Unsupervised Probabilistic Learning with Latent Variables

Lecture 2: The variational EM algoritm, mixtures of VAEs, application to AV speech enhancement.

MLSS Africa 2023, Cape Town

Slides: https://xavirema.eu/MLSS2023/

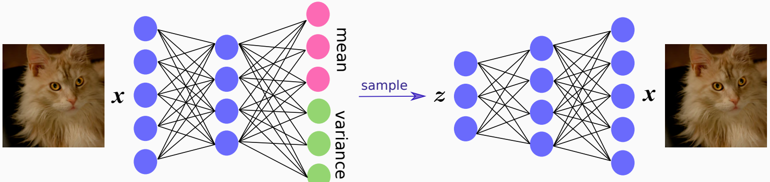
Xavier Alameda-Pineda

RobotLearn team, Inria at University Grenoble-Alpes,
Jean-Kuntzman CNRS Laboratory, Multidisciplinary Institute of Artificial Intelligence

VAE: Summary

Generative model. Prior: $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ and decoder: $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}\Big(\boldsymbol{x}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{z})\Big)$.

Inference model (encoder): $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \approx q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\Big(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})\Big)$



Training criterion (maximise the evidence lower bound):

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \underbrace{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\hat{\boldsymbol{z}})}_{\text{Reconstruction}} - \underbrace{D_{\text{KL}}\Big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \Big\| p(\boldsymbol{z})\Big)}_{\text{Regularisation}}$$

where $\hat{\boldsymbol{z}} \sim q_{\boldsymbol{\phi}}$ is sampled using the reparametrisation trick.

Learning - ELBO

If we recall the formulation for the EM:

$$\log p(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{\mathsf{KL}}\Big(q(\boldsymbol{z}|\boldsymbol{x})\Big\|p(\boldsymbol{z}|\boldsymbol{x})\Big)$$

Learning - ELBO

If we recall the formulation for the EM:

$$\log p(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{\mathsf{KL}}\Big(q(\boldsymbol{z}|\boldsymbol{x})\Big\|p(\boldsymbol{z}|\boldsymbol{x})\Big)$$

Problem: the second term cannot be computed! But it's positive:

$$\log p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\phi}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right]$$

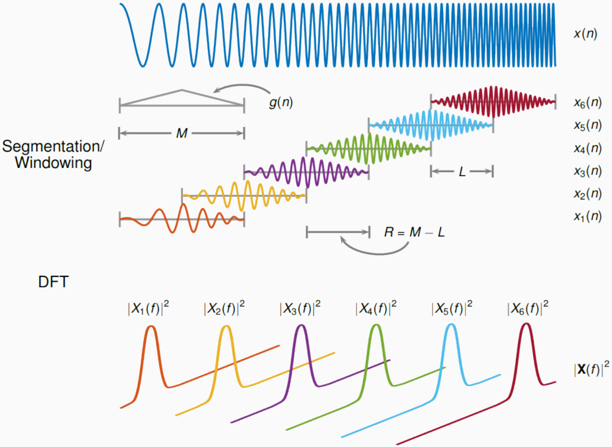
$$\log p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\phi}) \geq \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\Big]}_{\text{Reconstruction}} - \underbrace{D_{\mathsf{KL}}\Big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\Big\|p(\boldsymbol{z})\Big)}_{\text{Regularisation}}$$

This is known as **Evidence Lower-BOund or ELBO**: $\mathcal{L}_{\mathsf{ELBO}}(\boldsymbol{\theta},\boldsymbol{\phi})$.

Be VERY careful with these expressions: They look alike, but they are NOT the same.

The short-time Fourier transform (STFT)

STFT: segment the input signal, and apply DFT to each segment.



[Image from Mathworks]

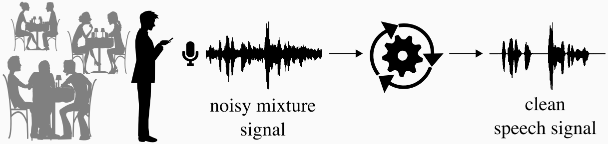
Lecture's Outline

1. Unsupervised Audio-visual Speech Enhancement
2. Conditional Audio-Visual VAE
3. Mixtures of VAEs
4. Mixture of Inference Networks VAE

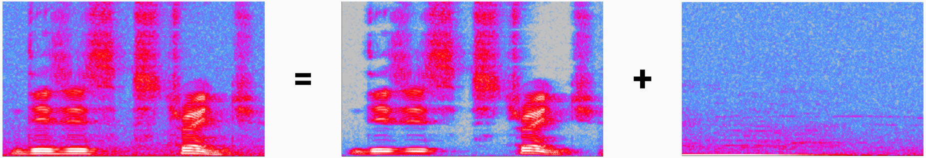
Unsupervised Audio-visual Speech Enhancement (AV-SE)

What is Speech Enhancement

Speech Enhancement: Remove the background noise from the observed mixture speech.



Short-time Fourier transform (STFT) is a time-frequency (matrix) representation.



$$\underbrace{\mathbf{x}}_{\text{Noisy mixture}} = \underbrace{\mathbf{s}}_{\text{Clean speech}} + \underbrace{\mathbf{b}}_{\text{Noise signal}}$$

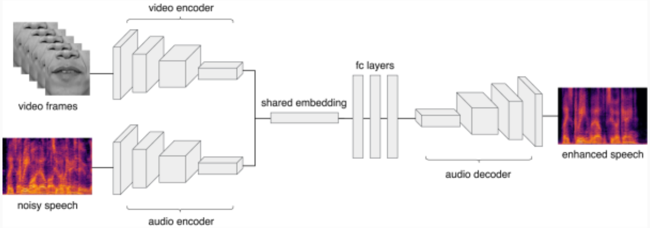
Audio-visual Speech Enhancement (AV-SE)

- 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 investigate the use of VAEs to efficiently fuse audio and visual modalities for SE.

Supervised vs. Unsupervised AVSE

Supervised: Learn a mapping to denoise audio data [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

Unsupervised SE is **more flexible** since it adapts to various noises.

Unsupervised speech enhancement: overview

Train a generative speech model with clean data: $\{\mathbf{s}_i\}_{i=1}^{N}$.

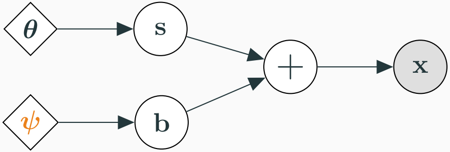
$$\boldsymbol{\theta} \longrightarrow \mathbf{s} \qquad\qquad p_{\boldsymbol{\theta}}(\mathbf{s}) \approx p_{\mathsf{data}}(\mathbf{s})$$

Unsupervised speech enhancement: overview

Train a generative speech model with clean data: $\{\mathbf{s}_i\}_{i=1}^{N}$.

$$p_{\boldsymbol{\theta}}(\mathbf{s}) \approx p_{\mathsf{data}}(\mathbf{s})$$

Test: learn the noise parameters of $\mathbf{x}$:

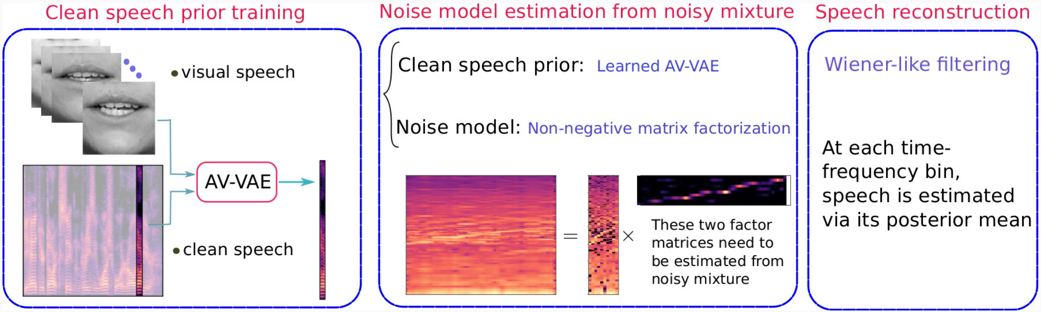


Note: at test time, $\boldsymbol{\theta}$ is frozen, and $\mathbf{s}$ becomes a latent variable.

Pre-training, noise estimation & enhancement

Generally speaking our pipeline has three phases:

1. Learn a deep generative model of the clean data.
2. Estimate the noise parameters from a mixture.
3. Enhance the speech signal (Wiener-like filtering).



Conditional Audio-Visual VAE

Conditional Audio-Visual VAE

1. Learn a deep generative model

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

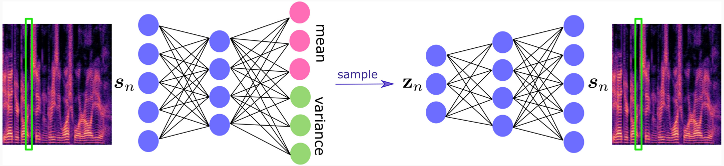
Generative model (Decoder):

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

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

Inference network (Encoder):

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



Video-only VAE [Sadeghi et al., 2020]

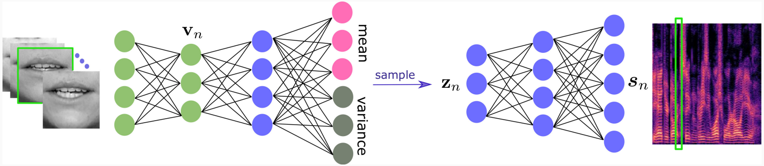
Generative model (Decoder):

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

Inference network (Encoder):

Infer the posterior using visual data only:

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



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

Audio-visual VAE [Sadeghi et al., 2020a]

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

Generative network (Decoder):

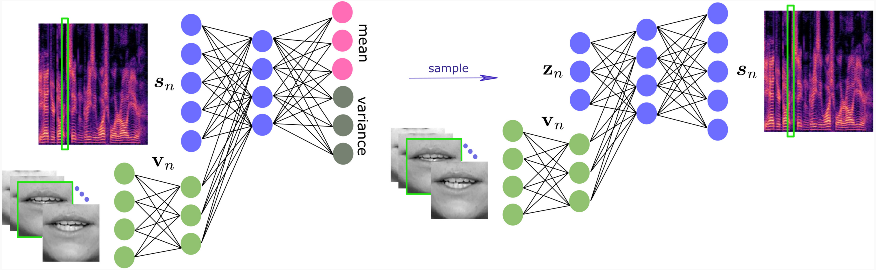
Each clean spectrogram time frame $\boldsymbol{s}_n$ is assumed to be generated as:

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

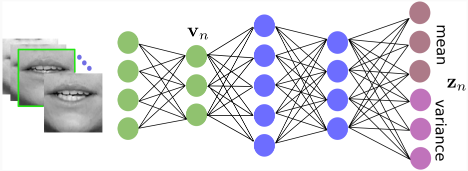
Inference network (Encoder):

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

Encoder and decoder networks:



Prior distribution for latent variables:



Training AV-VAE

Training samples:

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

Optimize the ELBO:

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

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

Conditional Audio-Visual VAE

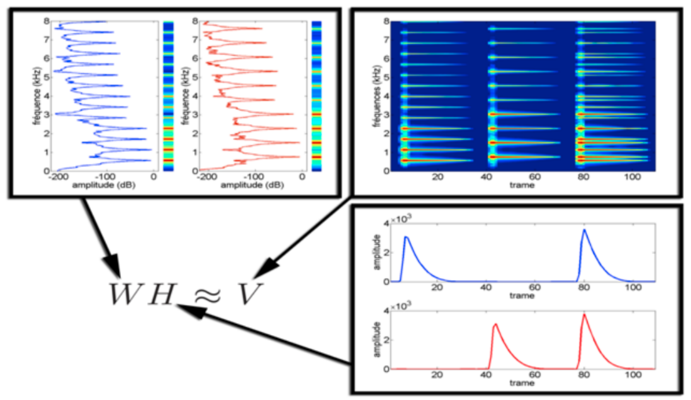
2. Estimate the noise parameters from x

Noise estimation: the model

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

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

Model based on non-negative matrix factorisation:



[Image from `https://perso.telecom-paristech.fr/essid/`]

Noise estimation: the problem formulation

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

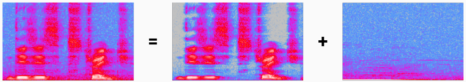
Noise model: $\forall n: \quad \boldsymbol{b}_n \sim \mathcal{N}_c\left(\mathbf{0}, \mathrm{diag}(\mathbf{W}_b \mathbf{H}_b[:, n])\right)$

Clean speech model: Pre-trained generative network:

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

Inference:

- ▷ Parameters to be estimated: $\psi = \{\mathbf{W}_b, \mathbf{H}_b\}$
- ▷ Observed variables: $\mathbf{x}_n$'s and $\boldsymbol{v}_n$'s.
- ▷ Latent variables: $\boldsymbol{z}_n$'s and $\boldsymbol{s}_n$'s.



$$\underbrace{\mathbf{x}}_{\text{Noisy mixture}} = \underbrace{\mathbf{s}}_{\text{Clean speech}} + \underbrace{\mathbf{b}}_{\text{Noise signal}}$$

Noise estimation: the optimisation criterion

We opt for an EM-like solution. Given an initial value of the parameters $\boldsymbol{\psi}^0$:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_n \mathbb{E}_{p(\boldsymbol{s}_n, \boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)} [\log p(\boldsymbol{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi})].$$

Noise estimation: the optimisation criterion

We opt for an EM-like solution. Given an initial value of the parameters $\boldsymbol{\psi}^0$:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_n \mathbb{E}_{p(\boldsymbol{s}_n, \boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)}[\log p(\boldsymbol{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi})].$$

Stop! There is an *easier* way. We can marginalise w.r.t. the $\boldsymbol{s}_n$'s:
$\big(p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi}) = \int_{\boldsymbol{s}_n} p(\boldsymbol{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi}) \mathrm{d}\boldsymbol{s}_n\big)$

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

Thus the expected complete-data log-likelihood can be rewritten as:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_n \mathbb{E}_{p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi})].$$

Noise estimation: the algorithm

One issue (of both $\mathcal{Q}$) is that the posterior is intractable:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_n \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{v}_n; \boldsymbol{\psi})].$$

Noise estimation: the algorithm

One issue (of both $\mathcal{Q}$) is that the posterior is intractable:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_n \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{v}_n; \boldsymbol{\psi})].$$

We propose to approximate the $\mathcal{Q}$ function with Monte-Carlo sampling, leading to a MCEM:

- **E-Step**: $Q(\boldsymbol{\psi}; \boldsymbol{\psi}^0) \approx \sum_{n=1}^{N} \frac{1}{R} \sum_{r=1}^{R} \ln p(\mathbf{x}_n, \boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n; \boldsymbol{\psi})$
  The samples $\left\{\boldsymbol{z}_n^{(r)}\right\}_{r=1,\ldots,R}$ are i.i.d. and drawn from $p(\boldsymbol{z}_n | \mathbf{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)$.
- **M-Step**: $\boldsymbol{\psi}^1 \leftarrow \operatorname{argmax}_{\boldsymbol{\psi}} Q(\boldsymbol{\psi}; \boldsymbol{\psi}^0) \Rightarrow$ multiplicative update rules.
  Standard formulae for NMF, not detailed here.

Conditional Audio-Visual VAE

3. Speech Enhancement

Posterior Speech Distribution

Quick reminder: pre-trained decoder $p(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{v}_n)$ and estimated noise parameters $\boldsymbol{\psi}^* = \{\boldsymbol{W}_b^*, \boldsymbol{H}_b^*\}$. The task is to find the expected value of $\boldsymbol{s}_n$ given $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$:

$$\hat{\boldsymbol{s}}_n = \mathbb{E}_{p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)}[\boldsymbol{s}_n] = \int_{\boldsymbol{s}_n} p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)\boldsymbol{s}_n \mathrm{d}\boldsymbol{s}_n.$$

Posterior Speech Distribution

Quick reminder: pre-trained decoder $p(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{v}_n)$ and estimated noise parameters $\boldsymbol{\psi}^* = \{\boldsymbol{W}_b^*, \boldsymbol{H}_b^*\}$. The task is to find the expected value of $\boldsymbol{s}_n$ given $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$:

$$\hat{\boldsymbol{s}}_n = \mathbb{E}_{p(\boldsymbol{s}_n|\boldsymbol{x}_n,\boldsymbol{v}_n)}[\boldsymbol{s}_n] = \int_{\boldsymbol{s}_n} p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)\boldsymbol{s}_n \mathrm{d}\boldsymbol{s}_n.$$

Problem: there is not close-form for $p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)$ because of $\boldsymbol{z}_n$, thus:

$$\hat{\boldsymbol{s}}_n = \int_{\boldsymbol{z}_n} \left(\int_{\boldsymbol{s}_n} p(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{v}_n)\boldsymbol{s}_n \mathrm{d}\boldsymbol{s}_n \right) p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)\mathrm{d}\boldsymbol{z}_n.$$

Inner integral: can be computed in closed-form.

Outer integral: approximated with the samples of the MCEM ($\mathbf{z}_n^{(r)}$).

Posterior Speech Distribution

Quick reminder: pre-trained decoder $p(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{v}_n)$ and estimated noise parameters $\boldsymbol{\psi}^* = \{\boldsymbol{W}_b^*, \boldsymbol{H}_b^*\}$. The task is to find the expected value of $\boldsymbol{s}_n$ given $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$:

$$\hat{\boldsymbol{s}}_n = \mathbb{E}_{p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)}[\boldsymbol{s}_n] = \int_{\boldsymbol{s}_n} p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)\boldsymbol{s}_n \mathrm{d}\boldsymbol{s}_n.$$

Problem: there is not close-form for $p(\boldsymbol{s}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)$ because of $\boldsymbol{z}_n$, thus:

$$\hat{\boldsymbol{s}}_n = \int_{\boldsymbol{z}_n} \left(\int_{\boldsymbol{s}_n} p(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{v}_n)\boldsymbol{s}_n \mathrm{d}\boldsymbol{s}_n \right) p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{v}_n)\mathrm{d}\boldsymbol{z}_n.$$

Inner integral: can be computed in closed-form.

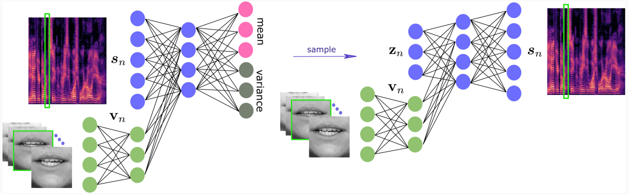
Outer integral: approximated with the samples of the MCEM ($\mathbf{z}_n^{(r)}$).

Clean speech estimated via a Wiener filter-like estimator (entry-wise vector operations):

$$\hat{\boldsymbol{s}}_n = \frac{1}{R} \sum_{r=1}^{R} \frac{\boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n)}{\boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n) + (\mathbf{W}_b^* \mathbf{H}_b^*)[;, n]} \boldsymbol{x}_n.$$

Summary

1. Learn a deep generative model of the clean data.



2. Estimate the noise parameters from a noisy mixture $\boldsymbol{x}$ using MCEM:

$$Q(\boldsymbol{\psi}; \boldsymbol{\psi}^0) \approx \sum_{n=1}^{N} \frac{1}{R} \sum_{r=1}^{R} \ln p(\mathbf{x}_n, \boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n; \boldsymbol{\psi})$$

3. Clean speech estimated via a Wiener filter-like estimator:

$$\hat{\boldsymbol{s}}_n = \frac{1}{R} \sum_{r=1}^{R} \frac{\boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n)}{\boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n) + (\mathbf{W}_b^* \mathbf{H}_b^*)[; , n]} \boldsymbol{x}_n.$$

Conditional Audio-Visual VAE

Experiments

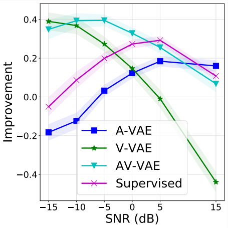
Settings

- ▷ 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*

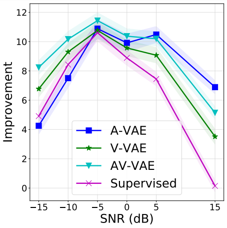
Results

Performance measures (the higher, the better) – improvement w.r.t. the noisy mixture:

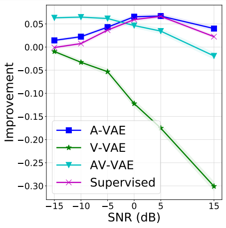
- Perceptual evaluation of speech quality (PESQ).
- Signal-to-distortion ratio (SDR).
- Short-time objective intelligibility (STOI).



(a) PESQ



(b) SDR



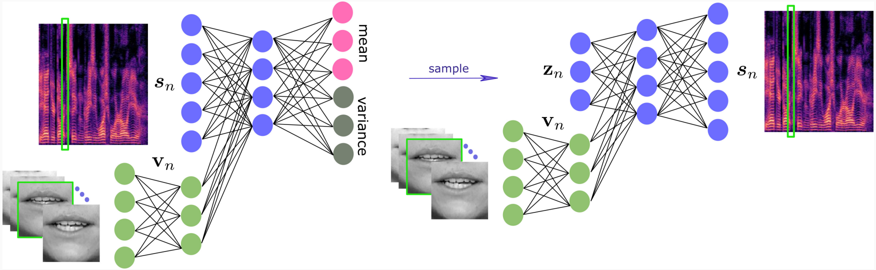
(c) STOI

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

Systematic AV fusion

The conditional AV-VAE uses ALWAYS visual information.

This can be a problem when dealing with noisy visual data.



Mixtures of VAEs

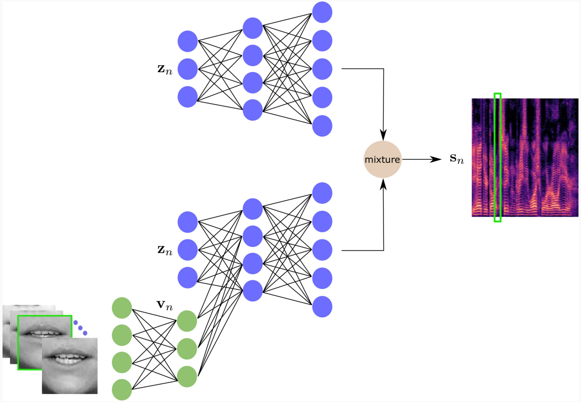
Motivation

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

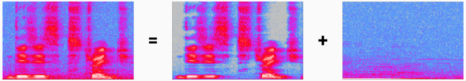
How to effectively benefit from AV-VAE for in-the-wild video recordings?

VAE mixture model (VAE-MM) [Sadeghi et al., 2020b]

A mixture of A-VAE plus AV-VAE generative model: use visual information only if clean.



After trained, used for unsupervised AV-SE.



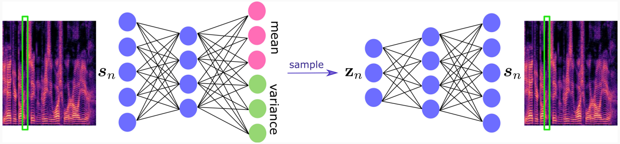
$$\underbrace{\mathbf{x}}_{\text{Noisy mixture}} = \underbrace{\mathbf{s}}_{\text{Clean speech}} + \underbrace{\mathbf{b}}_{\text{Noise signal}}$$

Mixtures of VAEs

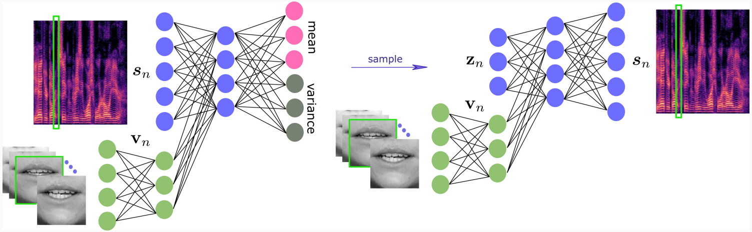
1. Learn a deep generative model

Pre-training as in the previous model

Audio-only VAE: $\boldsymbol{s}_n | \boldsymbol{z}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_s(\boldsymbol{z}_n))\Big)$ and $\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$



Conditional AV VAE: $\boldsymbol{s}_n | \boldsymbol{z}_n, \boldsymbol{v}_n \sim \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_s(\boldsymbol{z}_n, \boldsymbol{v}_n))\Big)$ and $\boldsymbol{z}_n | \boldsymbol{v}_n \sim \mathcal{N}\Big(\boldsymbol{\mu}_z(\boldsymbol{v}_n), \mathrm{diag}(\boldsymbol{\sigma}_z(\boldsymbol{v}_n))\Big)$



Formalising VAE-MM

VAE-MM clean speech model: Combine A-VAE with AV-VAE:

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

$\alpha_n \in \{0, 1\}$ is a latent variable selecting the VAE used in the n-th frame.

Mixtures of VAEs

2. & 3. Estimate the noise parameters from x

Noise model and optimisation problem

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

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

Clean speech model: Mixture of A-VAE and AV-VAE.

Inference:

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

Noise model and optimisation problem

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

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

Clean speech model: Mixture of A-VAE and AV-VAE.

Inference:

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

Ideally:

$$Q(\boldsymbol{\psi}, \boldsymbol{\psi}^0) = \sum_{n=1}^N \mathbb{E}_{p(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \mathbf{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0)}[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi})]$$

Approximating the posterior distribution

Challenge: The posterior is intractable, we propose a variational approximation:

(in this case, marginalisation w.r.t. $\boldsymbol{s}_n$ does not simplify things)

$$p(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \mathbf{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0) \approx q(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n) = q_{\boldsymbol{s}}(\boldsymbol{s}_n)\, q_{\boldsymbol{z}}(\boldsymbol{z}_n)\, q_{\alpha}(\alpha_n).$$

Approximating the posterior distribution

Challenge: The posterior is intractable, we propose a variational approximation:
(in this case, marginalisation w.r.t. $\boldsymbol{s}_n$ does not simplify things)

$$p(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \mathbf{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0) \approx q(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n) = q_{\boldsymbol{s}}(\boldsymbol{s}_n)\, q_{\boldsymbol{z}}(\boldsymbol{z}_n)\, q_{\alpha}(\alpha_n).$$

The optimal values for the variational distributions [Bishop, 2006] are obtained with:

VE $\boldsymbol{s}_n$-step: $$q_{\boldsymbol{s}}(\boldsymbol{s}_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{z}}(\boldsymbol{z}_n) q_{\alpha}(\alpha_n)}\left[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\right]\right)$$

VE α_n-step: $$q_{\alpha}(\alpha_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{s}}(\boldsymbol{s}_n) q_{\boldsymbol{z}}(\boldsymbol{z}_n)}\left[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\right]\right)$$

VE $\boldsymbol{z}_n$-step: $$q_{\boldsymbol{z}}(\boldsymbol{z}_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{s}}(\boldsymbol{s}_n) q_{\alpha}(\alpha_n)}\left[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\right]\right)$$

Do we have any hope for $\boldsymbol{z}_n$?

Approximating the posterior distribution

Challenge: The posterior is intractable, we propose a variational approximation:
(in this case, marginalisation w.r.t. $\boldsymbol{s}_n$ does not simplify things)

$$p(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \mathbf{x}_n, \boldsymbol{v}_n; \boldsymbol{\psi}^0) \approx q(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n) = q_{\boldsymbol{s}}(\boldsymbol{s}_n)\, q_{\boldsymbol{z}}(\boldsymbol{z}_n)\, q_\alpha(\alpha_n).$$

The optimal values for the variational distributions [Bishop, 2006] are obtained with:

VE $\boldsymbol{s}_n$-step: $$q_{\boldsymbol{s}}(\boldsymbol{s}_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{z}}(\boldsymbol{z}_n) q_\alpha(\alpha_n)}\Big[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\Big]\right)$$

VE α_n-step: $$q_\alpha(\alpha_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{s}}(\boldsymbol{s}_n) q_{\boldsymbol{z}}(\boldsymbol{z}_n)}\Big[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\Big]\right)$$

VE $\boldsymbol{z}_n$-step: $$q_{\boldsymbol{z}}(\boldsymbol{z}_n) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{s}}(\boldsymbol{s}_n) q_\alpha(\alpha_n)}\Big[\log p(\mathbf{x}_n, \boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n; \boldsymbol{\psi}^0)\Big]\right)$$

Do we have any hope for $\boldsymbol{z}_n$? Clearly not! $\rightarrow$ Sampling (from $q_{\boldsymbol{z}}(\boldsymbol{z}_n)$): $\boldsymbol{z}_n^{(r)}$.

Variational E-steps

VE α_n-step: It *follows* that q_α is a Bernoulli distribution with parameter:
($g(x) = 1/(1+\exp(-x))$ denotes the sigmoid function)

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

Audio (numerator) *vs.* audio-visual (denominator).

Variational E-steps

VE α_n-step: It *follows* that q_α is a Bernoulli distribution with parameter:

($g(x) = 1/(1+\exp(-x))$ denotes the sigmoid function)

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

Audio (numerator) *vs.* audio-visual (denominator).

VE $\boldsymbol{s}_n$-step (speech estimation), $q_{\boldsymbol{s}}$ is a complex Gaussian with:

$$\hat{\boldsymbol{s}}_n = \frac{1}{R}\sum_{r=1}^{R}\frac{\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n)}{\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n) + \mathbf{W}_b^*\mathbf{H}_b^*[;, n]}\boldsymbol{x}_n \qquad \Big(\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n) \leftrightarrow \boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n)\Big)$$

Variational E-steps

VE α_n-step: It *follows* that q_α is a Bernoulli distribution with parameter:
$(g(x) = 1/(1+\exp(-x))$ denotes the sigmoid function)

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

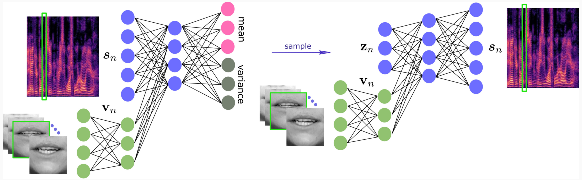
Audio (numerator) *vs.* audio-visual (denominator).

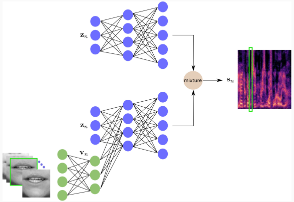
VE $\boldsymbol{s}_n$-step (speech estimation), $q_{\boldsymbol{s}}$ is a complex Gaussian with:

$$\hat{\boldsymbol{s}}_n = \frac{1}{R}\sum_{r=1}^{R}\frac{\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n)}{\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n) + \mathbf{W}_b^*\mathbf{H}_b^*[;,n]}\boldsymbol{x}_n \qquad \Big(\boldsymbol{\gamma}_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n) \leftrightarrow \boldsymbol{\sigma}_s(\mathbf{z}_n^{(r)}, \boldsymbol{v}_n)\Big)$$

with $\boldsymbol{\gamma}_n(\boldsymbol{z}_n, \boldsymbol{v}_n) = \left(\pi_n\frac{1}{\boldsymbol{\sigma}_s^a(\boldsymbol{z}_n)} + (1-\pi_n)\frac{1}{\boldsymbol{\sigma}_s^{av}(\boldsymbol{z}_n, \boldsymbol{v}_n)}\right)^{-1}.$

Summary so far





Conditional VAE:

- Training VAE via SGD.
- Learning noise parameters via MCEM.

VAE-MM:

- Training two VAEs via SGD.
- Learning noise parameters via VEM.

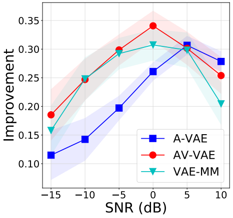
Mixtures of VAEs

Experiments

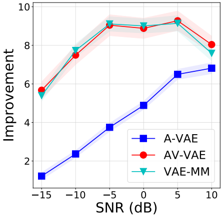
Settings & Results

- **Noisy+clean speech**: NTCD-TIMIT database [Abdelaziz, 2017]
- **VAE models**: Pre-trained A-VAE and AV-VAE [Sadeghi et al., 2019]
- **Setup**: Very similar than in the previous experiments. Clean and noisy lips region visual information ($\sim$ one-third of total video frames per sample).

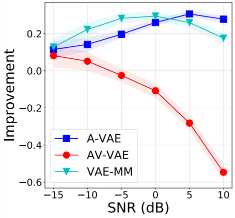
Improvement with respect to the input:



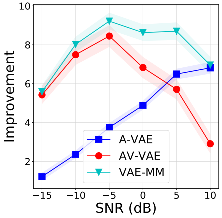
(a) PESQ (clean)



(b) SDR (clean)



(c) PESQ (noisy)

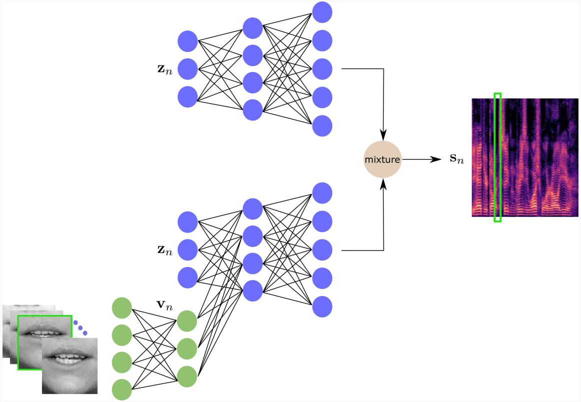


(d) SDR (noisy)

Two different models for the clean speech!

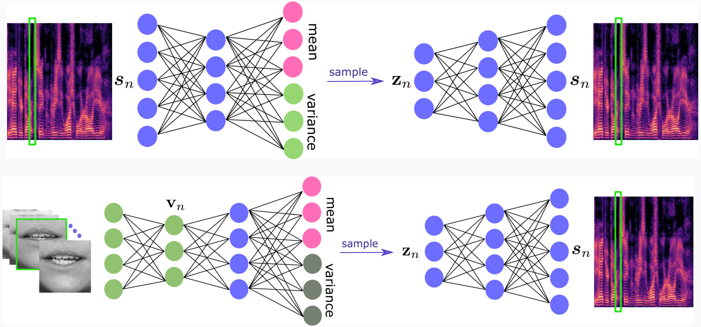
VAE-MM can effectively choose when to use visual information.

However, it has two different probabilistic models for the SAME speech signal!



Mixture of Inference Networks VAE

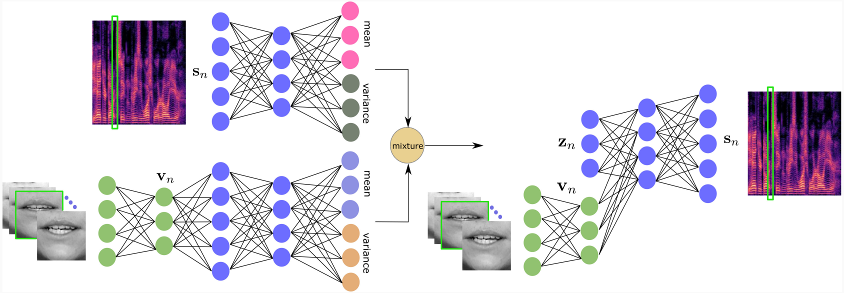
Motivation



How to exploit the mixture model mechanism while having a single generative model for the clean speech signal?

Mixture of Inference Networks VAE (MIN-VAE) [Sadeghi et al., 2021]

Train a mixture of audio and visual inference networks:



- 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

Mixture of Inference Networks: Generative Model

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

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

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

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

- Each latent code, $\boldsymbol{z}_n$, 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 latent variable $\alpha_n \in \{0, 1\}$ describes whether $\boldsymbol{z}_n$ corresponds to the audio or to the video prior.

Mixture of Inference Networks VAE

1. Learn a deep generative model

Training MIN-VAE

- Observed variables: $\mathbf{s} = \{\boldsymbol{s}_n\}_{n=1}^{N_{tr}}$ and $\mathbf{v} = \{\boldsymbol{v}_n\}_{n=1}^{N_{tr}}$
- Latent variables: $\mathbf{z} = \{\boldsymbol{z}_n\}_{n=1}^{N_{tr}}$ and $\boldsymbol{\alpha} = \{\alpha_n\}_{n=1}^{N_{tr}}$
- Parameters to be estimated: $\underbrace{\boldsymbol{\theta}}_{\text{decoder}}, \underbrace{\boldsymbol{\phi}_a, \boldsymbol{\phi}_v,}_{\text{encoders}} \underbrace{\boldsymbol{\mu}_a, \sigma_a}_{\text{audio prior}}, \underbrace{\boldsymbol{\mu}_v, \sigma_v}_{\text{visual prior}}, \underbrace{\pi}_{\text{mixing prior}}$

We target a lower bound on the data log-likelihood:

$$\sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n, \alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n)} [\log p(\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n | \boldsymbol{v}_n)]$$

What is going to happen to the posterior $p(\boldsymbol{z}_n, \alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n)$?

Training MIN-VAE: approximating the posterior

Posterior distribution of the latent variables:

$$p(\boldsymbol{z}_n, \alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n) = \underbrace{p(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{v}_n, \alpha_n)}_{\text{Code posterior}} \cdot \underbrace{p(\alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n)}_{\text{Mixing posterior}}.$$

The two posteriors in the RHS are intractable.

Training MIN-VAE: approximating the posterior

Posterior distribution of the latent variables:

$$p(\boldsymbol{z}_n, \alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n) = \underbrace{p(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{v}_n, \alpha_n)}_{\text{Code posterior}} \cdot \underbrace{p(\alpha_n | \boldsymbol{s}_n, \boldsymbol{v}_n)}_{\text{Mixing posterior}}.$$

The two posteriors in the RHS are intractable.

We assume the following variational approximation for the codes:

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

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

Final approximation:

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

Training MIN-VAE: the algorithm

We can obtain a variational EM (details omited):

- E-step: $q(\alpha_n)$ is a Bernoulli distribution with parameter:

$$\pi_n = \pi \cdot g\Big(J_n(\alpha_n = 1) - J_n(\alpha_n = 0)\Big),$$

  where $J_n(\alpha_n)$ is the VAE cost of representing the n-th vector with audio ($\alpha_n = 1$) or video ($\alpha_n = 0$) encoding (g is the sigmoid).

Training MIN-VAE: the algorithm

We can obtain a variational EM (details omited):

- E-step: $q(\alpha_n)$ is a Bernoulli distribution with parameter:

$$\pi_n = \pi \cdot g\Big(J_n(\alpha_n = 1) - J_n(\alpha_n = 0)\Big),$$

  where $J_n(\alpha_n)$ is the VAE cost of representing the n-th vector with audio ($\alpha_n = 1$) or video ($\alpha_n = 0$) encoding (g is the sigmoid).

- M-step: the parameters are updated by maximizing:

$$\sum_{n=1}^{N_{tr}} \mathbb{E}_{q(\alpha_n)}\Big[J_n(\alpha_n)\Big] - D_{\mathsf{KL}}\Big(q\left(\alpha_n\right) \parallel p(\alpha_n)\Big)$$

  $\pi = \dfrac{1}{N_{tr}} \displaystyle\sum_{n=1}^{N_{tr}} \pi_n$ and SGD for $\boldsymbol{\theta}, \boldsymbol{\phi}_a, \boldsymbol{\phi}_v, \boldsymbol{\mu}_a, \boldsymbol{\mu}_v, \sigma_a, \sigma_v$.

Mixture of Inference Networks VAE

2. & 3. Estimate the noise parameters from x

Parameter and Speech Estimation

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: Trained MIN-VAE

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

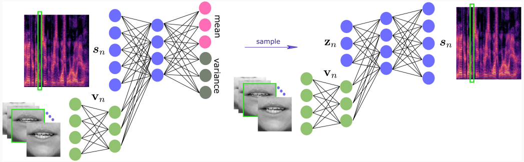
$$\boldsymbol{z}_n | \alpha_n \sim \Big[\mathcal{N}(\boldsymbol{\mu}_a, \sigma_a \mathbf{I})\Big]^{\alpha_n} \cdot \Big[\mathcal{N}(\boldsymbol{\mu}_v, \sigma_v \mathbf{I})\Big]^{1-\alpha_n}$$

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

Inference: (we skip the VEM as it follows a similar principle than the previous one)

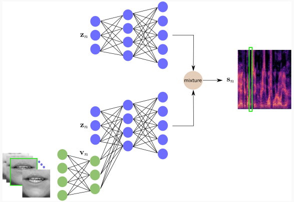
- ▷ Observed variables: $\{\mathbf{x}_n, \boldsymbol{v}_n\}_{n=1}^N$
- ▷ Latent variables: $\Big\{\boldsymbol{s}_n, \boldsymbol{z}_n, \alpha_n\Big\}_{n=1}^N$
- ▷ Parameters to be estimated: $\boldsymbol{\psi} = \{\mathbf{W}_b, \mathbf{H}_b, \pi\}$

Summary of the three models



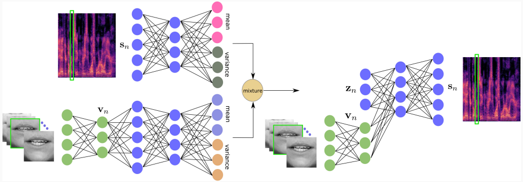
Conditional VAE:

- Training VAE via SGD.
- Systematic AV fusion.
- Learning noise parameters via MCEM.



VAE-MM:

- Training two VAEs via SGD.
- Mixing AV fusion - two speech models.
- Learning noise parameters via VEM.



MIN-VAE:

- Training all 3 networks via VEM+SGD.
- Mixing AV fusion - single speech model.
- Learning noise parameters via VEM.

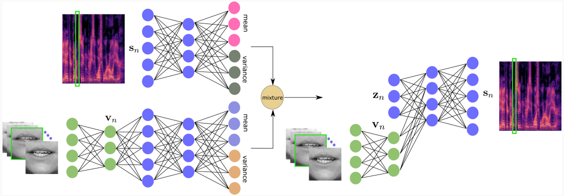
Mixture of Inference Networks VAE

Experiments

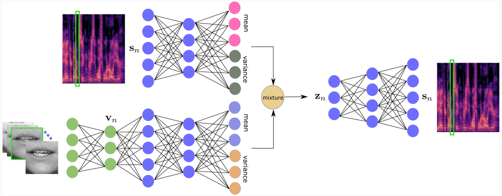
Settings

Settings similar as before with the NTCD-TIMIT dataset. Two possible models.

MIN-VAE-v1:

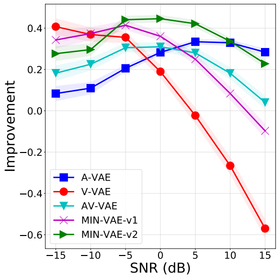


MIN-VAE-v2:

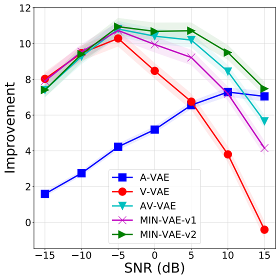


Results

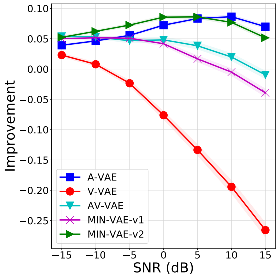
Adding noise to one-third of spectrograms input to audio-encoder:



(a) PESQ



(b) SDR



(c) STOI

Conclusion

- Variational autoencoders are an excellent framework for unsupervised audio-visual SE.
- However, (V)EM-based inference may be slow. Re-using the trained encoders is an alternative.
- Extending the mixture VAE to more than two encoders, to include other sources of information.
- Huge limitation so far: STFT speech frames are processed independently (not sequentially).

Lecture 3: Dynamical variational autoencoders (VAE's that can process sequences).

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.
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*," IEEE TASLP, 2020.
5. M. Sadeghi and X. Alameda-Pineda, "*Robust Unsupervised Audio-Visual Speech Enhancement using a Mixture of VAEs*," IEEE ICASSP 2020.
6. M. Sadeghi and X. Alameda-Pineda, "*Mixture of Inference Networks for VAE-based Audio-visual Speech Enhancement*", IEEE TST, 2021.
7. C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag Berlin, Heidelberg, 2006.
8. A. H. Abdelaziz, "*NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition*," in Proc. INTERSPEECH, 2017.