function, the parameters are updated using a re-parametrization trick along with a stochastic gradient descent algorithm, e.g. the Adam optimizer. Finally, optimizing (23) over π leads to minimizing the following KL-divergence:

$$\mathcal{D}_{KL}\Big(r\left(\alpha_n\right) \parallel p(\alpha_n) \Big) = \pi_n \log \frac{\pi_n}{\pi} + (1-\pi_n) \log \frac{1-\pi_n}{\pi}, \tag{24}$$

yielding

$$\pi = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \pi_n. \tag{25}$$

Now, with the derived variational inference formulas, we obtain the inference mixture for the MIN-VAE:

$$p(\mathbf{z}_n|\mathbf{s}_n;\mathbf{v}_n) = \pi_n \mathcal{N}\Big(\boldsymbol{\mu}_z^a(\mathbf{s}_n), \mathrm{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n) \Big) \Big) \tag{26}$$
$$+ (1-\pi_n) \mathcal{N}\Big(\boldsymbol{\mu}_z^v(\mathbf{v}_n), \mathrm{diag}\Big(\boldsymbol{\sigma}_z^v(\mathbf{v}_n) \Big) \Big). \tag{27}$$

The overall training algorithm then consists of alternating the variational distribution update of α_n via (22), the update of $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$ via stochastic gradient descent of (23), and the update of π using (25).

D. Noise Modeling

At test time, once the MIN-VAE is trained, the STFT time frames of the observed noisy speech are modeled as $\mathbf{x}_n = \mathbf{s}_n + \mathbf{b}_n$, for $n = 0, \ldots, N-1$, with $\mathbf{b}_n$ denoting noise STFT time frame. For the probabilistic modeling of $\mathbf{s}_n$, we use the generative model trained on clean data (i.e. the previous section). For $\mathbf{b}_n$, the following NMF based model is considered [16]:

$$\mathbf{b}_n \sim \mathcal{N}\Big(\mathbf{0}, \mathrm{diag}\Big(\mathbf{W}\boldsymbol{h}_n \Big) \Big), \tag{28}$$

where, $\mathbf{W} \in \mathbb{R}_+^{F \times K}$, and $\boldsymbol{h}_n$ denotes the n-th column of $\mathbf{H} \in \mathbb{R}_+^{K \times N}$. The parameters, i.e. $\{\mathbf{W}, \mathbf{H}\}$, as well as the unknown speech are then estimated following a variational inference method [33]. This strategy is inspired by the recent literature [16], [31]. The details are provided in Appendix B.

IV. Experiments

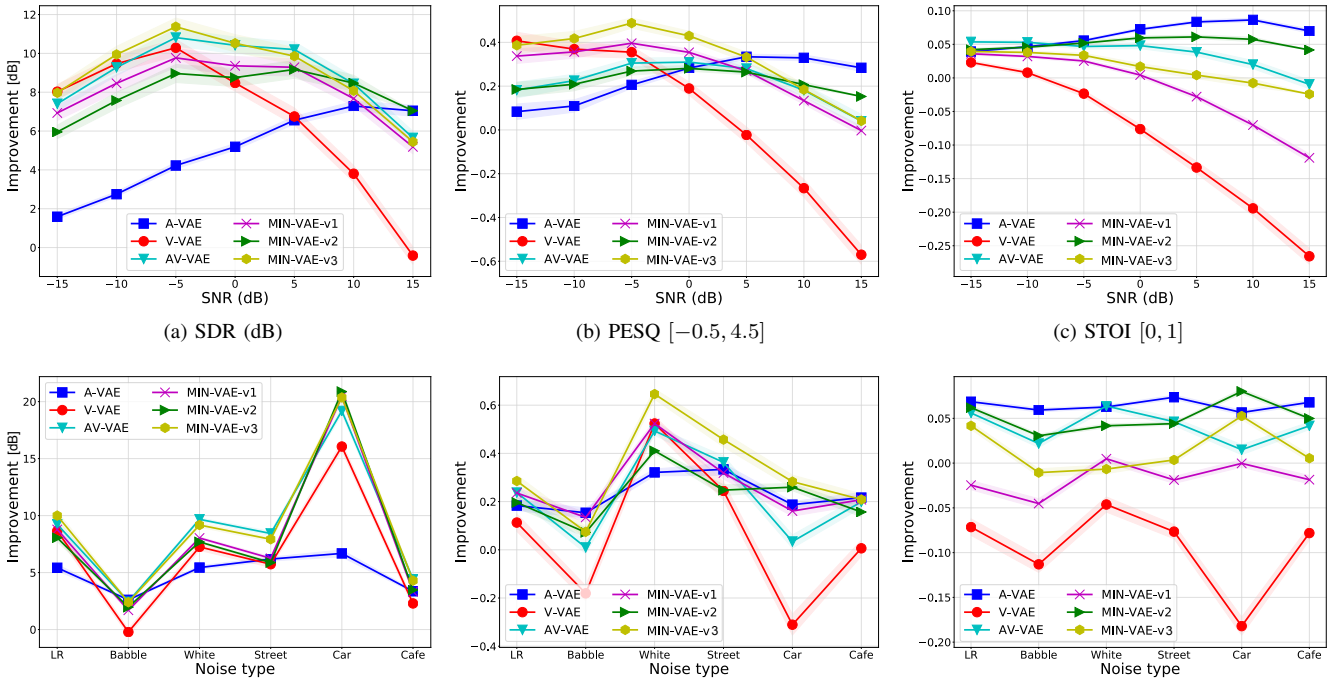
In this section, we aim to evaluate the speech enhancement performance of different VAE architectures, including A-VAE [16], V-VAE [30], AV-VAE [30], and the proposed MIN-VAE. To measure the performance, we use standard scores, including the signal-to-distortion ratio (SDR) [34], the perceptual evaluation of speech quality (PESQ) [35], and the short-time objective intelligibility (STOI) [36]. SDR is measured in decibels (dB), while PESQ and STOI values lie in the intervals $[-0.5, 4.5]$ and $[0, 1]$, respectively (the higher the better). For each measure, we report the averaged difference between the output value (evaluated on the enhanced speech signal) and the input value (evaluated on the noisy/unprocessed mixture signal). The average values of SDR, PESQ, and STOI computed on the input noisy speech signals are reported in Table II.

A. Experimental Set-up

a) Dataset: We use the NTCD-TIMIT dataset [37], which contains audio-visual (AV) recordings from 56 English speakers with an Irish accent, uttering 5488 different TIMIT sentences [38]. The visual data consist of 30 FPS videos of lips region of interests (ROIs). Each frame (ROI) is of size 67×67 pixels. The speech signal is sampled at 16 kHz, and the audio spectral features are computed using an STFT window of 64 ms (1024 samples per frame) with 47.9% overlap, hence $F = 513$. The dataset is divided into 39 speakers for training, 8 speakers for validation, and 9 speakers for testing, as proposed in [37]. The test set includes about 1 hour noisy speech, along with their corresponding lips ROIs, with six different noise types, including *Living Room (LR)*, *White*, *Cafe*, *Car*, *Babble*, and *Street*, with noise levels: $\{-15, -10, -5, 0, 5, 10, 15\}$ dB.

b) Architecture and training details: The generative networks (decoders) of A-VAE and V-VAE consist of a single hidden layer with 128 nodes and hyperbolic tangent activations. The dimension of the latent space is $L = 32$. The A-VAE encoder has a single hidden layer with 128 nodes and hyperbolic tangent activations. The V-VAE encoder is similar to the A-VAE encoder, except for extracting visual features, embedding lip ROIs into a feature vector $\mathbf{v}_n \in \mathbb{R}^M$, with $M = 128$. This is composed of two fully-connected layers with 512 and 128 nodes. The dimension of the input corresponds to a single vectorized frame, namely $4489 = 67 \times 67$. AV-VAE combines the architectures of A-VAE and V-VAE as illustrated in Fig. 1. The audio and the visual encoders in Fig. 2 share also the same architectures as those of A-VAE and V-VAE encoders, respectively.

To have a fair comparison, we fine-tunned the A-VAE and V-VAE of [30], which have been trained with a standard Gaussian prior for the latent variables, by using a parametric Gaussian prior, as the ones in (10). The decoder parameters of MIN-VAE-v1 and MIN-VAE-v2 (see Section III-A) are initialized with those of the pretrained AV-VAE and A-VAE, respectively. The parameters of the audio and the visual encoders are also initialized with the corresponding parameters in the pretrained A-VAE and V-VAE encoders. Then, all the parameters are fine-tuned using the Adam optimizer [39] with



(a) SDR (dB) (b) PESQ $[-0.5, 4.5]$ (c) STOI $[0, 1]$

(d) SDR (dB) (e) PESQ $[-0.5, 4.5]$ (f) STOI $[0, 1]$

Fig. 3: Performance comparison of different VAE architectures for speech enhancement. Top row shows the averaged results in terms of input noise levels, whereas the bottom row reports the averaged results versus different noise types. Here, no noise was added to the input of the audio-encoders of MIN-VAE-v1 and MIN-VAE-v2 during training.

a step size of 10^{-4}, for 100 epochs, and with a batch-size of 128.

We also considered another way to combine A-VAE with V-VAE, in which these two VAE architectures share the same decoder, and they are trained alternately. That is, at each epoch, the shared decoder is trained using latent samples coming from either the encoder of A-VAE or that of V-VAE. As a result, at each epoch, only the encoder parameters of the corresponding VAE, i.e. A-VAE or V-VAE, are updated while those of the other encoder are kept fixed. This training strategy is considered as a baseline where we do not use a mixture model for the encoder to automatically choose between the audio and the visual encoders. Instead, an alternating sampling from the two encoders is performed. We refer to the resulting VAE as MIN-VAE-v3. A description of all the proposed VAE architectures is given in Table I.

TABLE I: Description of the proposed VAE networks.

Name	Description
MIN-VAE-v1	The architecture shown in Fig. 2.
MIN-VAE-v2	Same as MIN-VAE-v1 but without using visual modality in the decoder.
MIN-VAE-v3	Alternately training an A-VAE and a V-VAE with a shared decoder.

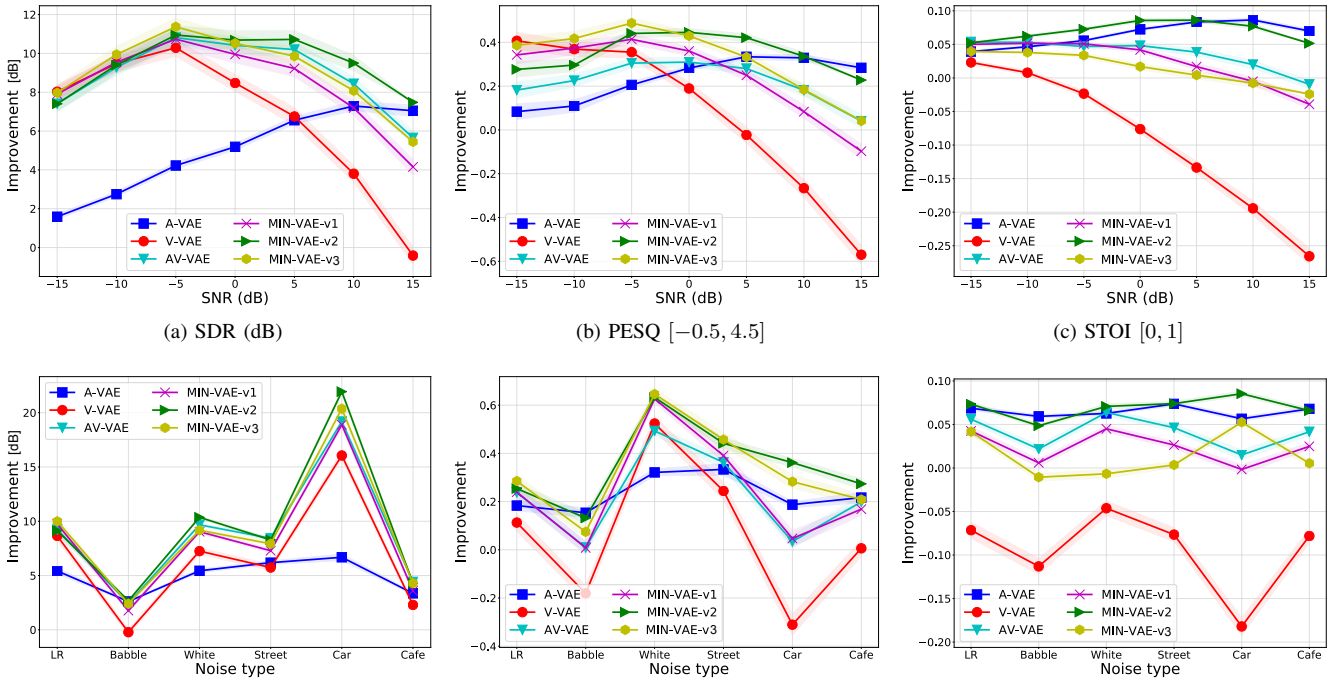
c) Speech enhancement parameters: For all the methods, the rank of $\mathbf{W}$ and $\mathbf{H}$ in the noise model (28) is set to $K = 10$, and these matrices are randomly initialized with non-negative entries. At the first iteration of the inference algorithms, the Markov chain of the Metropolis-Hastings algorithm (see Section B-A2 in Appendix B) is initialized by using the noisy observed speech and the visual features as input to the associated encoders, and taking the posterior mean as the initialization of the latent codes. For the proposed VAE architectures, i.e. MIN-VAE-v1, MIN-VAE-v2, and MIN-VAE-v3, the visual-encoders are used.

TABLE II: Average score values computed on the input noisy speech signals.

SNR (dB)	**-15**	**-10**	**-5**	**0**	**5**	**10**	**15**
SDR (dB)	-19	-16	-12.3	-7.4	-3	2	7
PESQ	1.22	1.31	1.44	1.69	1.98	2.32	2.64
STOI	0.32	0.38	0.46	0.55	0.65	0.76	0.81

B. *Results and Discussion*

Figure 3 summarizes the results of all the VAE architectures, in terms of SDR, PESQ, and STOI. The top row of this figure reports the averaged results versus different noise levels, whereas the bottom row shows the averaged results in terms of noise type. From this figure we can see that V-VAE performs pretty well at high noise levels. However, the intelligibility improvements in terms of STOI are not as good as those of the other algorithms. MIN-VAE-v3 outperforms other methods in terms of SDR and PESQ. Nevertheless, its intelligibility improvement is not satisfactory. The proposed MIN-VAE methods also outperform A-VAE, especially at



(a) SDR (dB) (b) PESQ $[-0.5, 4.5]$ (c) STOI $[0, 1]$

(d) SDR (dB) (e) PESQ $[-0.5, 4.5]$ (f) STOI $[0, 1]$

Fig. 4: Performance comparison of different VAE architectures for speech enhancement. Top row shows the averaged results in terms of input noise levels, whereas the bottom row reports the averaged results versus different noise types. Here, some uniform noise was added to the input of the audio-encoders in MIN-VAE-v1 and MIN-VAE-v2 during training.

high noise levels. As explained earlier, this might be due to the facts that the proposed networks efficiently make use of the robust initialization provided by the visual data, and also by the richer generative models (decoders) which are trained using both audio and visual latent codes. At high noise levels, MIN-VAE-v1 outperforms MIN-VAE-v2, implying the importance of using visual modality in the decoder when the input speech is very noisy. A related observation is that, MIN-VAE-v2 outperforms both MIN-VAE-v1 and AV-VAE when the level of noise is low, implying that the visual features in the generative model contribute mainly in high noise regimes. The average posterior probability of the α_n variable given in (22) is 0.96, implying that the contribution of the audio encoder in generating the latent code is 96%. Part of the worse performance of AV-VAE could be explained by the way the latent codes are initialized, which is based on concatenation of noisy audio and clean visual data. It is worth mentioning that in the low noise regime, the amount of performance improvement is decreasing for all the methods, as the speech signals are already clean enough.

Regarding noise type, we see that the algorithms perform very differently. The *Babble* noise is the most difficult noise environment according to the bottom row of Fig. 3. In terms of SDR, all the methods show their best performance for the *Car* noise, with a very large improvement achieved by the audio-visual based methods. In terms of PESQ, the *White* noise is the easiest one for all the methods, especially MIN-VAE-v3 that shows the best performance. Finally, in terms of STOI, MIN-VAE-v2 achieves the best performance for the *Car* noise.

To encourage the proposed MIN-VAE networks to make use of the visual data in the encoder more efficiently, we added some uniformly distributed noise, with the SNR being about 0 dB and fixed during training, to about one-third of speech spectrogram time frames that are fed to the audio encoders. We also added noise to the audio encoder's inputs of A-VAE and AV-VAE. However, no performance improvements were observed. The amount of noise and percentage of noisy frames have been found empirically. Figure 4 presents the results of this experiment. A clear performance improvement is observed compared to Fig. 3, especially for MIN-AVE-v2. With this new training, the proposed algorithms outperform AV-VAE in all noise levels. The SDR improvements for high noise levels, however, are very close. Regarding the improvement margin, we see that on average, MIN-VAE-v2 outperforms AV-VAE, about 1dB in terms of SDR (at high SNRs), more than 0.1 in terms of PESQ, and about 0.03-0.04 in terms of STOI. The PESQ and STOI improvements remain almost stable for different values of SNR. Overall, even if different MIN-VAE strategies may obtain the best improvement over AV-VAE depending on the measure and on the SNR, the superiority of MIN-VAE w.r.t. the AV-VAE is clear. Finally, the best performing algorithm turns out to be MIN-VAE-v2, outperforming MIN-VAE-v3, especially at low levels of noise. The average posterior probability of the α_n variable given in (22) is now 0.80, which is the best-performing value according to our experiments. Some audio examples are available at https://team.inria.fr/perception/research/min-vae-se/.

V. Conclusions

Inspired by the importance of latent variable initialization for VAE-based speech enhancement, and as another way than simple concatenation to effectively fuse audio and visual modalities in the encoder of VAE, we proposed a mixture of inference (audio and visual encoders) networks, which are jointly trained with a shared generative network. The overall architecture is named MIN-VAE. A variational inference approach was proposed to estimate the parameters of the model. At test phase of the speech enhancement, the initialization of the latent variables, as required by the MCEM inference method, is based on the visual modality, which is assumed to be clean in contrast to audio data. As such, it provides a better performance than initializing with noisy audio data. This is confirmed by our experiments, comparing different VAE architectures.

For future works, dynamical VAE architectures [40] will be investigated, which take into account the temporal correlation between audio and visual frames. This is expected to better handle visual modality, thus achieving superior performance compared to audio-only VAE. Furthermore, we will consider robustifying the proposed algorithms to noisy visual data, e.g. by using the mixture idea dveloped in [32]. Finally, reducing the computational complexity of the inference will be another promising research direction.

Appendix A
KL divergence computation

The KL divergence between two Gaussian distributions is given by the following lemma:

Lemma 1. *Let $p_1(\mathbf{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)$ and $p_2(\mathbf{x};\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$ be two multivariate Gaussian distributions in $\mathbb{R}^n$. Then, the KL divergence between p_1 and p_2 reads:*

$$\mathcal{D}_{KL}(p_1\|p_2)=\frac{1}{2}\Big(\log\frac{\det\boldsymbol{\Sigma}_2}{\det\boldsymbol{\Sigma}_1}-n+\mathrm{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1)+(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\Big). \quad (29)$$

Utilizing the above lemma, we can write the KL divergence term in (21) (for $\alpha_n=1$) as follows:

$$\mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n;\boldsymbol{\phi}_a)\parallel p(\mathbf{z}_n|\alpha_n)\Big)=\frac{1}{2}\log\frac{\sigma_a^L}{\Big|\mathrm{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big|}-+\frac{\mathrm{trace}\Big(\mathrm{diag}\Big(\boldsymbol{\sigma}_z^a(\mathbf{s}_n)\Big)\Big)+\|\boldsymbol{\mu}_z^a(\mathbf{s}_n)-\boldsymbol{\mu}_a\|^2}{2\sigma_a}-\frac{L}{2}, \quad (30)$$

and analogously for the vision-based term ($\alpha_n=0$).

Appendix B
Speech enhancement

The generative model is given in (9) – (11), where all the parameters except π have already been trained on clean audio and visual data. The observations are noisy STFT frames $\mathbf{x}=\{\mathbf{x}_n\}_{n=0}^{N-1}$, as well as the visual data $\mathbf{v}=\{\mathbf{v}_n\}_{n=0}^{N-1}$. The latent variables of the model are $\mathbf{s}=\{\mathbf{s}_n\}_{n=0}^{N-1}$, $\mathbf{z}=\{\mathbf{z}_n\}_{n=0}^{N-1}$, and $\boldsymbol{\alpha}=\{\alpha_n\}_{n=0}^{N-1}$. Furthermore, the parameters of the model are $\Theta=\{\mathbf{W},\mathbf{H},\pi\}$.

A. Parameters Estimation

The full posterior is written as:

$$p(\mathbf{s}_n,\mathbf{z}_n,\alpha_n|\mathbf{x}_n;\mathbf{v}_n,\Theta)\propto p(\mathbf{x}_n,\mathbf{s}_n,\mathbf{z}_n,\alpha_n;\mathbf{v}_n,\Theta)=p(\mathbf{x}_n|\mathbf{s}_n;\Theta)\times p(\mathbf{s}_n|\mathbf{z}_n;\mathbf{v}_n)\times p(\mathbf{z}_n|\alpha_n)\times p(\alpha_n) \quad (31)$$

To estimate the parameter set, we develop a variational inference method [33], where in the variational expectation step (VE-step), the above intractable posterior is approximated by a variational distribution $r(\mathbf{s}_n,\mathbf{z}_n,\alpha_n)$, as similarly done in [31]. The maximization step (M-step) performs parameters update using the obtained variational distributions. We assume that r factorizes as follows:

$$r(\mathbf{s}_n,\mathbf{z}_n,\alpha_n)=r(\mathbf{s}_n)\times r(\mathbf{z}_n)\times r(\alpha_n). \quad (32)$$

where for notational convenience, we have omitted the dependence on Θ. Denoting the current estimate of the parameters as Θ^{old}, the VEM approach consists of iterating between the VE-steps and the M-step, which are detailed below.

1) VE-$r(\mathbf{s}_n)$ step: The variational distribution of $\mathbf{s}_n$ is computed as [33]:

$$r(\mathbf{s}_n)\propto\exp\Big(\mathbb{E}_{r(\mathbf{z}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n,\mathbf{s}_n,\mathbf{z}_n,\alpha_n;\mathbf{v}_n,\Theta^{old})\Big]\Big)=\exp\Big(-\sum_f\Big[\frac{|x_{fn}-s_{fn}|^2}{(\mathbf{WH})_{fn}}+\frac{|s_{fn}|^2}{\gamma_{fn}}\Big]\Big), \quad (33)$$

where,

$$\gamma_{fn}^{-1}=\mathbb{E}_{r(\mathbf{z}_n)}\left[\frac{1}{\sigma_{s,f}(\mathbf{z}_n^{(d)},\mathbf{v}_n)}\right]\approx\frac{1}{D}\sum_{d=1}^{D}\frac{1}{\sigma_{s,f}(\mathbf{z}_n^{(d)},\mathbf{v}_n)}, \quad (34)$$

and $\Big\{\mathbf{z}_n^{(d)}\Big\}_{d=1}^{D}$ is a sequence sampled from $r(\mathbf{z}_n)$. From (33), we can see that $r(s_{fn})=\mathcal{N}_c(m_{fn},\nu_{fn})$, where:

$$\begin{cases}m_{fn} &= \frac{\gamma_{fn}}{\gamma_{fn}+(\mathbf{WH})_{fn}}\cdot x_{fn}\\ \nu_{fn} &= \frac{\gamma_{fn}\cdot(\mathbf{WH})_{fn}}{\gamma_{fn}+(\mathbf{WH})_{fn}}\end{cases}. \quad (35)$$

2) VE-$r(\mathbf{z}_n)$ step: The variational distribution of $\mathbf{z}_n$ can be computed by the following standard formula:

$$\begin{aligned}r(\mathbf{z}_n)&\propto\exp\Big(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n,\mathbf{s}_n,\mathbf{z}_n,\alpha_n;\mathbf{v}_n,\Theta^{old})\Big]\Big)\\&\propto\exp\Big(\sum_f-\log\Big(\sigma_{s,f}(\mathbf{z}_n,\mathbf{v}_n)\Big)-\frac{|m_{fn}|^2+\nu_{fn}}{\sigma_{s,f}(\mathbf{z}_n,\mathbf{v}_n)}\\&\quad+\sum_{\alpha_n\in\{0,1\}}r(\alpha_n)\cdot\Big[\log p(\mathbf{z}_n|\alpha_n)\Big]\Big)\triangleq\tilde{r}(\mathbf{z}_n)\end{aligned} \quad (36)$$

This gives us an unnormalized distribution $\tilde{r}(\mathbf{z}_n)$ whose normalization constant cannot be computed in closed-from, due to the non-linear terms. However, we use the Metropolis-Hastings algorithm [33] to sample from it. To that end, we need to start with an initialization, $\mathbf{z}^{(0)}$. At the beginning of the inference, $\mathbf{z}^{(0)}$ is set to be the posterior mean in the output of the visual-encoder, i.e. the bottom-left network in Fig. 2, where $\mathbf{v}_n$ is given as the input. Then, a candidate sample denoted $\mathbf{z}^{(c)}$

is obtained by sampling from a proposal distribution, usually chosen to be a Gaussian:

$$\mathbf{z}^{(c)}|\mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{z}^{(0)}, \epsilon\mathbf{I}), \tag{37}$$

where, $\epsilon > 0$ controls the speed of convergence. Then, $\mathbf{z}^{(c)}$ is set to be the next sample $\mathbf{z}^{(1)}$ with the following probability:

$$p = \min\Big(1, \frac{\tilde{r}(\mathbf{z}^{(c)})}{\tilde{r}(\mathbf{z}^{(0)})}\Big). \tag{38}$$

That means, some u is drawn from a uniform distribution between 0 and 1. Then, if $u < p$, the sample is accepted and $\mathbf{z}^{(1)} = \mathbf{z}^{(c)}$. Otherwise, it is rejected and $\mathbf{z}^{(1)} = \mathbf{z}^{(0)}$. This procedure is repeated until the required number of samples is achieved. The first few samples are usually discarded, as they are not so reliable.

3) VE-$r(\alpha_n)$ step: The variational distribution of α_n is computed as:

$$\begin{aligned} r(\alpha_n) &\propto \exp\Big(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n; \mathbf{v}_n, \Theta^{old})\Big]\Big) \\ &\propto p(\alpha_n) \times \exp\Big(\mathbb{E}_{r(\mathbf{z}_n)}\Big[\alpha_n \cdot \log p(\mathbf{z}_n|\alpha_n = 1) \\ &\quad + (1-\alpha_n)\cdot \log p(\mathbf{z}_n|\alpha_n = 0)\Big]\Big) \end{aligned} \tag{39}$$

which is a Bernoulli distribution with the following parameter:

$$\pi_n = g\Big(\mathbb{E}_{r(\mathbf{z}_n)}\Big[\log\frac{p(\mathbf{z}_n|\alpha_n=1)}{p(\mathbf{z}_n|\alpha_n=0)}\Big] + \log\frac{\pi}{1-\pi}\Big), \tag{40}$$

with $g(.)$ being the sigmoid function.

4) M-step: After updating all the variational distributions, the next step is to update the set of parameters, i.e. $\Theta = \{\mathbf{W}, \mathbf{H}, \pi\}$. To do so, we need to optimize the complete-data log-likelihood which reads:

$$\begin{aligned} Q(\Theta; \Theta^{old}) &= \mathbb{E}_{r(\mathbf{s})\cdot r(\mathbf{z})\cdot r(\boldsymbol{\alpha})}\Big[\log p(\mathbf{x}, \mathbf{s}, \mathbf{z}, \boldsymbol{\alpha}; \mathbf{v}, \Theta)\Big] \\ &\overset{cte.}{=} \sum_{f,n} -\frac{|x_{fn} - m_{fn}|^2 + \nu_{fn}}{(\mathbf{WH})_{fn}} - \log(\mathbf{WH})_{fn} \\ &\quad + \pi_n \log\pi + (1-\pi_n)\log(1-\pi) \end{aligned} \tag{41}$$

The update formulas for $\mathbf{W}$ and $\mathbf{H}$ can be obtained by using standard multiplicative updates [41]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top\Big(\mathbf{V}\odot(\mathbf{WH})^{\odot -2}\Big)}{\mathbf{W}^\top(\mathbf{WH})^{\odot -1}}, \tag{42}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\Big(\mathbf{V}\odot(\mathbf{WH})^{\odot -2}\Big)\mathbf{H}^\top}{(\mathbf{WH})^{\odot -1}\mathbf{H}^\top}, \tag{43}$$

where $\mathbf{V} = \Big[|x_{fn} - m_{fn}|^2 + \nu_{fn}\Big]_{(f,n)}$, and $\odot$ denotes element-wise operation. Optimizing over π leads to a similar update formula as in (25):

$$\pi = \frac{1}{N}\sum_{n=1}^{N}\pi_n. \tag{44}$$

B. Speech Estimation

Let $\Theta^* = \{\mathbf{W}^*, \mathbf{H}^*, \pi^*\}$ denote the optimal set of parameters found by the above VEM procedure. An estimation of the clean speech is then obtained as the variational posterior mean ($\forall f, n$):

$$\hat{s}_{fn} = \mathbb{E}_{r(s_{fn})}[s_{fn}] = \frac{\gamma^*_{fn}}{\gamma^*_{fn} + (\mathbf{W}^*\mathbf{H}^*)_{fn}}\cdot x_{fn}, \tag{45}$$

where, γ^*_{fn}, defined in (34), is computed using the optimal parameters.

REFERENCES

[1] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*, Wiley, 2018.

[2] P. C. Loizou, *Speech enhancement: theory and practice*, CRC press, 2007.

[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[4] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[5] W. DeLiang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[6] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[7] X. Li and R. Horaud, "Multichannel speech enhancement based on time-frequency masking using subband long short-term memory," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, pp. 1–5.

[8] K. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, 2008, pp. 4029–4032.

[9] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[10] F. Sedighin, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Multimodal soft nonnegative matrix co-factorization for convolutive source separation," vol. 65, no. 12, pp. 3179–3190, 2017.

[11] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[12] F. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proc. Int. Conf. Indep. Component Analysis and Signal Separation*, 2007, pp. 414–421.

[13] G. J. Mysore and P. Smaragdis, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[14] M. Sun, X. Zhang, and T. F. Zheng, "Unseen noise estimation using separable deep auto encoder for speech enhancement," vol. 24, no. 1, pp. 93–104, 2016.

[15] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 716–720.

[16] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.

[17] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian multichannel speech enhancement with a deep speech prior," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 1233–1239.

[18] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.

[19] S. Leglaive, U. Şimşekli, A. Liutkus, L. Girin, and R. Horaud, "Speech enhancement with variational autoencoders and alpha-stable distributions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 541–545.

[20] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Computation*, vol. 31, no. 9, pp. 1–24, 2019.

[21] V. Nhat Nguyen, M. Sadeghi, E. Ricci, and X. Alameda-Pineda, "Deep variational generative models for audio-visual speech separation," *arXiv preprint arXiv:2008.07191*, 2020.

[22] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[24] Jan Cech, Ravi Mittal, Antoine Deleforge, Jordi Sanchez-Riera, Xavier Alameda-Pineda, and Radu Horaud, "Active-speaker detection and localization with microphones and cameras embedded into a robotic head," in *IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 203–210.

[25] Yutong Ban, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud, "Variational bayesian inference for audio-visual tracking of multiple speakers," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[26] Yutong Ban, Xavier Alameda-Pineda, Fabien Badeig, Sileye Ba, and Radu Horaud, "Tracking a varying number of people with a visually-controlled robotic head," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4144–4151.

[27] L. Girin, J.-L. Schwartz, and G. Feng, "Audio-visual enhancement of speech in noise," *The Journal of the Acoustical Society of America*, vol. 109, no. 6, pp. 3007–3020, 2001.

[28] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3244–3248.

[29] A. Gabbay, A. Shamir, and S. Peleg, "Visual speech enhancement," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1170–1174.

[30] M. Sadeghi, S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "Audio-visual speech enhancement using conditional variational auto-encoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 1788–1800, 2020.

[31] M. Sadeghi and X. Alameda-Pineda, "Robust unsupervised audio-visual speech enhancement using a mixture of variational autoencoders," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[32] M. Sadeghi and X. Alameda-Pineda, "Switching variational auto-encoders for noise-agnostic audio-visual speech enhancement," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

[33] C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag Berlin, Heidelberg, 2006.

[34] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[35] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 749–752.

[36] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[37] A.-H. Abdelaziz, "NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 3752–3756.

[38] M. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic phonetic continuous speech corpus," in *Linguistic data consortium*, 1993.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[40] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[41] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.



Mostafa Sadeghi received the B.Sc. degree from Ferdowsi University of Mashhad, Iran, in 2010, the M.Sc. degree from Sharif University of Technology, Tehran, Iran, in 2012, and the Ph.D. degree from the same university in 2018, all in electrical engineering. From August 2018 to October 2020 he was a post-doctoral researcher in the Perception team at Inria Grenoble Rhône-Alpes. Currently, he is a researcher in the Multispeech team at Inria Nancy - Grand Est. His main research interests are latent variable generative models, unsupervised audio-visual speech enhancement, Bayesian inference and probabilistic machine learning, and local/global optimization.



Xavier Alameda-Pineda is a (tenured) Research Scientist at Inria, in the Perception Group. He obtained the M.Sc. (equivalent) in Mathematics in 2008, in Telecommunications in 2009 from BarcelonaTech and in Computer Science in 2010 from Universit Grenoble-Alpes (UGA). He the worked towards his Ph.D. in Mathematics and Computer Science, and obtained it 2013, from UGA. After a two-year post-doc period at the Multimodal Human Understanding Group, at University of Trento, he was appointed with his current position. Xavier is an active member of SIGMM, and a senior member of IEEE and a member of ELLIS. He is co-chairing the Audio-visual machine perception and interaction for companion robots chair of the Multidisciplinary Institute of Artificial Intelligence. Xavier is the Coordinator of the H2020 Project SPRING: Socially Pertinent Robots in Gerontological Healthcare. Xaviers research interests are in combining machine learning, computer vision and audio processing for scene and behavior analysis and human-robot interaction.