Unsupervised Audio-visual Speech Enhancement based on Variational Autoencoders

Mostafa Sadeghi

*Joint work with Simon Leglaive, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud

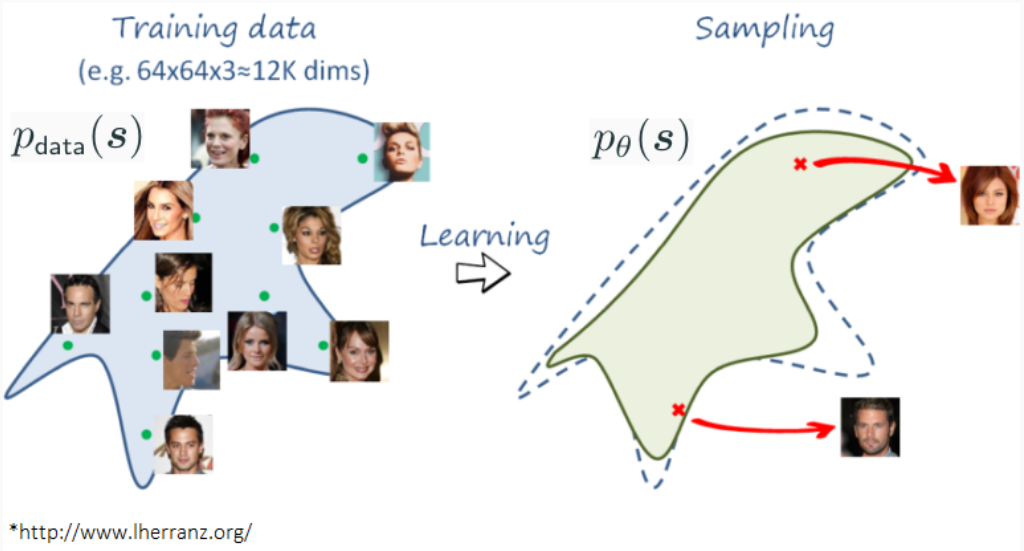
Perception team, Inria Grenoble Rhône-Alpes, France

May 28, 2020

1. Deep Latent Variable Generative Models
2. Audio-visual Speech Enhancement
3. Robust Audio-visual Speech Enhancement
4. Mixture of Audio-visual Inference Networks

Deep Latent Variable Generative Models

Generative Models



Objective: Learning/simulating a complicated probability distribution of data, p_{data}, given some training samples: $\boldsymbol{s}_i \sim p_{\text{data}}(\boldsymbol{s}), \quad i = 1, \ldots, N.$

Learn a parametric distribution $p_\theta(\boldsymbol{s})$ as close as possible to $p_{\text{data}}(\boldsymbol{s})$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \; D_{\text{KL}}\Big(p_{\text{data}}(\boldsymbol{s}) \,\|\, p_\theta(\boldsymbol{s})\Big) = \underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{p_{\text{data}}}\Big[\log \frac{p_{\text{data}}(\boldsymbol{s})}{p_\theta(\boldsymbol{s})}\Big]$$

$$= \underset{\theta}{\operatorname{argmax}} \; \mathbb{E}_{p_{\text{data}}}\Big[\log p_\theta(\boldsymbol{s})\Big] \approx \boxed{\underset{\theta}{\operatorname{argmax}} \; \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(\boldsymbol{s}_i)}$$

Latent Variable Generative Models

Latent variable models provide a flexible and hierarchical way to approximate probability distribution of data:

- $\boldsymbol{s} \in \mathbb{R}^n$: observed variable
- $\boldsymbol{z} \in \mathbb{R}^\ell$: latent variable, a concise representation of $\boldsymbol{s}$ ($\ell \ll n$)

$$\begin{cases} \boldsymbol{z} \sim p_\theta(\boldsymbol{z}) \\ \boldsymbol{s}|\boldsymbol{z} \sim p_\theta(\boldsymbol{s}|\boldsymbol{z}) \end{cases} \rightarrow \; p_\theta(\boldsymbol{s}) = \int p_\theta(\boldsymbol{s}, \boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int p_\theta(\boldsymbol{s}|\boldsymbol{z}) p_\theta(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}$$

Generating new samples:

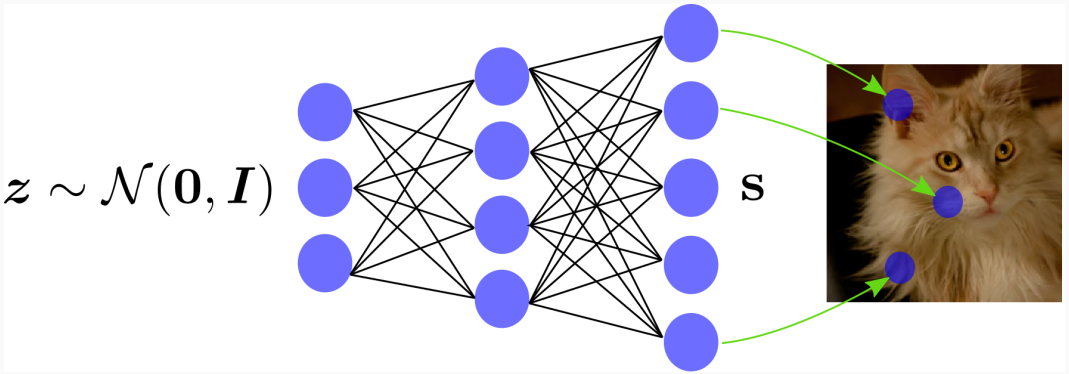
Draw $\boldsymbol{z}_k \sim p_\theta(\boldsymbol{z})$, then draw a new sample $\boldsymbol{s}_k \sim p_\theta(\boldsymbol{s}|\boldsymbol{z}_k)$

Latent Variable Generative Models

Usually, $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and $p_\theta(\boldsymbol{s}|\boldsymbol{z})$ is a parametric Gaussian distribution:

- Linear Gaussian model: $p_\theta(\boldsymbol{s}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{A}\boldsymbol{z} + \boldsymbol{\mu}, \sigma\boldsymbol{I})$
  - $\theta = \{\boldsymbol{A}, \boldsymbol{\mu}, \sigma\}$
  - Parameter estimation is straightforward
  - Not expressive enough

- Non-linear Gaussian model: $p_\theta(\boldsymbol{s}|\boldsymbol{z}) = \mathcal{N}\Big(\boldsymbol{\mu}_\theta(\boldsymbol{z}), \boldsymbol{\Sigma}_\theta(\boldsymbol{z})\Big)$
  - $\boldsymbol{\mu}_\theta(.), \boldsymbol{\Sigma}_\theta(.)$: Non-linear functions implemented as deep neural networks
  - Parameter estimation is challenging
  - Often expressive enough



Parameter Estimation

Expectation-Maximization (EM)

$$\begin{aligned}\log p_\theta(\boldsymbol{s}) &= \mathbb{E}_{p_\theta(\boldsymbol{z}|\boldsymbol{s})}\Big[\log p_\theta(\boldsymbol{s})\Big] \\ &= \mathbb{E}_{p_\theta(\boldsymbol{z}|\boldsymbol{s})}\Big[\log \frac{p_\theta(\boldsymbol{s},\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{s})}\Big] \\ &= \mathbb{E}_{p_\theta(\boldsymbol{z}|\boldsymbol{s})}\Big[\log p_\theta(\boldsymbol{s},\boldsymbol{z})\Big] - \mathbb{E}_{p_\theta(\boldsymbol{z}|\boldsymbol{s})}\Big[\log p_\theta(\boldsymbol{z}|\boldsymbol{s})\Big]\end{aligned}$$

- E-step: Update the posterior of $\boldsymbol{z}_i$, $\forall i$:

$$p_{\theta^{old}}(\boldsymbol{z}_i|\boldsymbol{s}_i) = \frac{p_{\theta^{old}}(\boldsymbol{s}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)}{p_{\theta^{old}}(\boldsymbol{s}_i)}$$

- M-step: Update the parameters:

$$\theta^{new} = \arg\max_\theta \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p_{\theta^{old}}(\boldsymbol{z}_i|\boldsymbol{s}_i)}\Big[\log p_\theta(\boldsymbol{s}_i,\boldsymbol{z}_i)\Big]$$

Parameter Estimation: Variational Inference

Variational Inference (VI)

- VI seeks an approximate density $q(\boldsymbol{z}|\boldsymbol{s})$ by minimizing a measure of dissimilarity:

$$D_{\mathsf{KL}}\Big(q(\boldsymbol{z}|\boldsymbol{s}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{s})\Big) = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{s})}\Big[\log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{s})}{q(\boldsymbol{z}|\boldsymbol{s})}\Big]$$
$$= -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{s})}\Big[\log \frac{p_\theta(\boldsymbol{s},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{s})}\Big] + \log p_\theta(\boldsymbol{s}) \geq 0$$

- Evidence Lower-bound (ELBO):

$$\text{Evidence} = \log p_\theta(\boldsymbol{s}) \geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{s})}\Big[\log \frac{p_\theta(\boldsymbol{s},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{s})}\Big] \triangleq \mathcal{F}(q,\theta)$$

- Optimize the ELBO:

$$q^* = \arg\max_{q\in\mathcal{C}} \ \mathcal{F}(q,\theta)$$

Parameter Estimation: Variational Inference (continued)

In some cases, the posterior can be computed in closed-from, and $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{s})$. As such, the KL term is zero and $\mathcal{F}(q, \theta) = \log p_\theta(\boldsymbol{s})$.

- Variational E-step: For $i = 1, \ldots, N$,

$$q_i^{new} = \arg\max_{q_i \in \mathcal{C}} \frac{1}{N} \sum_{j=1}^{N} \mathcal{F}_j(q_j, \theta^{old})$$

- M-step: Update the parameters:

$$\theta^{new} = \arg\max_{\theta} \frac{1}{N} \sum_{j=1}^{N} \mathcal{F}_j(q_j^{new}, \theta)$$

Parameter Estimation: MCEM

Monte Carlo Expectation-Maximization (MCEM)

- E-step: Sample $\{\boldsymbol{z}_i^{(r)}\}_{r=1}^{R}$ from the posterior of $\boldsymbol{z}_i$, $\forall i$:

$$p_{\theta^{old}}(\boldsymbol{z}_i|\boldsymbol{s}_i) \propto p_{\theta^{old}}(\boldsymbol{s}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)$$

- M-step: Update the parameters:

$$\begin{aligned}\theta^{new} &= \arg\max_{\theta} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p_{\theta^{old}}(\boldsymbol{z}_i|\boldsymbol{s}_i)}\Big[\log p_\theta(\boldsymbol{s}_i, \boldsymbol{z}_i)\Big] \\ &\approx \arg\max_{\theta} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{R}\sum_{r=1}^{R} \log p_\theta(\boldsymbol{s}_i, \boldsymbol{z}_i^{(r)})\end{aligned}$$

- Given infinite computational resources, it yields accurate results
- Computationally expensive and not scalable to high-dimensions

Parameter Estimation: Variational Autoencoder

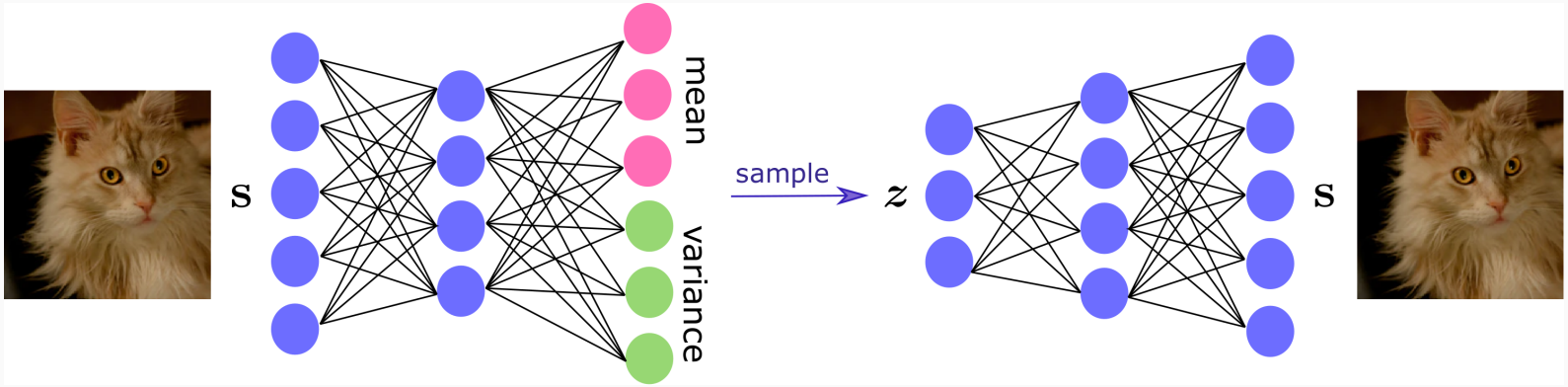
Variational Autoencoder (VAE)

Recall the generative model: $\begin{cases} p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ p_\theta(\boldsymbol{s}|\boldsymbol{z}) = \mathcal{N}\Big(\boldsymbol{\mu}_\theta(\boldsymbol{z}), \boldsymbol{\Sigma}_\theta(\boldsymbol{z})\Big) \end{cases}$

VAE approximates $p_\theta(\boldsymbol{z}|\boldsymbol{s})$ with a *parametric* Gaussian distribution:

$$q_\psi(\boldsymbol{z}|\boldsymbol{s}) = \mathcal{N}\Big(\boldsymbol{\mu}_\psi(\boldsymbol{s}), \boldsymbol{\Sigma}_\psi(\boldsymbol{s})\Big)$$

where, $\boldsymbol{\mu}_\psi(.)$ and $\boldsymbol{\Sigma}_\psi(.)$ are non-linear functions implemented as *deep neural networks* with parameters ψ. $\boldsymbol{\Sigma}_\psi(.)$ is diagonal.



Parameter Estimation: Variational Autoencoder (continued)

The set of parameters $\{\theta, \psi\}$ are estimated by maximizing the ELBO:

$$\begin{aligned}
\mathcal{F}(\theta, \psi) &= \mathbb{E}_{q_\psi(\boldsymbol{z}|\boldsymbol{s})}\left[\log \frac{p_\theta(\boldsymbol{s}, \boldsymbol{z})}{q_\psi(\boldsymbol{z}|\boldsymbol{s})}\right] \\
&= \underbrace{\mathbb{E}_{q_\psi(\boldsymbol{z}|\boldsymbol{s})}\Big[\log p_\theta(\boldsymbol{s}|\boldsymbol{z})\Big]}_{\text{Reconstruction term}} - \underbrace{D_{\mathsf{KL}}\Big(q_\psi(\boldsymbol{z}|\boldsymbol{s}) \parallel p(\boldsymbol{z})\Big)}_{\text{Regularization term}} \\
&\approx \log p_\theta(\boldsymbol{s}|\boldsymbol{z}_\psi) - D_{\mathsf{KL}}\Big(q_\psi(\boldsymbol{z}|\boldsymbol{s}) \parallel p(\boldsymbol{z})\Big)
\end{aligned}$$

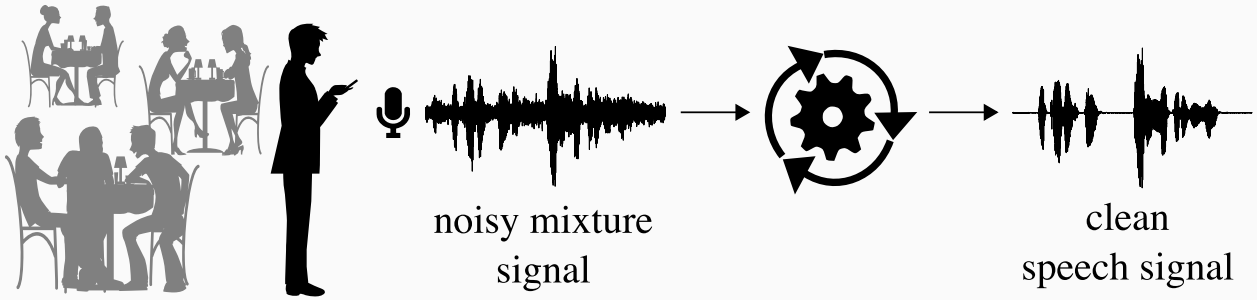
where, $\boldsymbol{z}_\psi = \boldsymbol{\mu}_\psi(\boldsymbol{s}) + \boldsymbol{\epsilon} \odot \mathsf{diag}(\boldsymbol{\Sigma}_\psi(\boldsymbol{s})) \sim q_\psi(\boldsymbol{z}|\boldsymbol{s}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

Stochastic gradient descent:

$$\theta^{(new)}, \psi^{(new)} = \arg\max_{\theta,\psi} \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(\boldsymbol{s}|\boldsymbol{z}_\psi^i) - D_{\mathsf{KL}}\Big(q_\psi(\boldsymbol{z}_i|\boldsymbol{s}) \parallel p(\boldsymbol{z}_i)\Big)$$

Audio-visual Speech Enhancement

Speech Enhancement



Remove the background noise from the observed mixture speech.

In the short-time Fourier transform (STFT) domain, for all $(f, n) \in \mathbb{B} = \{0, ..., F-1\} \times \{0, ..., N-1\}$, we observe:

$$x_{fn} = s_{fn} + b_{fn} \qquad \text{vectorized representation:} \qquad \mathbf{x}_n = \mathbf{s}_n + \mathbf{b}_n$$

- s_{fn} corresponds to clean speech signal.
- b_{fn} corresponds to noise signal.
- f is the frequency index and n the time-frame index.

Audio-visual Speech Enhancement (AVSE)

- Visual data, i.e. lips movements, provides some complementary information about the unknown speech.
- For highly noisy audio recordings, visual information can be very helpful.

We aim to efficiently fuse audio and visual modalities for speech enhancement.

Supervised/Unsupervised AVSE

Supervised: Learn a mapping from clean visual data and noisy audio data to clean audio data:

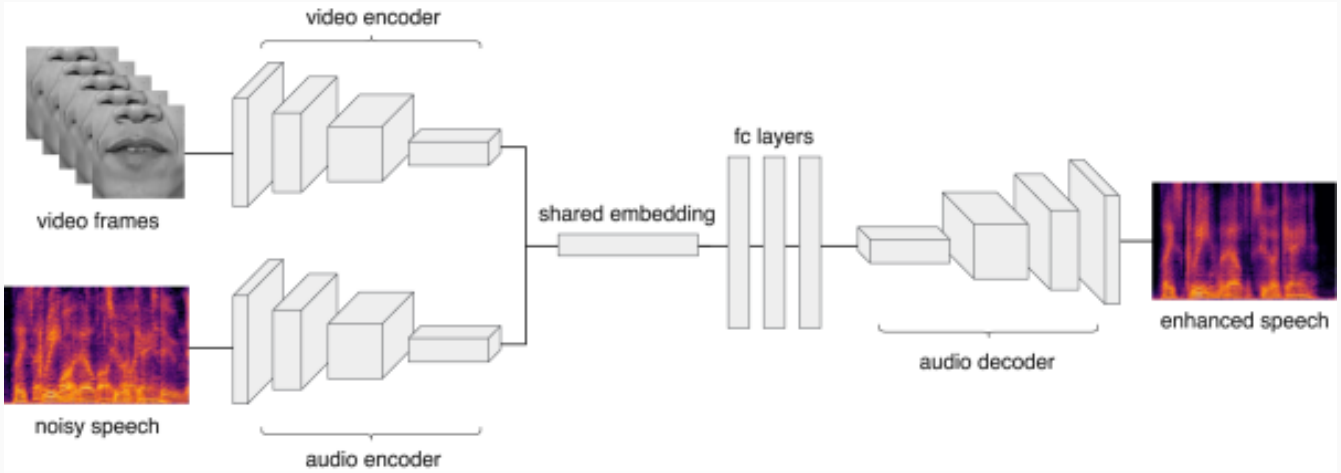


Figure 1: Supervised audio-visual speech enhancement [Gabbby et al., 2018]

Unsupervised (our work): Learn a generative audio-visual model for clean speech and combine it with an unsupervised noise model at test time

Audio-only VAE [Bando et al., 2018; Leglaive et al., 2018]

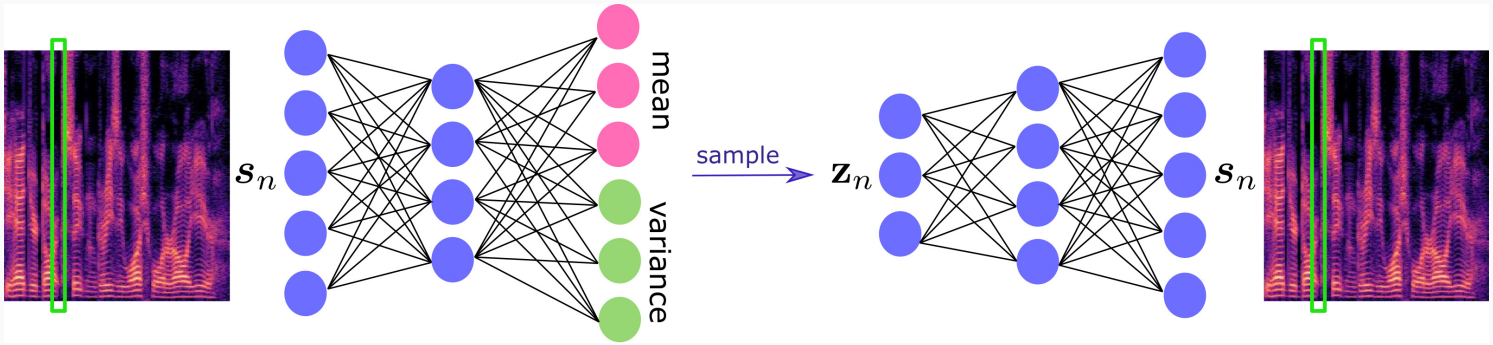
Generative model (Decoder):

Each clean spectrogram time frame $\boldsymbol{s}_n$ is assumed to be generated as:

$$\boldsymbol{s}_n|\mathbf{z}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_s(\mathbf{z}_n))\Big), \qquad p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Inference network (Encoder):

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



Video-only VAE [Sadeghi et al., 2019]

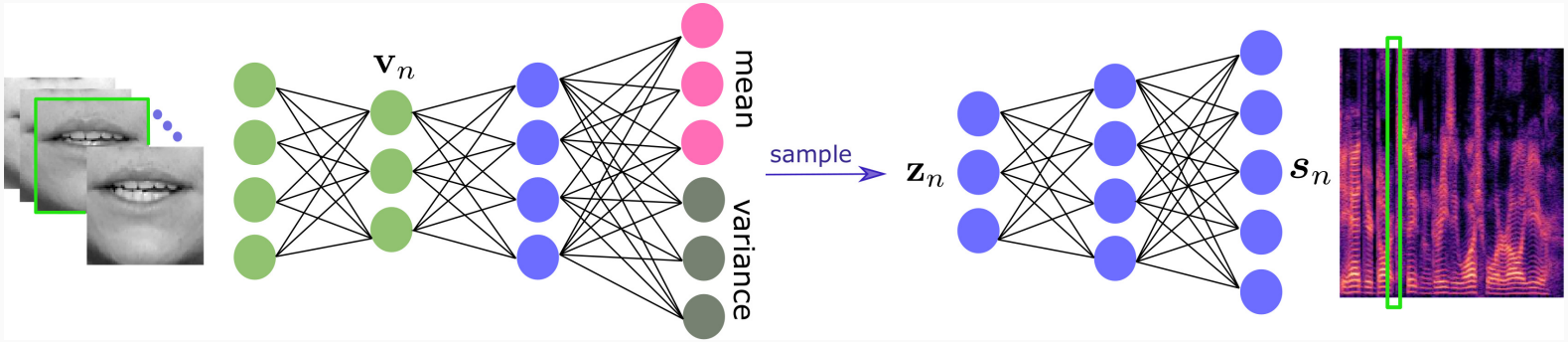
Generative model (Decoder)[1]:

$$\boldsymbol{s}_n|\mathbf{z}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_s(\mathbf{z}_n))\Big), \qquad p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Inference network (Encoder):

Infer the posterior using visual data only:

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



[1] M. Sadeghi et al., "Audio-visual Speech Enhancement Using Conditional Variational Auto-Encoder," [Available Online] https://arxiv.org/abs/1908.02590, August 2019.

Audio-visual VAE [Sadeghi et al., 2019]

Inspired by conditional VAE (CVAE), use visual data as some deterministic information

Generative network (Decoder):

Each clean spectrogram time frame $\boldsymbol{s}_n$ is assumed to be generated as:

$$\begin{cases} \boldsymbol{s}_n|\mathbf{z}_n, \mathbf{v}_n & \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n))\Big) \\ \mathbf{z}_n|\mathbf{v}_n & \sim \mathcal{N}\Big(\boldsymbol{\mu}_z(\boldsymbol{v}_n), \text{diag}(\boldsymbol{\sigma}_z(\mathbf{v}_n))\Big) \end{cases}$$

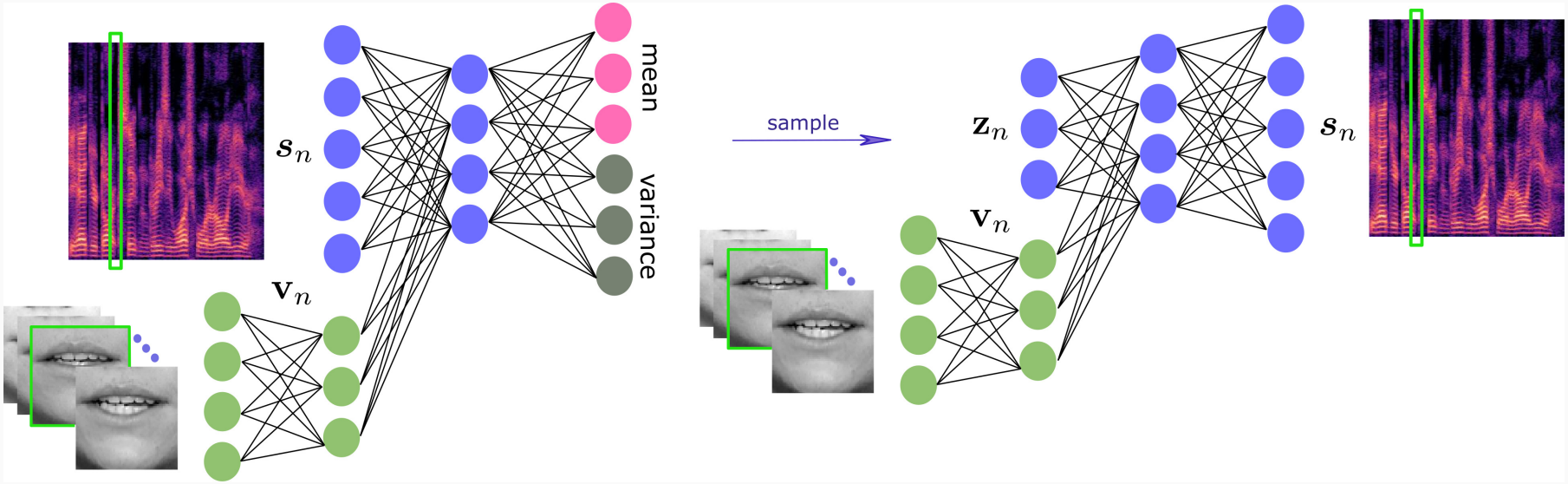
▷ $\mathbf{v}_n$ is an embedding for the image of the speaker lips at frame n.

Inference network (Encoder):

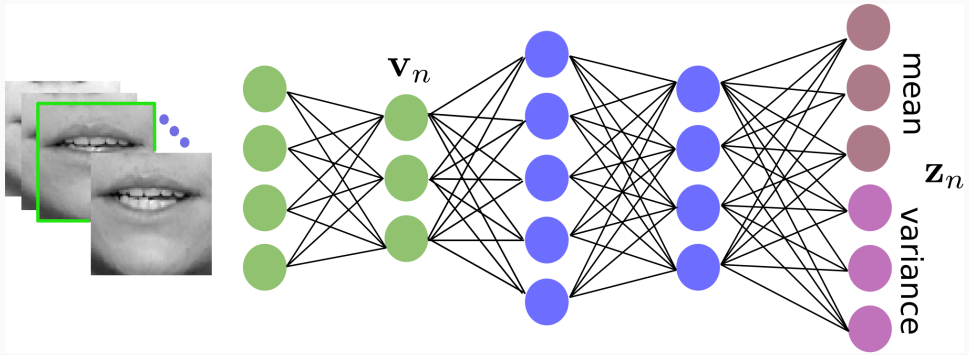
$$q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n; \boldsymbol{\psi}) = \mathcal{N}\Big(\boldsymbol{\mu}_z^{av}(\mathbf{s}_n, \boldsymbol{v}_n), \text{diag}(\boldsymbol{\sigma}_z^{av}(\mathbf{s}_n, \mathbf{v}_n))\Big)$$

Audio-visual VAE

Encoder and decoder networks:



Prior distribution for latent variables:



Training AV-VAE

Training samples:

- Clean speech spectrogram time frames: $\{\mathbf{s}_n\}_{n=0}^{N_{tr}-1}$
- Associated visual data: $\{\mathbf{v}_n\}_{n=0}^{N_{tr}-1}$

Optimize the ELBO:

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\psi}\right) = \frac{1}{N_{tr}} \sum_{n=0}^{N_{tr}-1} \alpha \cdot \mathbb{E}_{p(\mathbf{z}_n|\mathbf{v}_n)} \Big[\ln p\left(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}\right) \Big] +$$

$$(1-\alpha) \cdot \Big(\mathbb{E}_{q(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n; \boldsymbol{\psi})} \Big[\ln p\left(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}\right) \Big] - D_{\mathsf{KL}} \Big(q\left(\mathbf{z}_n|\mathbf{s}_n, \mathbf{v}_n; \boldsymbol{\psi}\right) \parallel p(\mathbf{z}_n|\mathbf{v}_n) \Big) \Big)$$

- $0 \leq \alpha \leq 1$ gives some reconstruction power to the prior network

Speech Enhancement

Noisy speech model: $\forall n: \quad \boldsymbol{x}_n = \boldsymbol{s}_n + \boldsymbol{b}_n$

Noise model: $\forall n: \quad \boldsymbol{b}_n \sim \mathcal{N}_c\left(\mathbf{0}, \mathrm{diag}(\mathbf{W}_b \mathbf{H}_b[:, n])\right)$

Clean speech model: Trained generative network:

$$\begin{cases} p(\boldsymbol{s}_n|\mathbf{z}_n, \mathbf{v}_n) & = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}\big(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n)\big)\Big) \\ p(\mathbf{z}_n|\mathbf{v}_n) & = \mathcal{N}\Big(\boldsymbol{\mu}_z(\mathbf{v}_n), \mathrm{diag}\big(\boldsymbol{\sigma}_z(\mathbf{v}_n)\big)\Big) \end{cases}$$

Inference:

- Parameters to be estimated: $\boldsymbol{\theta}_u = \{\mathbf{W}_b, \mathbf{H}_b\}$
- Observed variables: $\mathbf{x} = \{\mathbf{x}_n\}_{n=0}^{N-1}, \mathbf{v} = \{\mathbf{v}_n\}_{n=0}^{N-1}$
- Latent variables: $\mathbf{z} = \left\{\mathbf{z}_n\right\}_{n=0}^{N-1}$
- Likelihood:

$$p(\boldsymbol{x}_n|\mathbf{z}_n, \mathbf{v}_n; \boldsymbol{\theta}_u) = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}\big(\boldsymbol{\sigma}_s(\mathbf{z}_n, \mathbf{v}_n)\big) + \mathrm{diag}(\mathbf{W}_b \mathbf{H}_b[:, n])\Big)$$

Parameter Estimation: MCEM

From an initialization $\boldsymbol{\theta}_u^\star$ of the parameters, iterate:

- **E-Step**: $Q(\boldsymbol{\theta}_u; \boldsymbol{\theta}_u^\star) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x},\mathbf{v};\boldsymbol{\theta}_u^\star)}[\ln p(\mathbf{x}, \mathbf{z}, \mathbf{v}; \boldsymbol{\theta}_u)]$.

  Intractable expectation $\rightarrow$ Markov chain Monte Carlo method.

$$Q(\boldsymbol{\theta}_u; \boldsymbol{\theta}_u^\star) \approx \frac{1}{R} \sum_{r=1}^{R} \ln p(\mathbf{x}, \mathbf{z}^{(r)}, \mathbf{v}; \boldsymbol{\theta}_u)$$

  The samples $\left\{\mathbf{z}^{(r)}\right\}_{r=1,\dots,R}$ are i.i.d. and asymptotically drawn from $p(\mathbf{z}|\mathbf{x}, \mathbf{v}; \boldsymbol{\theta}_u^\star)$ using the Metropolis-Hastings method.

  ▷ Note: $\ln p(\mathbf{x}, \mathbf{z}^{(r)}, \mathbf{v}; \boldsymbol{\theta}_u) = \ln p(\mathbf{x}|\mathbf{z}^{(r)}, \mathbf{v}; \boldsymbol{\theta}_u) + \ln p(\mathbf{z}^{(r)}|\mathbf{v})$

- **M-Step**: $\boldsymbol{\theta}_u^\star \leftarrow \operatorname{argmax}_{\boldsymbol{\theta}_u} Q(\boldsymbol{\theta}_u; \boldsymbol{\theta}_u^\star)$.

  Minorize-maximize approach leading to multiplicative update rules.

Speech Estimation

Once the parameters are estimated, the speech STFT frames are estimated via a Wiener-like filtering ($\forall f, n$):

$$\begin{aligned}
\hat{s}_{fn} &= \mathbb{E}_{p(s_{fn}|x_{fn},\mathbf{v}_n;\boldsymbol{\theta}_u^*)}\left[s_{fn}\right] \\
&= \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n,\mathbf{v}_n;\boldsymbol{\theta}_u^*)}\left[\mathbb{E}_{p(s_{fn}|\mathbf{z}_n,\mathbf{v}_n,\mathbf{x}_n;\boldsymbol{\theta}_u^*)}\left[s_{fn}\right]\right] \\
&= \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n,\mathbf{v}_n;\boldsymbol{\theta}_u^*)}\left[\frac{g_n^*\sigma_{s,f}(\mathbf{z}_n,\mathbf{v}_n)}{g_n^*\sigma_{s,f}(\mathbf{z}_n,\mathbf{v}_n)+(\mathbf{W}_b^*\mathbf{H}_b^*)_{f,n}}\right]x_{fn}.
\end{aligned}$$

where, $\boldsymbol{\theta}_u^*$ denotes the set of estimated parameters by the MCEM method.

▷ As before, the intractable posterior is approximated by a Markov chain Monte Carlo method.

Experiments

- NTCD-TIMIT dataset [Abdelaziz, 2017]
  - Audio-visual recordings in controlled conditions
  - Clean audio as well as noisy versions
  - Frontal video frames with 30 FPS- 67×67 lips images
- Training set ($\sim$ 5 hours): 39 speakers $\times$ 98 sentences $\times$ 5 seconds
- Test set ($\sim$ 1 hour): 9 speakers $\times$ 98 sentences $\times$ 5 seconds
- Noise levels: -15 dB, -10 dB, -5 dB, 0 dB, 5 dB and 15 dB
- Noise types: *Living Room (LR)*, *White*, *Cafe*, *Car*, *Babble*, and *Street*

Experiments

Networks architectures:

1. A-VAE:
   - Decoder: Single hidden layer, 128 nodes, hyperbolic tangent activations. Input dimension: 32 (latent space).
   - Encoder: Single hidden layer, 128 nodes, hyperbolic tangent activations. Input dimension: 513 (spectrogram time frame).
2. V-VAE:
   - Decoder: Same as A-VAE.
   - Encoder: Two hidden layers, 512 and 128 nodes, ReLU activations. Input dimension: 4489 (67×67).
3. AV-VAE: Shares the same architecure as that of AV-VAE with visual embeddings being concatenated with the encoder's and decoder's input.

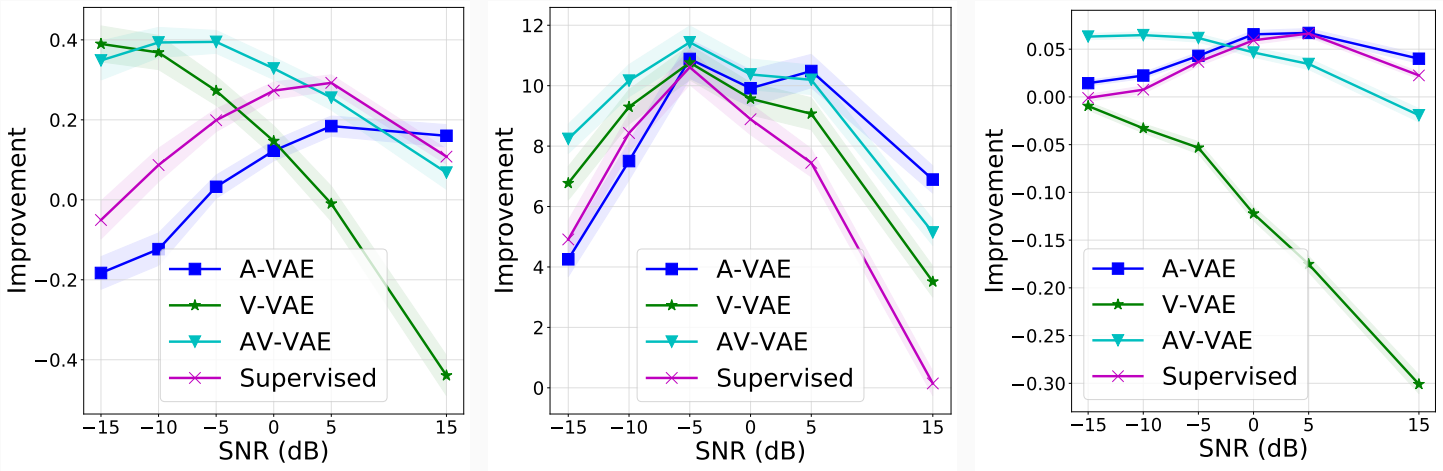
▷ The prior network of the latent variables in AV-VAE takes the same architecture as that of V-VAE's encoder.

Results

Objective measures (the higher, the better)

- Signal-to-distortion ratio (SDR).
- Perceptual evaluation of speech quality (PESQ) measure.
- Short-time objective intelligibility (STOI).

Improvement with respect to the input:



(a) PESQ **(b)** SDR **(c)** STOI

Audio Examples: `https://team.inria.fr/perception/research/av-vae-se/`

Robust Audio-visual Speech Enhancement

Introduction

AV-VAE usually yields better results than A-VAE, especially at low SNRs, provided clean (frontal, non-occluded) visual data [Sadeghi et al., 2019].

Noisy visual data:

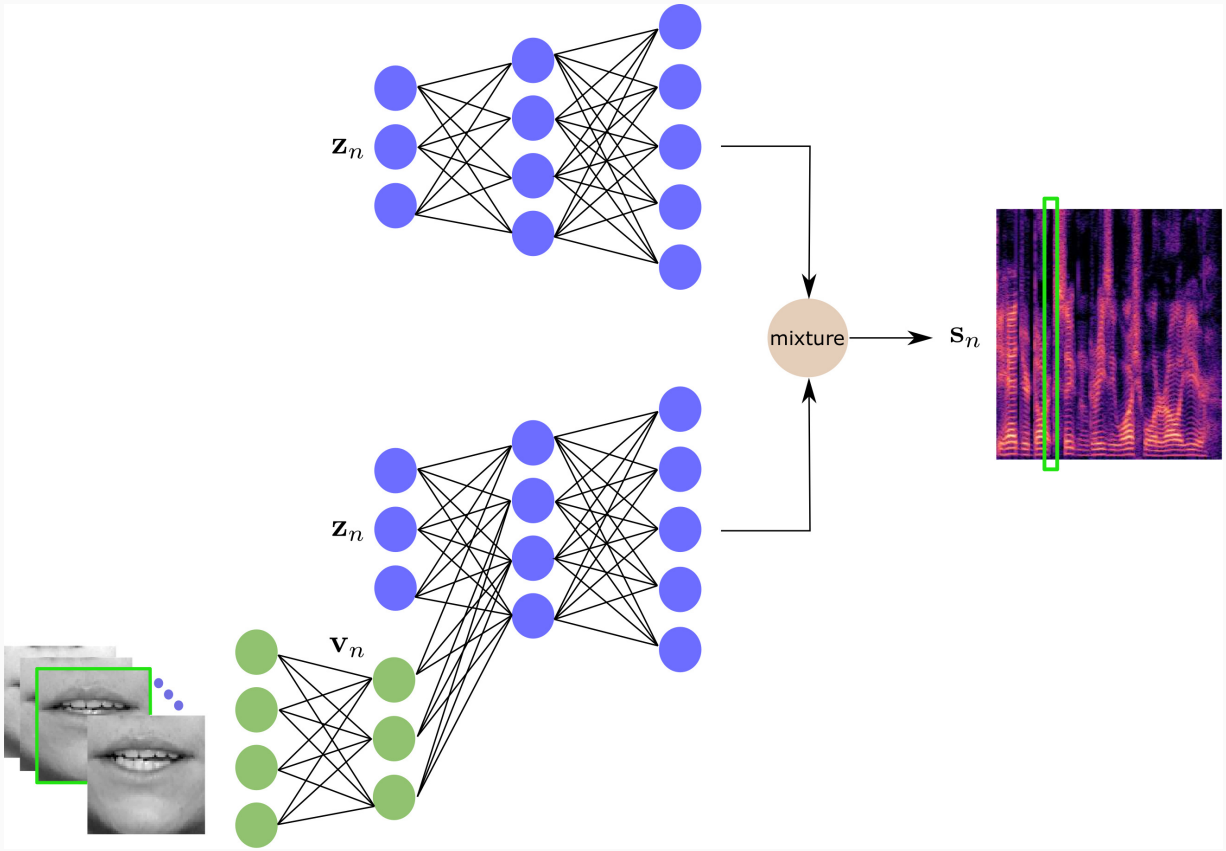
Some video frames might contain occluded and/or non-frontal lips region.

How to effectively benefit from AV-VAE for in-the-wild video recordings?

Our work: VAE mixture model (VAE-MM)

A mixture of A-VAE plus AV-VAE generative model:

- If the lips region is clean, use AV-VAE, otherwise use A-VAE.



The A-VAE and AV-VAE have been already trained on clean data.

Generative model

Mixture generative model: Combine A-VAE with AV-VAE

$$\begin{cases} p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n, \alpha_n) & = \Big[\mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_s^a(\mathbf{z}_n))\Big)\Big]^{\alpha_n} \times \Big[\mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_s^{av}(\mathbf{z}_n, \mathbf{v}_n))\Big)\Big]^{1-\alpha_n} \\ p(\mathbf{z}_n|\mathbf{v}_n, \alpha_n) & = \Big[\mathcal{N}(\mathbf{0}, \mathbf{I})\Big]^{\alpha_n} \times \Big[\mathcal{N}\Big(\boldsymbol{\mu}_z^v(\boldsymbol{v}_n), \mathrm{diag}(\boldsymbol{\sigma}_z^v(\mathbf{v}_n))\Big)\Big]^{1-\alpha_n}, \\ p(\alpha_n) & = \pi^{\alpha_n} \times (1-\pi)^{1-\alpha_n}. \end{cases}$$

$\alpha_n \in \{0, 1\}$ is a latent variable specifying the component of the mixture model that is used by the n-th frame.

Parameter Estimation (test time)

Noisy speech model: $\forall n: \quad \boldsymbol{x}_n = \boldsymbol{s}_n + \boldsymbol{b}_n$

Noise model: $\forall n: \quad \boldsymbol{b}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\mathbf{W}_b \mathbf{H}_b[:, n])\Big)$

Clean speech model: Mixture of A-VAE and AV-VAE generative networks

---

Inference:

- ▷ Observed variables: $\{\mathbf{x}_n, \mathbf{v}_n\}_{n=0}^{N-1}$
- ▷ Latent variables: $\{\mathbf{s}_n, \mathbf{z}_n, \alpha_n\}_{n=0}^{N-1}$
- ▷ Parameters to be estimated: $\boldsymbol{\theta}_u = \{\mathbf{W}_b, \mathbf{H}_b, \pi\}$

Parameter Estimation (test time)

Variational Expectation-maximization (VEM)

Variational E-Step:

The intractable posterior $p(\mathbf{s}_n, \mathbf{z}_n, \alpha_n | \mathbf{x}_n, \mathbf{v}_n; \boldsymbol{\theta}_u)$ is approximated by a variational distribution factorizing as follows:

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

The update formulas for the variational distributions [Bishop, 2006]:

VE $\mathbf{s}_n$-step: $$r(\mathbf{s}_n) \propto \exp\left(\mathbb{E}_{r(\mathbf{z}_n)\cdot r(\alpha_n)}\left[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \boldsymbol{\theta}_u)\right]\right)$$

VE $\mathbf{z}_n$-step: $$r(\mathbf{z}_n) \propto \exp\left(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\alpha_n)}\left[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \boldsymbol{\theta}_u)\right]\right)$$

VE α_n-step: $$r(\alpha_n) \propto \exp\left(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)}\left[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \boldsymbol{\theta}_u)\right]\right)$$

VE s_n-step

$$r(\mathbf{s}_n) = \mathcal{N}_c(\boldsymbol{m}_n, \mathrm{diag}(\boldsymbol{\nu}_n)), \quad \begin{cases} m_{fn} & = \frac{\gamma_{fn}}{\gamma_{fn} + (\mathbf{W}_b \mathbf{H}_b)_{fn}} \cdot x_{fn} \\ \nu_{fn} & = \frac{\gamma_{fn} \cdot (\mathbf{W}_b \mathbf{H}_b)_{fn}}{\gamma_{fn} + (\mathbf{W}_b \mathbf{H}_b)_{fn}} \end{cases}$$

which can be interpreted is an averaged Wiener filtering.

$\gamma_{fn}^{-1} = \sum_{\alpha_n \in \{0,1\}} r(\alpha_n) \cdot \eta_{fn}^{\alpha_n}$ (weighted precision over audio and audio-visual cases),

$\eta_{fn}^{\alpha_n} = \mathbb{E}_{r(\mathbf{z}_n)} \left[\frac{1}{\sigma_{s,f}^{\alpha_n}(\mathbf{z}_n, \mathbf{v}_n)} \right] \approx \frac{1}{D} \sum_{d=1}^{D} \frac{1}{\sigma_{s,f}^{\alpha_n}(\mathbf{z}_n^{(d)}, \mathbf{v}_n)}$ (average precision),

and $\{\mathbf{z}_n^{(d)}\}_{d=1}^{D}$ is a sequence sampled from $r(\mathbf{z}_n)$. Moreover:

$$\sigma_{s,f}^{\alpha_n}(\mathbf{z}_n, \mathbf{v}_n) = \begin{cases} \sigma_{s,f}^{a}(\mathbf{z}_n) & \alpha_n = 1 \\ \sigma_{s,f}^{av}(\mathbf{z}_n, \mathbf{v}_n) & \alpha_n = 0 \end{cases}.$$

VE z_n-step

For $r(\mathbf{z}_n)$ we obtain the following result:

$$r(\mathbf{z}_n) \propto \exp\Bigg(\sum_{\alpha_n \in \{0,1\}} r(\alpha_n) \cdot \Bigg[\log p(\mathbf{z}_n | \mathbf{v}_n, \alpha_n) + \sum_f -\log\left(\sigma_{s,f}^{\alpha_n}(\mathbf{z}_n, \mathbf{v}_n)\right) - \frac{|m_{fn}|^2 + \nu_{fn}}{\sigma_{s,f}^{\alpha_n}(\mathbf{z}_n, \mathbf{v}_n)} \Bigg] \Bigg).$$

The above distribution cannot be computed in closed-from. Nevertheless, we can draw samples from it using the Metropolis-Hastings (MH) algorithm (see our paper for more details).

VE α_n-step

To update the variational distribution of α_n, we can write:

$$r(\alpha_n) \propto \exp\left(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)}\left[\log p(\mathbf{s}_n|\mathbf{z}_n, \mathbf{v}_n, \alpha_n) + \log p(\mathbf{z}_n|\mathbf{v}_n, \alpha_n) + \log p(\alpha_n)\right]\right)$$

which is a Bernoulli distribution with

$$\pi_n = g\left(\mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)}\left[\log \frac{p(\mathbf{s}_n, \mathbf{z}_n|\mathbf{v}_n, \alpha_n = 1)}{p(\mathbf{s}_n, \mathbf{z}_n|\mathbf{v}_n, \alpha_n = 0)}\right] + \log \frac{\pi}{1-\pi}\right)$$

as the parameter, which is an averaged audio/audio-visual ratio. Here, $g(.)$ denotes the sigmoid function defined as $g(x) = 1/(1+\exp(-x))$.

Parameters Update and Speech Estimation

M-Step:

Update parameters by optimizing the complete data log-likelihood:

$$
\begin{aligned}
Q(\boldsymbol{\theta}_u, \boldsymbol{\theta}_u^{\text{old}}) &\stackrel{c}{=} \mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \boldsymbol{\theta}_u)\Big] \\
&\stackrel{c}{=} \sum_{f,n} -\log\left(\mathbf{W}_b\mathbf{H}_b\right)_{fn} - \left(\frac{|x_{fn} - m_{fn}|^2 + \nu_{fn}}{\left(\mathbf{W}_b\mathbf{H}_b\right)_{fn}}\right) \\
&+ \pi_n \log \pi + (1 - \pi_n)\log(1 - \pi).
\end{aligned}
$$

Speech Estimation:

After the convergence of the VEM, the speech STFT frames are estimated using an averaged Wiener filtering:

$$
\hat{s}_{fn} = \mathbb{E}_{r(s_{fn})}[s_{fn}] = \frac{\gamma_{fn}^*}{\gamma_{fn}^* + \left(\mathbf{W}_b^*\mathbf{H}_b^*\right)_{fn}} \cdot x_{fn} \quad \forall (f, n)
$$

Experiments

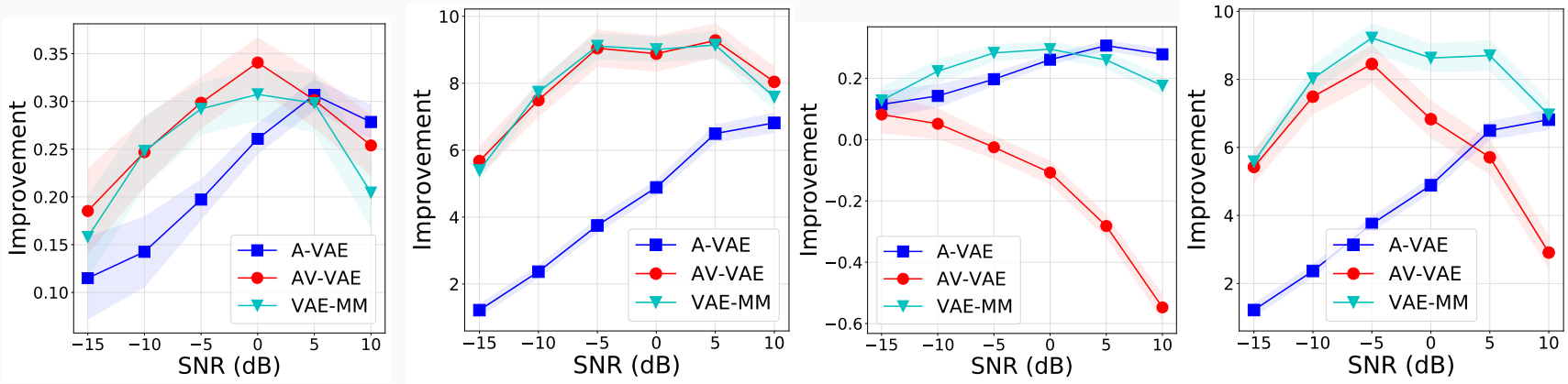
- **Noisy+clean speech**: NTCD-TIMIT database [Abdelaziz, 2017]
- **VAE models**: Pre-trained A-VAE and AV-VAE [Sadeghi et al., 2019]
- **Setup**:
  - Testing set of NTCD-TIMIT database;
  - $\sim$ 1 hours of speech;
  - 9 speakers;
  - Noise types: *LR*, *White*, *Cafe*, *Car*, *Babble*, and *Street*;
  - Noise levels: $-15, -10, -5, 0, 5, 10$ dB;
  - 270 noisy mixtures per noise level;
  - Different speakers and sentences than in the training set;
  - Clean lips region as well as noisy versions ($\sim$ one-third of total video frames per sample)

Results

Objective measures (the higher, the better)

- Signal-to-distortion ratio (SDR).
- Perceptual evaluation of speech quality (PESQ) measure.

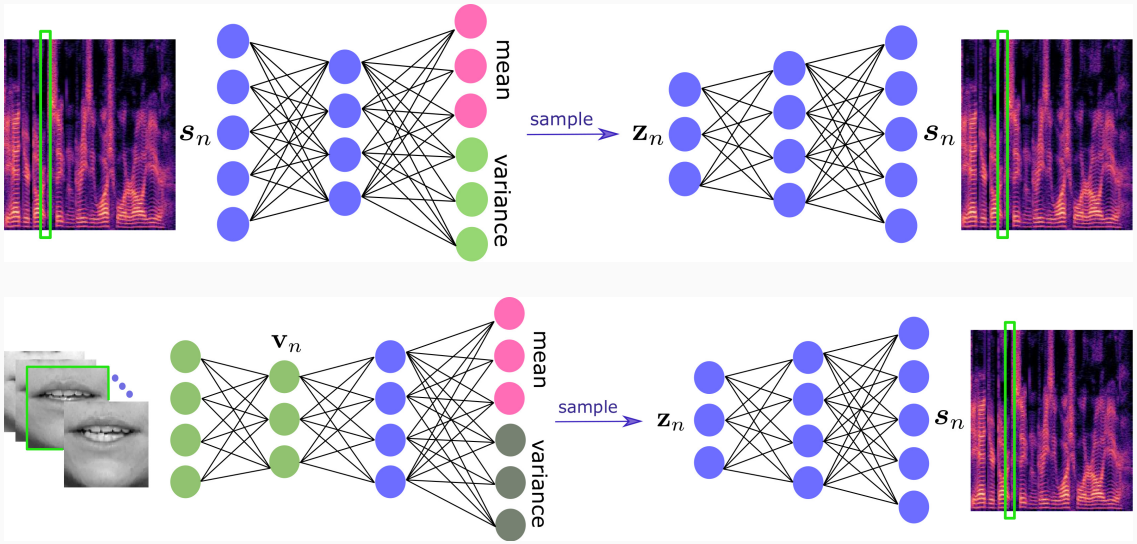
Improvement with respect to the input:



(a) PESQ(clean) **(b)** SDR(clean) **(c)** PESQ(noisy) **(d)** SDR(noisy)

Mixture of Audio-visual Inference Networks

Introduction



▷ Initialization of latent codes at test phase: Posterior mean of the encoder.

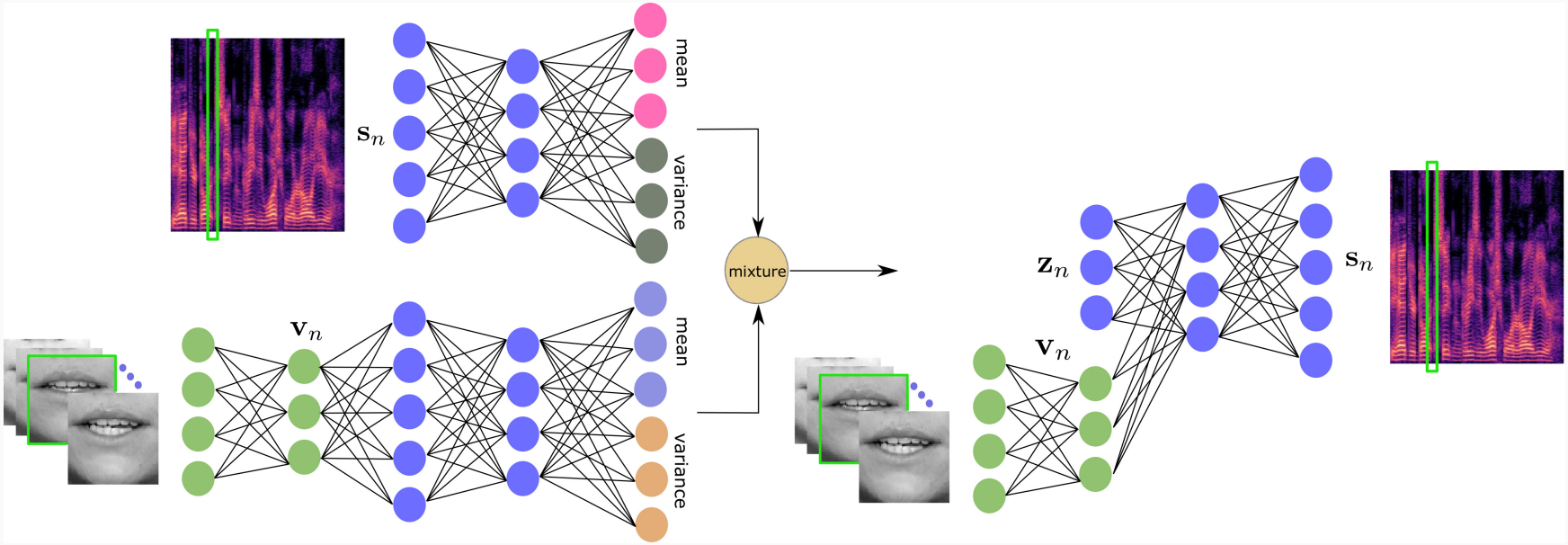
A-VAE: inputs noisy speech, while V-VAE inputs clean visual data.

▷ Better audio-visual fusion: Concatenating audio and visual data or tight fusion (AV-VAE) is not good when one modality is very weak.

How to take advantage of both A-VAE and V-VAE and provide a better audio-visual fusion?

Mixture of Inference Networks VAE (MIN-VAE)

Train a mixture of audio and visual inference networks:[2]



- ☞ The shared generative model (decoder) is trained using both audio and visual latent codes
- ☞ Once trained, the latent codes at test phase can be initialized using the visual encoder

[2] M. Sadeghi and X. Alameda-Pineda, "Mixture of Inference Networks for VAE-based Audio-visual Speech Enhancement," [Available Online] `https://arxiv.org/abs/1912.10647`, May 2020.

Mixture of Inference Networks: Generative Model

The generative model for each $\mathbf{s}_n$:

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

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

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

- Each latent code, $\mathbf{z}_n$, is generated either from an audio or from a video prior,
- The audio and video priors are parametrized by $(\boldsymbol{\mu}_a, \sigma_a)$ and $(\boldsymbol{\mu}_v, \sigma_v)$,
- A mixing variable $\alpha_n \in \{0, 1\}$ describes whether $\mathbf{z}_n$ corresponds to the audio or to the video prior.

Training MIN-VAE

Posterior distribution of the latent variables:

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

The two posteriors in the RHS are intractable. We assume some variational approximation:

$$q(\mathbf{z}_n | \mathbf{s}_n, \mathbf{v}_n, \alpha_n; \boldsymbol{\phi}) = \begin{cases} q(\mathbf{z}_n | \mathbf{s}_n; \boldsymbol{\phi}_a) & \alpha_n = 1 \quad \text{(audio encoder)} \\ q(\mathbf{z}_n | \mathbf{v}_n; \boldsymbol{\phi}_v) & \alpha_n = 0 \quad \text{(video encoder)} \end{cases}$$

A variational distribution, denoted $r(\alpha_n)$, is considered for $p(\alpha_n | \mathbf{s}_n, \mathbf{v}_n)$.

Final approximation

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

Training MIN-VAE

Inference:

- ▷ Observed variables: $\mathbf{s} = \{\mathbf{s}_n\}_{n=0}^{N_{tr}-1}$ and $\mathbf{v} = \{\mathbf{v}_n\}_{n=0}^{N_{tr}-1}$
- ▷ Latent variables: $\mathbf{z} = \{\mathbf{z}_n\}_{n=0}^{N_{tr}-1}$ and $\boldsymbol{\alpha} = \{\alpha_n\}_{n=0}^{N_{tr}-1}$
- ▷ Parameters to be estimated: $\boldsymbol{\Theta} = \{\boldsymbol{\psi}, \boldsymbol{\phi}, \pi, \boldsymbol{\mu}_a, \boldsymbol{\mu}_v, \sigma_a, \sigma_v\}$ and r

---

We target a lower bound on the data log-likelihood $\log p(\mathbf{s}, \mathbf{v}; \boldsymbol{\theta})$:

$$\max_{\boldsymbol{\Theta}, r} \ \mathcal{L}(\boldsymbol{\Theta}, r) =$$

$$\max_{\boldsymbol{\Theta}, r} \ \int_{\mathbb{Z},\mathbb{A}} q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\psi}) r(\boldsymbol{\alpha}) \log \frac{p(\mathbf{s}|\mathbf{z}; \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})}{q(\mathbf{z}|\mathbf{s}, \mathbf{v}, \boldsymbol{\alpha}; \boldsymbol{\psi}) r(\boldsymbol{\alpha})} \mathrm{d}\mathbf{z}\mathrm{d}\boldsymbol{\alpha}$$

Training MIN-VAE

- $r(\alpha_n) = \pi_n^{\alpha_n} \cdot (1-\pi_n)^{1-\alpha_n}$

$$\begin{cases} \pi_n = g\Big(J_n(\alpha_n=0) - J_n(\alpha_n=1) + \log\frac{\pi}{1-\pi}\Big), \\ J_n(\alpha_n) = \mathcal{D}_{KL}\Big(q(\mathbf{z}_n|\mathbf{s}_n,\mathbf{v}_n,\alpha_n;\boldsymbol{\phi})\Big\|p(\mathbf{z}_n|\alpha_n)\Big) - \log p(\mathbf{s}_n|\mathbf{z}_n^{\alpha_n},\mathbf{v}_n;\boldsymbol{\theta}) \\ \mathbf{z}_n^{\alpha_n} \sim q(\mathbf{z}_n|\mathbf{s}_n,\mathbf{v}_n,\alpha_n;\boldsymbol{\phi}) \end{cases}$$

- $\boldsymbol{\Theta}$ is updated by:

$$\min_{\boldsymbol{\Theta}} \sum_{n=0}^{N_{tr}} \mathbb{E}_{r(\alpha_n)}\Big[J_n(\alpha_n)\Big] + \mathcal{D}_{KL}\Big(r\left(\alpha_n\right) \parallel p(\alpha_n)\Big)$$

  - Stochastic gradient descent for $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\mu}_a, \boldsymbol{\mu}_v, \sigma_a, \sigma_v$
  - $\pi = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \pi_n$

Parameter Estimation (test time)

Noisy speech model: $\forall n: \quad \boldsymbol{x}_n = \boldsymbol{s}_n + \boldsymbol{b}_n$

Noise model: $\forall n: \quad \boldsymbol{b}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \mathsf{diag}(\mathbf{W}_b \mathbf{H}_b[:, n])\Big)$

Clean speech model: Trained MIN-VAE

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

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

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

Inference:

- ▷ Observed variables: $\{\mathbf{x}_n, \mathbf{v}_n\}_{n=0}^{N-1}$
- ▷ Latent variables: $\{\mathbf{s}_n, \mathbf{z}_n, \alpha_n\}_{n=0}^{N-1}$
- ▷ Parameters to be estimated: $\boldsymbol{\theta}_u = \{\mathbf{W}_b, \mathbf{H}_b, \pi\}$

Parameter Estimation (test time)

Intractable posterior$\rightarrow$ variational approximation:

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

VE $\mathbf{s}_n$-step:
$$r(\mathbf{s}_n) = \mathcal{N}_c(m_{fn}, \nu_{fn}), \begin{cases} m_{fn} & = \frac{\gamma_{fn}}{\gamma_{fn}+(\mathbf{WH})_{fn}} \cdot x_{fn} \\ \nu_{fn} & = \frac{\gamma_{fn}\cdot(\mathbf{WH})_{fn}}{\gamma_{fn}+(\mathbf{WH})_{fn}} \\ \gamma_{fn}^{-1} & = \frac{1}{D}\sum_{d=1}^{D} \frac{1}{\sigma_{s,f}(\mathbf{z}_n^{(d)}, \mathbf{v}_n)} \\ \mathbf{z}_n^{(d)} & \sim r(\mathbf{z}_n), \quad d = 1, \ldots, D \end{cases}$$

VE $\mathbf{z}_n$-step:
$$r(\mathbf{z}_n) \propto \exp\Big(\sum_f -\log\Big(\sigma_{s,f}(\mathbf{z}_n, \mathbf{v}_n)\Big) - \frac{|m_{fn}|^2+\nu_{fn}}{\sigma_{s,f}(\mathbf{z}_n, \mathbf{v}_n)} + \sum_{\alpha_n \in \{0,1\}} r(\alpha_n) \cdot \Big[\log p(\mathbf{z}_n|\alpha_n)\Big]\Big)$$

VE α_n-step:
$$r(\alpha_n) = \pi_n^{\alpha_n} \cdot (1-\pi_n)^{1-\alpha_n},$$

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

Parameters Update and Speech Estimation

M-Step:

Update parameters by optimizing the complete data log-likelihood:

$$
\begin{aligned}
Q(\boldsymbol{\theta}_u, \boldsymbol{\theta}_u^{\text{old}}) &\overset{c}{=} \mathbb{E}_{r(\mathbf{s}_n)\cdot r(\mathbf{z}_n)\cdot r(\alpha_n)}\Big[\log p(\mathbf{x}_n, \mathbf{s}_n, \mathbf{z}_n, \alpha_n, \mathbf{v}_n; \boldsymbol{\theta}_u)\Big] \\
&\overset{c}{=} \mathbb{E}_{r(\mathbf{s}_n)}\Big[\log p(\mathbf{x}_n|\mathbf{s}_n; \boldsymbol{\theta}_u)\Big] + \mathbb{E}_{r(\alpha_n)}\Big[\log p(\alpha_n)\Big] \\
&\overset{c}{=} \sum_{f,n} -\log\left(\mathbf{W}_b\mathbf{H}_b\right)_{fn} - \left(\frac{|x_{fn} - m_{fn}|^2 + \nu_{fn}}{\left(\mathbf{W}_b\mathbf{H}_b\right)_{fn}}\right) \\
&+ \pi_n \log \pi + (1 - \pi_n)\log(1 - \pi).
\end{aligned}
$$

Speech Estimation:

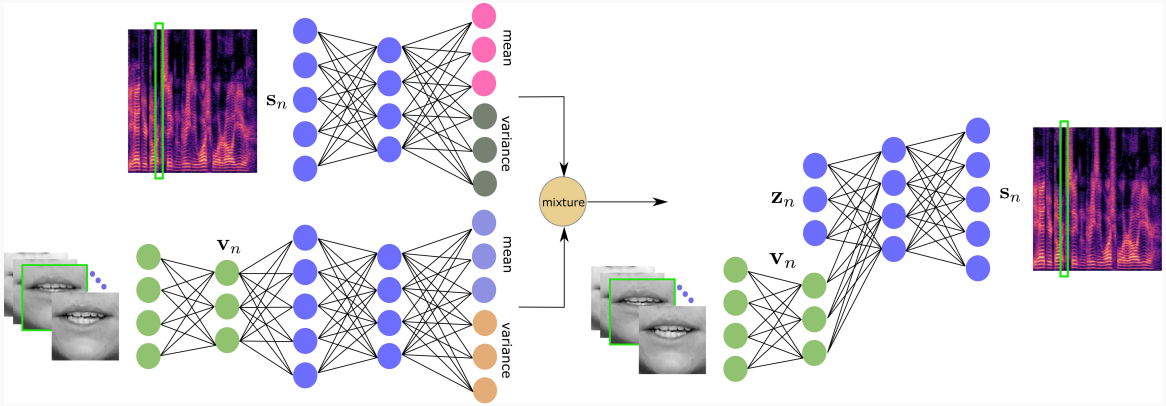
After the convergence of the VEM, the speech STFT frames are estimated using an averaged Wiener filtering:

$$
\hat{s}_{fn} = \mathbb{E}_{r(s_{fn})}[s_{fn}] = \frac{\gamma_{fn}^*}{\gamma_{fn}^* + \left(\mathbf{W}_b^*\mathbf{H}_b^*\right)_{fn}} \cdot x_{fn} \quad \forall (f, n)
$$

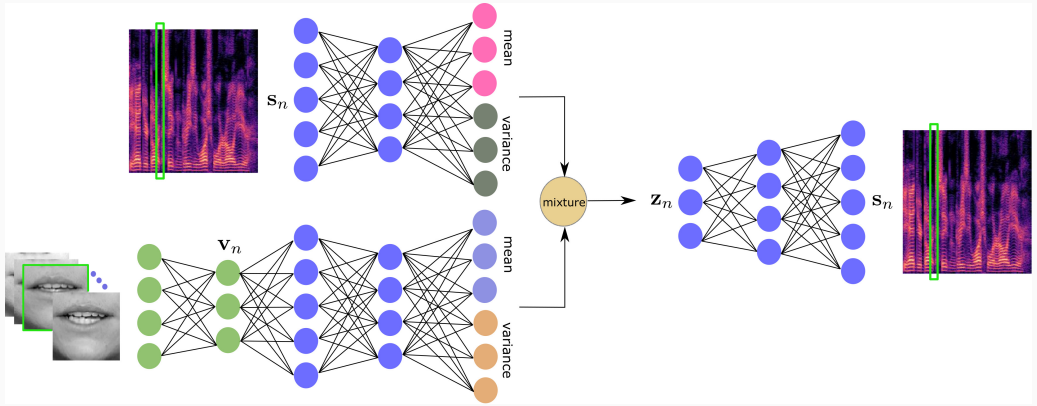
Experiments

Settings similar as before with the NTCD-TIMIT dataset

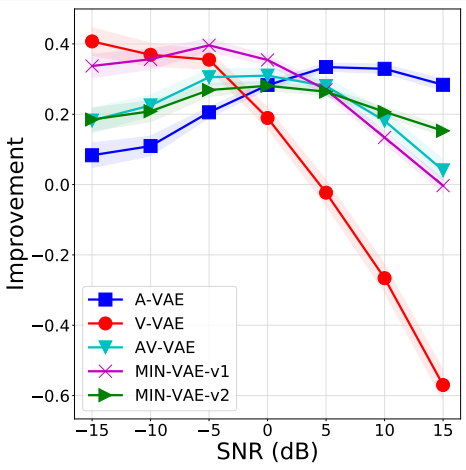
MIN-VAE-v1:



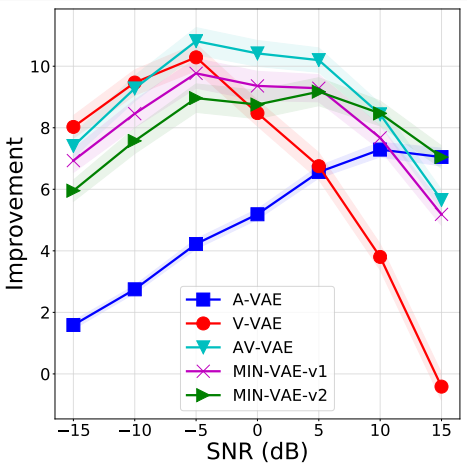
MIN-VAE-v2:



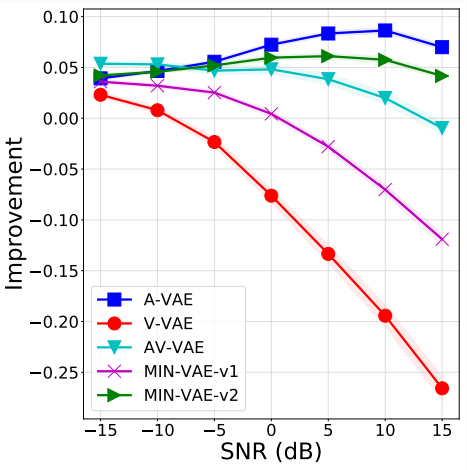
Results



(a) PESQ

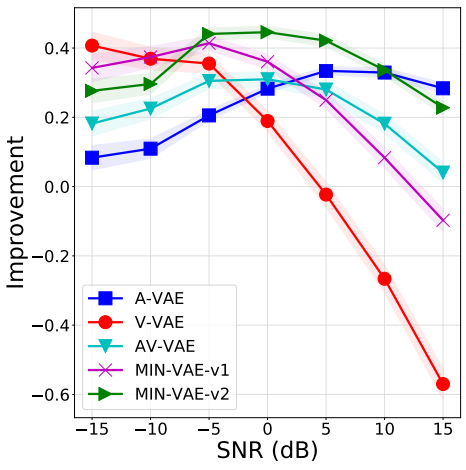


(b) SDR

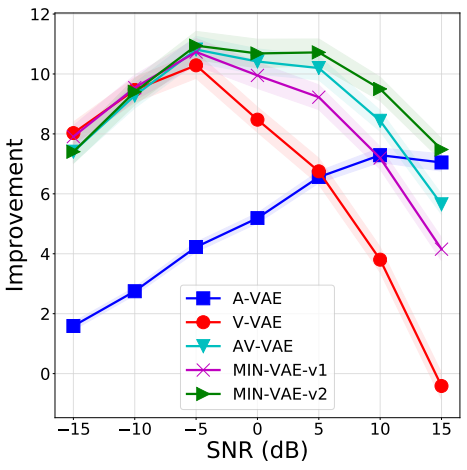


(c) STOI

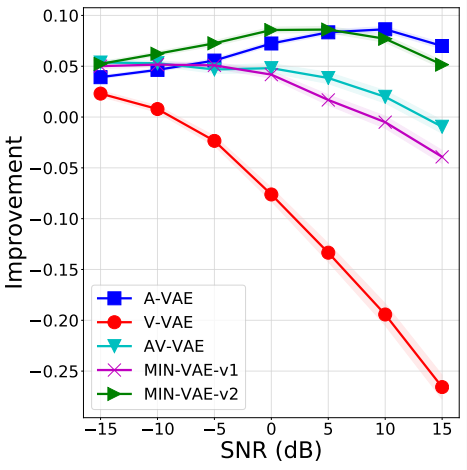
Adding noise to one-third of spectrograms input to audio-encoder:



(d) PESQ



(e) SDR



(f) STOI

Conclusion and Future Work

Variational autoencoders provide efficient ways to fuse audio and visual modalities for clean speech modeling and speech enhancement.

- The VEM framework is slow. Trying to re-use the trained encoders at inference time can reduce the complexity.
- Temporal modeling of the latent variables to benefit from time dependency between audio as well as visual frames.
- Extending the robust VAE to more than two models: A-VAE, V-VAE, and AV-VAE.

Thank you for your attention

References

1. D. P. Kingma and M. Welling, "*Auto-encoding variational Bayes*," ICLR, 2014.
2. Y. Bando et al., "*Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization*," in Proc. ICASSP, 2018, pp. 716–720
3. S. Leglaive et al., "*A variance modeling framework based on variational autoencoders for speech enhancement*," in Proc. MLSP, 2018.
4. M. Sadeghi et al., "*Audio-visual Speech Enhancement Using Conditional Variational Auto-Encoder*," `https://arxiv.org/abs/1908.02590`, August 2019.
5. C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag Berlin, Heidelberg, 2006.
6. A. H. Abdelaziz, "*NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition*," in Proc. INTERSPEECH, 2017.