Variational Audio-Visual Representation Learning

Xavier Alameda-Pineda (Xavi)

Keynote Talk @ ACM Multimedia Ottawa, 1st November 2023

Why Audio-Visual Unsupervised Representation Learning?

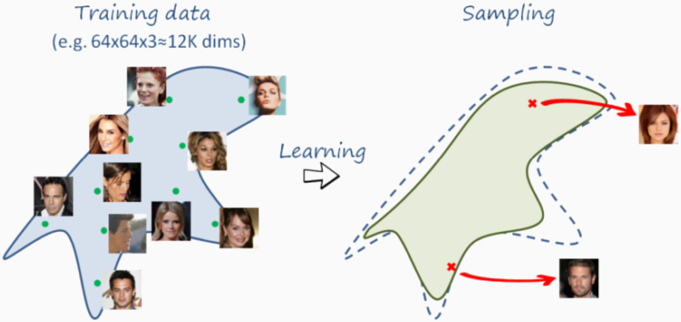
?

[Images under Creative Commons license]

- Audio: wide field of hearing, only usable when sources are active.
- Video: limited field of view, rich flow of information.
- Unsupervised: avoid the need for human labels (in new environments).

And probabilistic learning?

Probabilistic generative models aim to learn a (parametric) *distribution* $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ that approximates the complex data distribution $p_{\text{data}}(\boldsymbol{x})$:



- We can jointly learn them with other probabilistic models using *maximum likelihood*.

Disclaimer: Halloween is not over

Attendee Discretion is Advised: Scary Equations Ahead

The Kullback-Leibler divergence and the ML formulation

The Kullback-Leilbler (KL) divergence between two distributions writes:

$$D_{\mathsf{KL}}(p(\boldsymbol{x})\|q(\boldsymbol{x})) = -\mathbb{E}_{p(\boldsymbol{x})}\left[\log\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\right] = -\int_{\mathcal{X}} p(\boldsymbol{x})\log\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} \begin{cases} \geq 0 \\ 0 \Leftrightarrow p(\boldsymbol{x}) = q(\boldsymbol{x}) \\ \neq D_{\mathsf{KL}}(q(\boldsymbol{x})\|p(\boldsymbol{x})) \end{cases}$$

The Kullback-Leibler divergence and the ML formulation

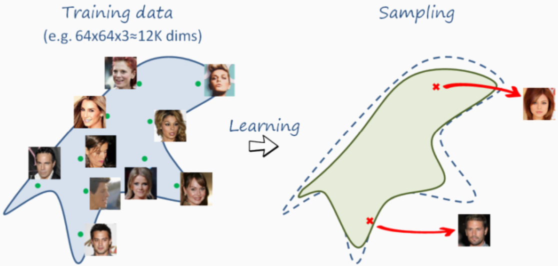
The Kullback-Leilbler (KL) divergence between two distributions writes:

$$D_{\mathsf{KL}}(p(\boldsymbol{x}) \| q(\boldsymbol{x})) = -\mathbb{E}_{p(\boldsymbol{x})}\left[\log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\right] = -\int_{\mathcal{X}} p(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \begin{cases} \geq 0 \\ 0 \Leftrightarrow p(\boldsymbol{x}) = q(\boldsymbol{x}) \\ \neq D_{\mathsf{KL}}(q(\boldsymbol{x}) \| p(\boldsymbol{x})) \end{cases}$$

Given a training set $\{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{x}_i \sim p_{\text{data}}(\boldsymbol{x})$,
ML minimizes the KL divergence:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \; D_{\mathsf{KL}}\Big(p_{\text{data}}(\boldsymbol{x}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x})\Big)$$

$$\approx \boxed{\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; \frac{1}{N} \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}$$



ML formulation with latent variables

ML with latent variable ($\boldsymbol{z}$) leads to EM[1] and VI[2] build from ($q(\boldsymbol{z})$ is an arbitrary distribution):

$$\log p(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})}\right]}_{\text{M-step or VLB}} + \underbrace{D_{\mathsf{KL}}\left(q(\boldsymbol{z})\middle\|p(\boldsymbol{z}|\boldsymbol{x})\right)}_{\text{E-step}}$$

Exact EM

Simple $p(\boldsymbol{z}|\boldsymbol{x})$

$q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x})$

Closed-form!

$D_{\mathsf{KL}} = 0$

[1] Dempster, A.P., *et. al.*, (1977), Journal of the Royal Statistical Society.

[2] Jordan, M. I., *et. al.*, (1999),. Machine Learning.

ML formulation with latent variables

ML with latent variable ($\boldsymbol{z}$) leads to EM[1] and VI[2] build from ($q(\boldsymbol{z})$ is an arbitrary distribution):

$$\log p(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})}\right]}_{\text{M-step or VLB}} + \underbrace{D_{\mathsf{KL}}\Big(q(\boldsymbol{z})\Big\|p(\boldsymbol{z}|\boldsymbol{x})\Big)}_{\text{E-step}}$$

<u>Exact EM</u>	<u>Variational EM</u>
Simple $p(\boldsymbol{z}\|\boldsymbol{x})$	$p(\boldsymbol{z}\|\boldsymbol{x})$ too complex
$q(\boldsymbol{z}) = p(\boldsymbol{z}\|\boldsymbol{x})$	$q(\boldsymbol{z}) = q_1(\boldsymbol{z}_1)q_2(\boldsymbol{z}_2)$
Closed-form!	$q_1, q_2 = \operatorname{argmin} D_{\mathsf{KL}}$
$D_{\mathsf{KL}} = 0$	$D_{\mathsf{KL}} > 0$ but closed form!

[1] Dempster, A.P., *et. al.*, (1977), Journal of the Royal Statistical Society.
[2] Jordan, M. I., *et. al.*, (1999),. Machine Learning.

ML formulation with latent variables

ML with latent variable ($\boldsymbol{z}$) leads to EM[1] and VI[2] build from ($q(\boldsymbol{z})$ is an arbitrary distribution):

$$\log p(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})}\right]}_{\text{M-step or VLB}} + \underbrace{D_{\mathsf{KL}}\Big(q(\boldsymbol{z})\Big\|p(\boldsymbol{z}|\boldsymbol{x})\Big)}_{\text{E-step}}$$

Exact EM	Variational EM	Variational AutoEncoder
Simple $p(\boldsymbol{z}\|\boldsymbol{x})$	$p(\boldsymbol{z}\|\boldsymbol{x})$ too complex	$p(\boldsymbol{z}\|\boldsymbol{x})$???
$q(\boldsymbol{z}) = p(\boldsymbol{z}\|\boldsymbol{x})$	$q(\boldsymbol{z}) = q_1(\boldsymbol{z}_1)q_2(\boldsymbol{z}_2)$	$q(\boldsymbol{z}) = q_{\boldsymbol{\phi}}(\boldsymbol{z})$
Closed-form!	$q_1, q_2 = \operatorname{argmin} D_{\mathsf{KL}}$	$q_{\boldsymbol{\phi}}$ optimises ELBO/VLB
$D_{\mathsf{KL}} = 0$	$D_{\mathsf{KL}} > 0$ but closed form!	$D_{\mathsf{KL}} > 0$ no closed form!

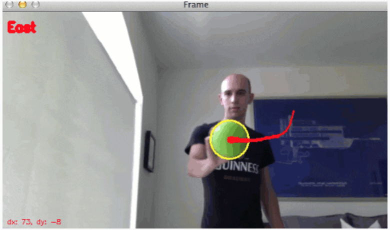
[1] Dempster, A.P., *et. al.*, (1977), Journal of the Royal Statistical Society.

[2] Jordan, M. I., *et. al.*, (1999),. Machine Learning.

Interest of Latent Variables

Speech Enhancement (noisy and clean signals)

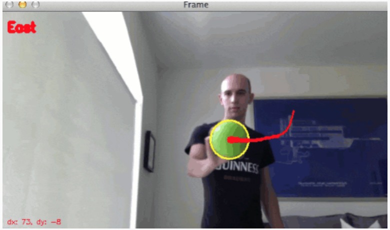
Visual Tracking (detections, true position)



Interest of Latent Variables

Speech Enhancement (noisy and clean signals)

Visual Tracking (detections, true position)



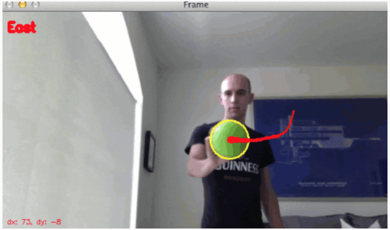
1. Define the model:

$$\begin{cases} p_{\boldsymbol{\theta}}(\boldsymbol{s}) \;\&\; p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{s}) & \text{(prior \& likelihood)} \\ q_{\boldsymbol{\phi}}(\boldsymbol{s}) \approx p_{\boldsymbol{\theta}}(\boldsymbol{s}|\boldsymbol{x}) & \text{(approx. posterior)} \end{cases}$$

Interest of Latent Variables

Speech Enhancement (noisy and clean signals)

Visual Tracking (detections, true position)



1. Define the model:

$$\begin{cases} p_{\boldsymbol{\theta}}(\boldsymbol{s}) \;\&\; p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{s}) & \text{(prior \& likelihood)} \\ q_{\boldsymbol{\phi}}(\boldsymbol{s}) \approx p_{\boldsymbol{\theta}}(\boldsymbol{s}|\boldsymbol{x}) & \text{(approx. posterior)} \end{cases}$$

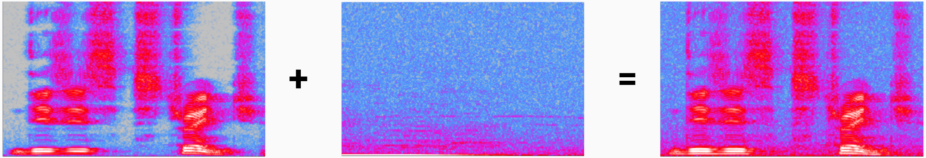
2. Learning & inference:

$$\underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmax}}\, \mathcal{L}_{\mathsf{ELBO}}(\boldsymbol{\theta},\boldsymbol{\phi})$$

$$\underset{\boldsymbol{s}}{\operatorname{argmax}}\, q_{\boldsymbol{\phi}^*}(\boldsymbol{s}|\boldsymbol{x})$$

Speech Enhancement & Wiener Filter

Extract the latent clean speech signal from the observed noisy mixture. (STFT domain)



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

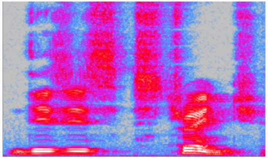
Speech Enhancement & Wiener Filter

Minimize MSE $\rightarrow$ Wiener Filter

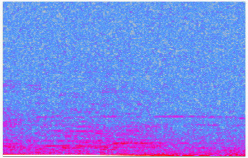
(operations are element-wise)

$$\hat{\boldsymbol{s}} = \frac{\boldsymbol{\sigma_s}}{\boldsymbol{\sigma_s} + \boldsymbol{\sigma_b}} \boldsymbol{x}$$

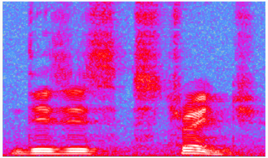
Extract the latent clean speech signal from the observed noisy mixture. (STFT domain)



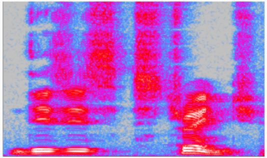
+



=



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



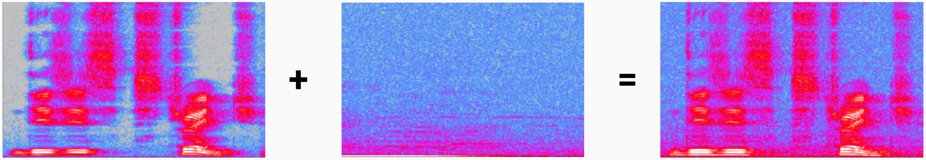
$$\underbrace{\mathbf{s}}_{\text{Clean speech}}$$

Train – Model for clean data: $\{\boldsymbol{s}_i\}_{i=1}^{N}$.

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

Then freeze $\boldsymbol{\theta}$ at test/adaptation time.

Unsupervised Probabilistic SE: paradigm



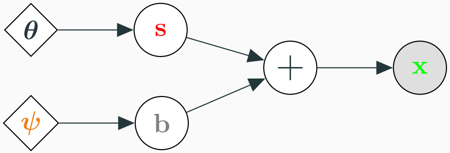
$$\underbrace{\mathbf{s}}_{\text{Clean speech}} + \underbrace{\mathbf{b}}_{\text{Noise signal}} = \underbrace{\mathbf{x}}_{\text{Noisy mixture}}$$

Train – Model for clean data: $\{\boldsymbol{s}_i\}_{i=1}^N$.

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

Then freeze $\boldsymbol{\theta}$ at test/adaptation time.

Test – learn the noise parameters from noisy samples $\mathbf{x}$ and estimate the clean speech $\hat{\boldsymbol{s}}$:



Audio-visual Speech Enhancement (AV-SE)

- Visual data (lip motion) provide complementary information about the unknown speech.
- For highly noisy audio recordings, visual information can be very helpful.

We investigate the VAE framework to fuse audio and visual data for speech enhancement.

▶ **Can we jointly learn from AV data for SE?**

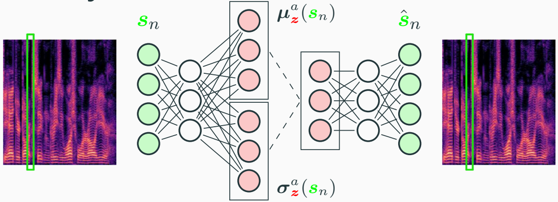
Mostafa Sadeghi

Laurent Girin

Radu Horaud

Mono-modal VAEs: the baselines

Audio-only:[3]



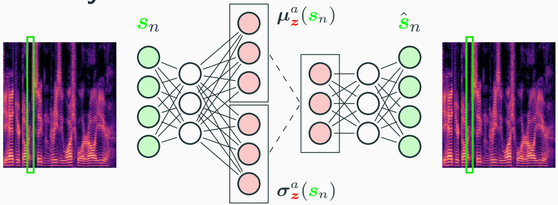
$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_n|\boldsymbol{z}_n) = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{s}}(\boldsymbol{z}_n))\Big)$$

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{s}_n) = \mathcal{N}\Big(\boldsymbol{\mu}^a_{\boldsymbol{z}}(\boldsymbol{s}_n), \mathrm{diag}(\boldsymbol{\sigma}^a_{\boldsymbol{z}}(\boldsymbol{s}_n))\Big)$$

[3] Leglaive, S., *et. al.*, (2018), IEEE MLSP.

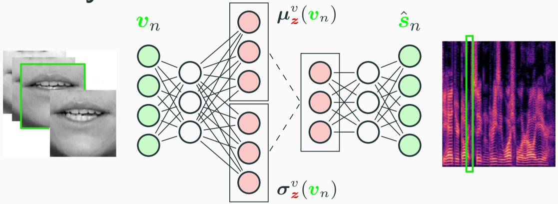
Mono-modal VAEs: the baselines

Audio-only:[3]



$$p_{\theta}(s_n|z_n) = \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\sigma_s(z_n))\Big)$$
$$q_{\phi}(z_n|s_n) = \mathcal{N}\Big(\mu_z^a(s_n), \text{diag}(\sigma_z^a(s_n))\Big)$$

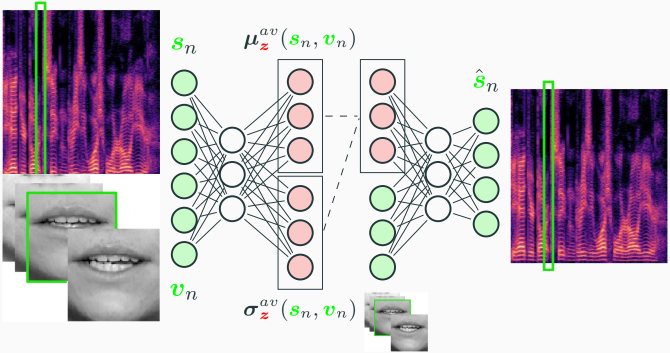
Video-only:



$$p_{\theta}(s_n|z_n) = \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\sigma_s(z_n))\Big)$$
$$q_{\phi}(z_n|v_n) = \mathcal{N}\Big(\mu_z^v(v_n), \text{diag}(\sigma_z^v(v_n))\Big)$$

[3] Leglaive, S., *et. al.*, (2018), IEEE MLSP.

Audio-visual Conditional VAE[4]

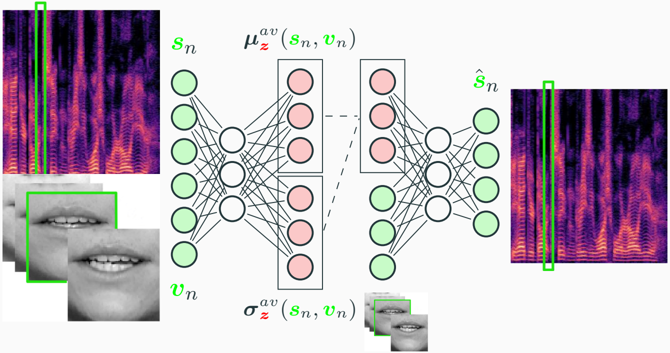


$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_n | \boldsymbol{z}_n, \boldsymbol{v}_n) = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{s}}(\boldsymbol{z}_n, \boldsymbol{v}_n))\Big)$$

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

[4] Sadeghi, M., *et. al.*, (2020), IEEE TASLP.

Audio-visual Conditional VAE[4]



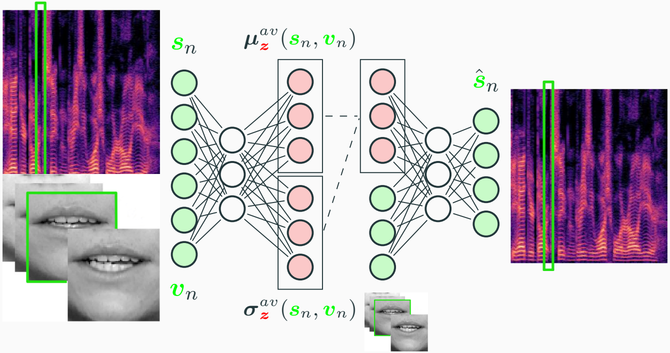
What will this learn?

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

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

[4] Sadeghi, M., *et. al.*, (2020), IEEE TASLP.

Audio-visual Conditional VAE[4]



What will this learn?

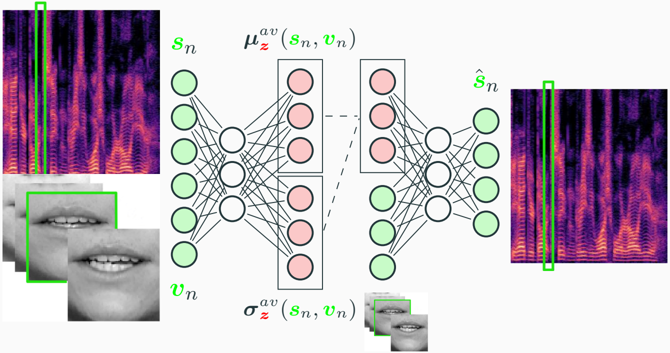
To ignore the video!

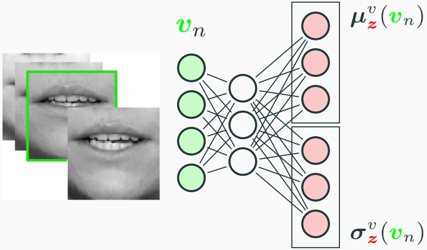
$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_n | \boldsymbol{z}_n, \boldsymbol{v}_n) = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_{s}(\boldsymbol{z}_n, \boldsymbol{v}_n))\Big)$$

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

[4] Sadeghi, M., *et. al.*, (2020), IEEE TASLP.

Audio-visual Conditional VAE[4]





$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_n|\boldsymbol{z}_n, \boldsymbol{v}_n) = \mathcal{N}_c\Big(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{s}}(\boldsymbol{z}_n, \boldsymbol{v}_n))\Big)$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_n|\boldsymbol{v}_n) = \mathcal{N}\Big(\boldsymbol{\mu}_{\boldsymbol{z}}^{v}(\boldsymbol{v}_n), \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{z}}^{v}(\boldsymbol{v}_n))\Big)$$

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{v}_n, \boldsymbol{s}_n) = \mathcal{N}\Big(\boldsymbol{\mu}_{\boldsymbol{z}}^{av}(\boldsymbol{v}_n, \boldsymbol{s}_n), \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{z}}^{av}(\boldsymbol{v}_n, \boldsymbol{s}_n))\Big)$$

[4] Sadeghi, M., *et. al.*, (2020), IEEE TASLP.

Learning AV Conditional VAE

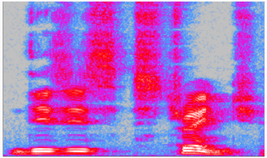
On top of adding a video-only prior, we optimize a modified ELBO:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_n (1-\alpha)\Big(\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{s}_n,\boldsymbol{v}_n;)}\Big[\ln p_{\boldsymbol{\theta}}(\boldsymbol{s}_n|\boldsymbol{z}_n,\boldsymbol{v}_n)\Big] - D_{\mathsf{KL}}\Big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{s}_n,\boldsymbol{v}_n) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z}_n|\boldsymbol{v}_n)\Big)\Big)$$
$$+\alpha\,\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}_n|\boldsymbol{v}_n)}\Big[\ln p_{\boldsymbol{\theta}}(\boldsymbol{s}_n|\boldsymbol{z}_n,\boldsymbol{v}_n)\Big]$$

$\Rightarrow \alpha > 0$ gives some reconstruction power to the visual prior!

This provides the parameters of the clean speech model $\boldsymbol{\theta}$. Let's enhance!

AV Conditional VAE for Speech Enhancement



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

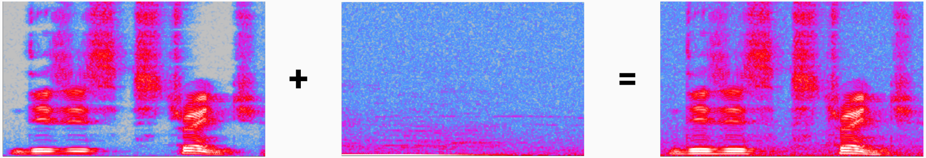
Train – Model for clean data: $\{\boldsymbol{s}_i\}_{i=1}^{N}$.

$$\boldsymbol{\theta}, \boldsymbol{v} \longrightarrow \boldsymbol{s}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{s}|\boldsymbol{v}) \approx p_{\text{data}}(\boldsymbol{s}|\boldsymbol{v})$$

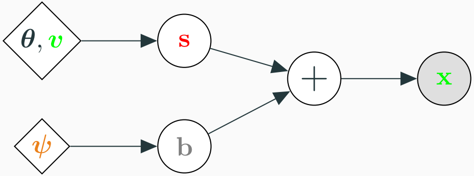
Then freeze $\boldsymbol{\theta}$ at test/adaptation time.

AV Conditional VAE for Speech Enhancement

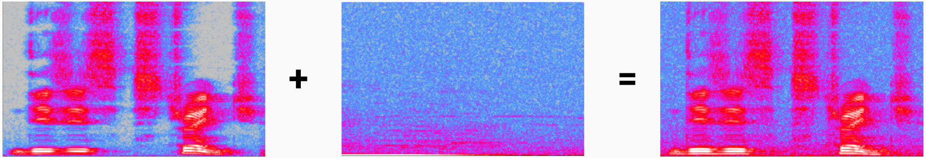


$$\underbrace{\mathbf{s}}_{\text{Clean speech}} + \underbrace{\mathbf{b}}_{\text{Noise signal}} = \underbrace{\mathbf{x}}_{\text{Noisy mixture}}$$

Test – learn the noise parameters from noisy samples $\mathbf{x}$ and estimate the clean speech $\hat{\boldsymbol{s}}$:



AV Conditional VAE for Speech Enhancement



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

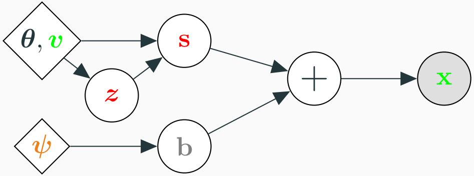
Speech Enhancement:

(Sample $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{x}_n, \boldsymbol{v}_n)$)

$$\hat{\boldsymbol{s}}_n = \left(\frac{1}{R} \sum_{r=1}^{R} \frac{\boldsymbol{\sigma}_s(\mathbf{z}^{(r)}, \boldsymbol{v}_n)}{\boldsymbol{\sigma}_s(\mathbf{z}^{(r)}, \boldsymbol{v}_n) + \psi_n^*} \right) \boldsymbol{x}_n.$$

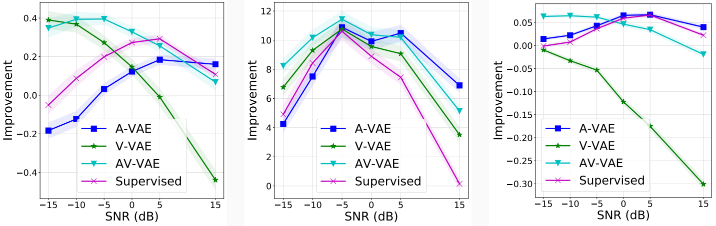
Wiener-like filter!

Test – learn the noise parameters from noisy samples $\mathbf{x}$ and estimate the clean speech $\hat{\boldsymbol{s}}$:



Experiments & Discussion

NTCD-TIMIT dataset:[5] AV recordings in controlled conditions, 5h/39 speakers training, 1h/9 speakers test, several noise levels (-15 to 15 dB) and types (car, living room, etc).
Metrics improvement w.r.t. the noisy mixture: perceptual evaluation of speech quality (PESQ), signal-to-distortion ratio (SDR), Short-time objective intelligibility (STOI).



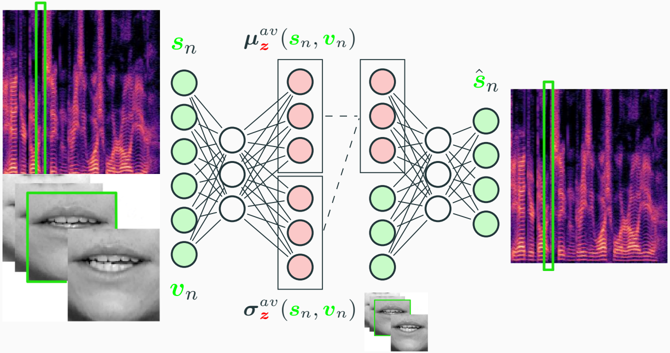
(a) PESQ **(b)** SDR **(c)** STOI

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

[5] Abdelaziz, A.H., (2017), Interspeech.

Limitations: systematic AV fusion

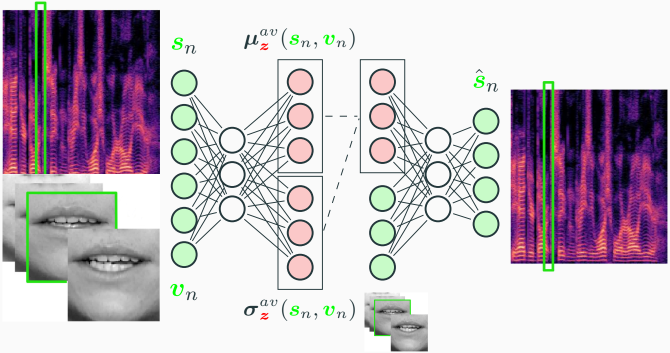
From the model design:



.

Limitations: systematic AV fusion

From the model design:



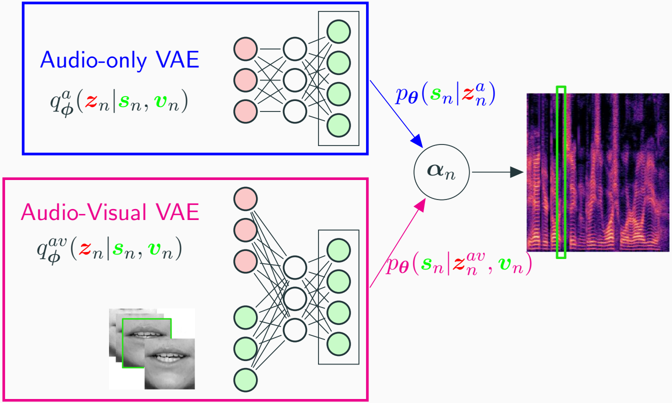
. What about clutter?

We ALWAYS concatenate audio and video information.

- **Can we unsupervisedly select what to use?**

Mostafa Sadeghi

VAE Mixture Model[6]



α_n is the "mixing" latent variable.

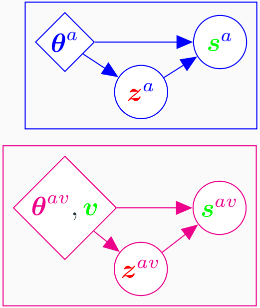
$$\begin{cases} \alpha_n = 1 \leftrightarrow \text{Audio-only} \\ \alpha_n = 0 \leftrightarrow \text{Audio-Visual} \end{cases}$$

Prior: $p(\alpha_n) = \pi^{\alpha_n}(1-\pi)^{1-\alpha_n}$.

The Audio-only and the Audio-Visual VAE are **mixed without supervision**.

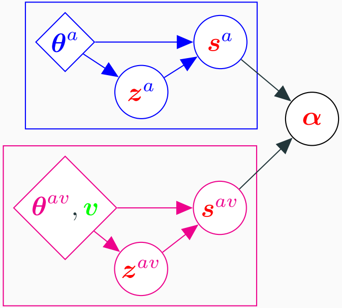
[6]Sadeghi, M., *et. al.*, (2021), IEEE ICASSP.

Train, adaptation & speech enhancement



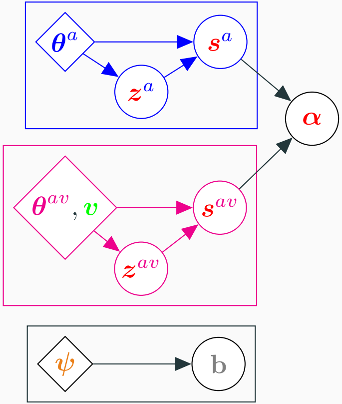
- Train A-VAE (with $\boldsymbol{s}$) and AV-VAE (with $(\boldsymbol{s}, \boldsymbol{v})$).

Train, adaptation & speech enhancement



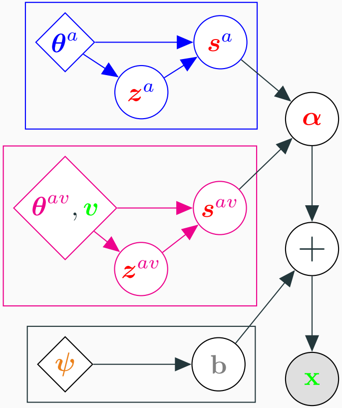
- Train A-VAE (with $\boldsymbol{s}$) and AV-VAE (with $(\boldsymbol{s}, \boldsymbol{v})$).
- The clean speeh is now latent $(\boldsymbol{s}^a, \boldsymbol{s}^{av})$, add the mixing latent variable $(\boldsymbol{\alpha})$.

Train, adaptation & speech enhancement



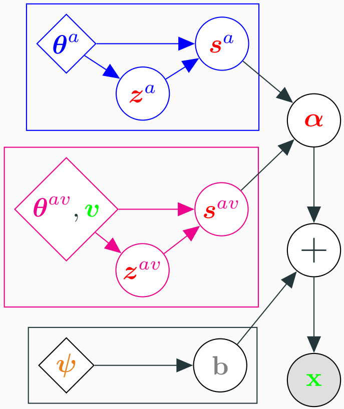
- Train A-VAE (with s) and AV-VAE (with (s, v)).
- The clean speeh is now latent (s^a, s^{av}), add the mixing latent variable (α).
- We also consider the noise variable and params.

Train, adaptation & speech enhancement



- Train A-VAE (with $\boldsymbol{s}$) and AV-VAE (with $(\boldsymbol{s}, \boldsymbol{v})$).
- The clean speeh is now latent $(\boldsymbol{s}^a, \boldsymbol{s}^{av})$, add the mixing latent variable $(\boldsymbol{\alpha})$.
- We also consider the noise variable and params.
- Learn ψ^* and estimate the clean speech. By defining $\gamma_n(\boldsymbol{z}_n, \boldsymbol{v}_n) = \left(\pi_n(\boldsymbol{\sigma}_s^a(\boldsymbol{z}_n))^{-1} + (1 - \pi_n)(\boldsymbol{\sigma}_s^{av}(\boldsymbol{z}_n, \boldsymbol{v}_n))^{-1}\right)^{-1}$

Train, adaptation & speech enhancement



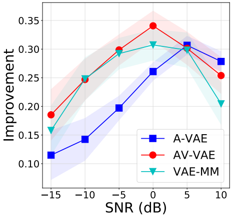
- Train A-VAE (with $\boldsymbol{s}$) and AV-VAE (with $(\boldsymbol{s}, \boldsymbol{v})$).
- The clean speeh is now latent $(\boldsymbol{s}^a, \boldsymbol{s}^{av})$, add the mixing latent variable $(\boldsymbol{\alpha})$.
- We also consider the noise variable and params.
- Learn ψ^* and estimate the clean speech. By defining $\gamma_n(\boldsymbol{z}_n, \boldsymbol{v}_n) = \big(\pi_n(\boldsymbol{\sigma}_s^a(\boldsymbol{z}_n))^{-1} + (1 - \pi_n)(\boldsymbol{\sigma}_s^{av}(\boldsymbol{z}_n, \boldsymbol{v}_n))^{-1}\big)^{-1}$:

$$\hat{\boldsymbol{s}}_n = \frac{1}{R} \sum_{r=1}^{R} \frac{\gamma_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n)}{\gamma_n(\boldsymbol{z}_n^{(r)}, \boldsymbol{v}_n) + \psi_n^*} \boldsymbol{x}_n$$

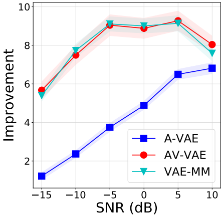
Settings & Results

- **Data & models**: Same dataset + pre-trained A-VAE and AV-VAE.
- **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)

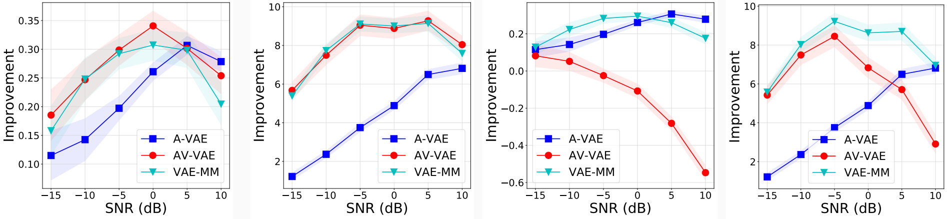


(b) SDR (clean)

Settings & Results

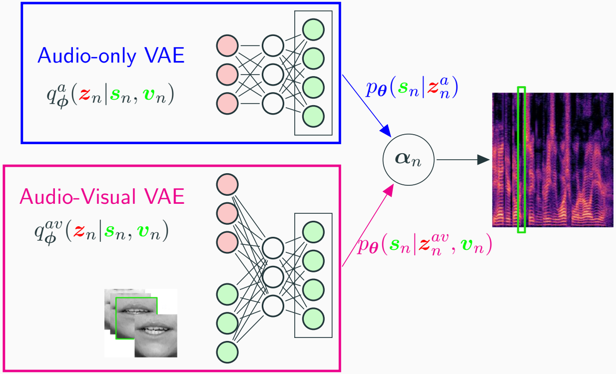
- **Data & models**: Same dataset + pre-trained A-VAE and AV-VAE.
- **Setup**: Very similar than in the previous experiments. Clean and noisy lips region visual information ($\sim$ one-third of total video frames per sample).

Improvement with respect to the input:



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

VAE Mixture Model: 2 models for the same signal?



For the same input $\boldsymbol{s}$, we compute:

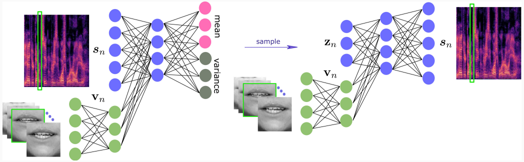
- An audio-based reconstruction $\boldsymbol{s}^a$.
- An AV-based reconstruction $\boldsymbol{s}^{av}$.

Not necessarily equal.

This is strange!!! We have proposed a model with a single decoder.[7]

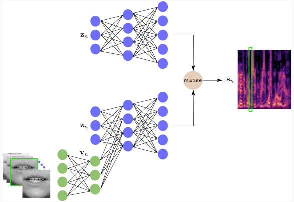
[7] Sadeghi, M., *et. al.*, (2021), IEEE TSP.

Conditional VAE, VAE-MM and Mixture of Inference Networks VAE



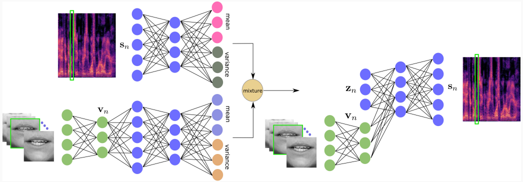
Conditional VAE:

- Training VAE via SGD.
- Systematic AV fusion.
- Appeared at IEEE TASLP in 2020.



VAE-MM:

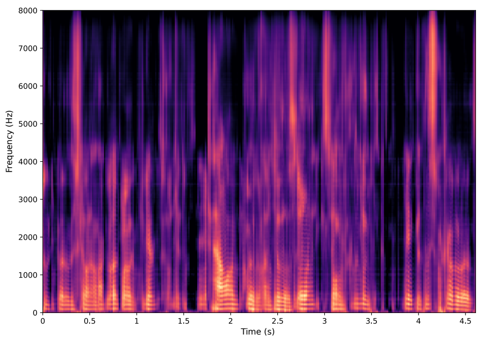
- Training two VAEs via SGD.
- Mixing (AV fusion) via two speech models.
- Appeared at IEEE ICASSP in 2021.

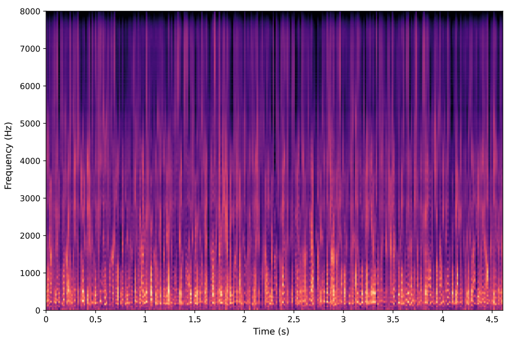


MIN-VAE:

- Training all 3 networks via VEM+SGD.
- Mixing (AV fusion) via a single speech model.
- Appeared at IEEE TSP in 2021.

Strong limitation of VAEs: frame modeled independently.





Frames are modeled independently, we need time/sequential modeling!

- **Dynamical** VAE (**DVAE**[8]) – "VAE for sequential modeling"

(family of methods including existing literature)

Xiaoyu Bie

Laurent Girin

Simon Leglaive

Thomas Hueber

Julien Diard

[8] Girin, L., *et. al.*, (2021), FnT Machine Learning – `Warning` $\sim$ `150 pages!!`

Probabilistic Sequential Modeling (decoder network)

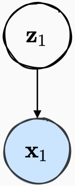
We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

$$p_{\boldsymbol{\theta}}^{\mathrm{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\mathrm{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$

Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

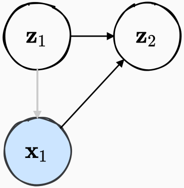
$$p_{\boldsymbol{\theta}}^{\mathrm{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\mathrm{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

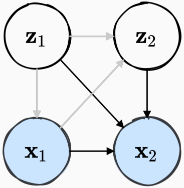
$$p_{\boldsymbol{\theta}}^{\mathrm{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\mathrm{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

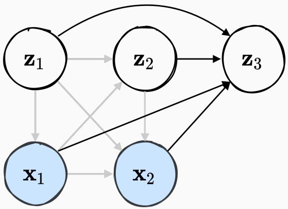
$$p_{\boldsymbol{\theta}}^{\text{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\text{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

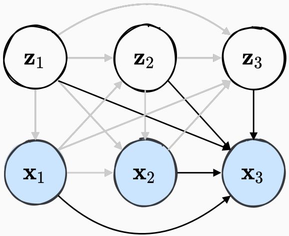
$$p_{\boldsymbol{\theta}}^{\text{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\text{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

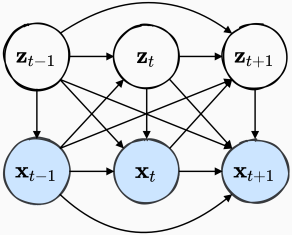
$$p_{\boldsymbol{\theta}}^{\mathrm{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\mathrm{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



Probabilistic Sequential Modeling (decoder network)

We would like to model sequences of observations ($\boldsymbol{x}_{1:T}$) and latent variables ($\boldsymbol{z}_{1:T}$):

$$p_{\boldsymbol{\theta}}^{\mathrm{DVAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) \neq \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_t) = p_{\boldsymbol{\theta}}^{\mathrm{VAE}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}).$$



$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t})$$

The *prior* distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t+1} | \boldsymbol{z}_{1:t}, \boldsymbol{x}_{1:t})$ is now conditional, parametric, and might be auto-regressive (AR).

$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1} | \boldsymbol{z}_{1:t+1}, \boldsymbol{x}_{1:t})$ might be AR as well.

Probabilistic Sequential Inference (encoder network)

As in the VAE, we need to approximate the posterior distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})$$

[9] Bishop, C. M., (2006).

Probabilistic Sequential Inference (encoder network)

As in the VAE, we need to approximate the posterior distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})$$

We can always use the Bayes theorem to write:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:T})$$

Can we simplify each of the terms further? It depends on generative model. Use D-separation[9] to find out the true dependencies, then choose whether to keep them (e.g. to ensure causality).

[9] Bishop, C. M., (2006).

Implementation (both decoder and encoder)

Let's take the general DVAE decoder:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{z}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}) p_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}).$$

[10] Several DVAEs @ `https://github.com/XiaoyuBIE1994/DVAE-speech`

Implementation (both decoder and encoder)

Let's take the general DVAE decoder:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{z}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}) p_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}).$$

The generative distributions can be implemented with an RNN:

- $\boldsymbol{h}_t = \sigma(\boldsymbol{W}_{xh}\boldsymbol{x}_{t-1} + \boldsymbol{W}_{zh}\boldsymbol{z}_{t-1} + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h)$,
- $p_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{z}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}) = \mathcal{N}\Big(\boldsymbol{z}_t; \boldsymbol{\mu}_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{h}_t), \text{diag}\{\boldsymbol{v}_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{h}_t)\}\Big)$,
- $p_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}_t | \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) = \mathcal{N}\Big(\boldsymbol{x}_t; \boldsymbol{\mu}_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{z}_t, \boldsymbol{h}_t), \text{diag}\{\boldsymbol{v}_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{z}_t, \boldsymbol{h}_t)\}\Big)$.

There are many possible implementations[10] for the same probabilistic dependencies!!!

[10] Several DVAEs @ `https://github.com/XiaoyuBIE1994/DVAE-speech`

Learning: ELBO

The objective is build in the same way as VAEs, but looks different:

$$\mathcal{L}(\boldsymbol{x}_{1:T};\boldsymbol{\phi},\boldsymbol{\theta}) \stackrel{?}{=} \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_{1:T})} \underbrace{\ln p_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1},\boldsymbol{z}_{1:t})}_{\text{Reconstruction}}$$
$$- \sum_{t=1}^{T} \underbrace{D_{\mathsf{KL}}\Big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1},\boldsymbol{x}_{1:T}) \,\Big\|\, p_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1},\boldsymbol{z}_{1:t-1})\Big)}_{\text{Regularization}}$$

- The reconstruction and regularisation terms are evaluated at every frame t.

Learning: ELBO

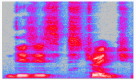
The objective is build in the same way as VAEs, but looks different:

$$\mathcal{L}(\boldsymbol{x}_{1:T}; \boldsymbol{\phi}, \boldsymbol{\theta}) = \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:t}|\boldsymbol{x}_{1:T})} [\underbrace{\ln p_{\boldsymbol{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t})}_{\text{Reconstruction}}]$$

$$- \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:t-1}|\boldsymbol{x}_{1:T})} \left[\underbrace{D_{\text{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:T}) \,\middle\|\, p_{\boldsymbol{\theta}_{\boldsymbol{z}}}(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}) \right)}_{\text{Regularization}} \right]$$

- The reconstruction and regularisation terms are evaluated at every frame t.
- Because of the model, the KL term depends on previous latent variables $\Rightarrow$ sampling.
- The sampling occurs sequentially and cannot be paralelized!

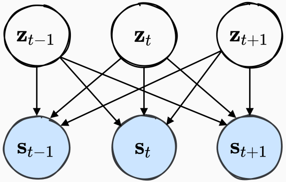
Application to Unsupervised Probabilistic SE (revisit)



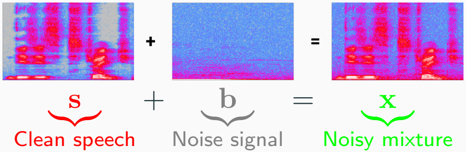
$$\underbrace{\mathbf{s}}_{\text{Clean speech}}$$

Train – Model $\boldsymbol{\theta}$ for clean data: $\{\boldsymbol{s}_{1:T}\}$.

$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_{1:T}) \approx p_{\text{data}}(\boldsymbol{s}_{1:T})$$

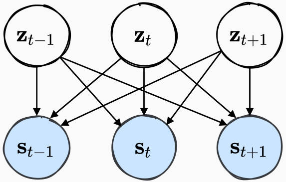


Application to Unsupervised Probabilistic SE (revisit)

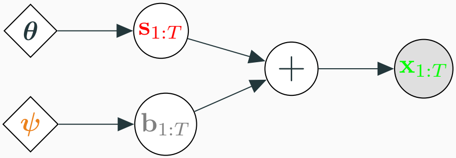


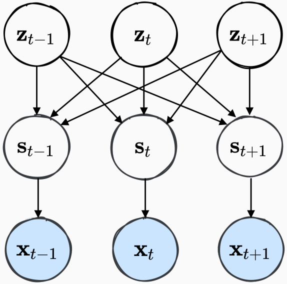
Train – Model $\boldsymbol{\theta}$ for clean data: $\{\boldsymbol{s}_{1:T}\}$.

$$p_{\boldsymbol{\theta}}(\boldsymbol{s}_{1:T}) \approx p_{\text{data}}(\boldsymbol{s}_{1:T})$$



Test – learn the noise parameters from noisy samples $\mathbf{x}$ and estimate the clean speech $\hat{\boldsymbol{s}}_{1:T}$:





Results on the Wall Street Journal (WSJ) and VoiceBank (VB)[11]

Method	Superv.	Test	Train	SI-SDR (dB)	Train	SI-SDR (dB)
Noisy mixture	-		-	−2.6	-	−2.6
VAE-VEM (ICASSP'20)	None	WSJ+QUT	Same	5.0		
RVAE-VEM (Proposed)	None			**5.8**		
MetricGAN-U (ICASSP'22)	Partial			N/A		
UMX* 2020	Full			<u>5.7</u>		
MetricGAN+* (Interspeech'21)	Full			3.6		
Noisy mixture	-					
VAE-VEM (ICASSP'20)	None					
RVAE-VEM (Proposed)	None					
NyTT EUSIPCO'21	Partial/Xtra					
MetricGAN-U (half) ICASSP'22	Partial					
UMX 2020	Full					
MetricGAN+ Interspeech'21	Full					

Results in dB: **best** and <u>second best</u> per section.

[11] Bie, X., *et. al.*, (2022), IEEE TASLP.

Results on the Wall Street Journal (WSJ) and VoiceBank (VB)[11]

Method	Superv.	Test	Train	SI-SDR (dB)	Train	SI-SDR (dB)
Noisy mixture	-		-	−2.6	-	−2.6
VAE-VEM (ICASSP'20)	None	WSJ+QUT	Same	5.0		
RVAE-VEM (Proposed)	None			**5.8**		
MetricGAN-U (ICASSP'22)	Partial			N/A		
UMX* 2020	Full			5.7		
MetricGAN+* (Interspeech'21)	Full			3.6		
Noisy mixture	-		-	8.4	-	8.4
VAE-VEM (ICASSP'20)	None	VB-DMD	Same	16.4		
RVAE-VEM (Proposed)	None			17.1		
NyTT EUSIPCO'21	Partial/Xtra			**17.7**		
MetricGAN-U (half) ICASSP'22	Partial			8.2		
UMX 2020	Full			14.0		
MetricGAN+ Interspeech'21	Full			8.5		

Results in dB: **best** and second best per section.

[11] Bie, X., *et. al.*, (2022), IEEE TASLP.

Results on the Wall Street Journal (WSJ) and VoiceBank (VB)[11]

Method	Superv.	Test	Train	SI-SDR (dB)	Train	SI-SDR (dB)
Noisy mixture	-		-	-2.6	-	-2.6
VAE-VEM (ICASSP'20)	None			5.0		3.8
RVAE-VEM (Proposed)	None			**5.8**		**4.3**
MetricGAN-U (ICASSP'22)	Partial	WSJ+QUT	Same	N/A	Different	-1.6
UMX* 2020	Full			<u>5.7</u>		<u>4.1</u>
MetricGAN+* (Interspeech'21)	Full			3.6		1.8
Noisy mixture	-		-	8.4	-	8.4
VAE-VEM (ICASSP'20)	None			16.4		
RVAE-VEM (Proposed)	None			<u>17.1</u>		
NyTT EUSIPCO'21	Partial/Xtra	VB-DMD	Same	**17.7**		
MetricGAN-U (half) ICASSP'22	Partial			8.2		
UMX 2020	Full			14.0		
MetricGAN+ Interspeech'21	Full			8.5		

Results in dB: **best** and <u>second best</u> per section.

[11] Bie, X., *et. al.*, (2022), IEEE TASLP.

Results on the Wall Street Journal (WSJ) and VoiceBank (VB)[11]

Method	Superv.	Test	Train	SI-SDR (dB)	Train	SI-SDR (dB)
Noisy mixture	-		-	-2.6	-	-2.6
VAE-VEM (ICASSP'20)	None	WSJ+QUT	Same	5.0	Different	3.8
RVAE-VEM (Proposed)	None			**5.8**		**4.3**
MetricGAN-U (ICASSP'22)	Partial			N/A		-1.6
UMX* 2020	Full			<u>5.7</u>		<u>4.1</u>
MetricGAN+* (Interspeech'21)	Full			3.6		1.8
Noisy mixture	-		-	8.4	-	8.4
VAE-VEM (ICASSP'20)	None	VB-DMD	Same	16.4	Different	<u>15.0</u>
RVAE-VEM (Proposed)	None			<u>17.1</u>		**17.3**
NyTT EUSIPCO'21	Partial/Xtra			**17.7**		N/A
MetricGAN-U (half) ICASSP'22	Partial			8.2		N/A
UMX 2020	Full			14.0		10.4
MetricGAN+ Interspeech'21	Full			8.5		3.9

Results in dB: **best** and <u>second best</u> per section.

[11] Bie, X., *et. al.*, (2022), IEEE TASLP.

DVAEs are good (great!) at temporal modeling!

All implementations use RNN (or variants).

- What about transformers?

Xiaoyu Bie

Wen Guo

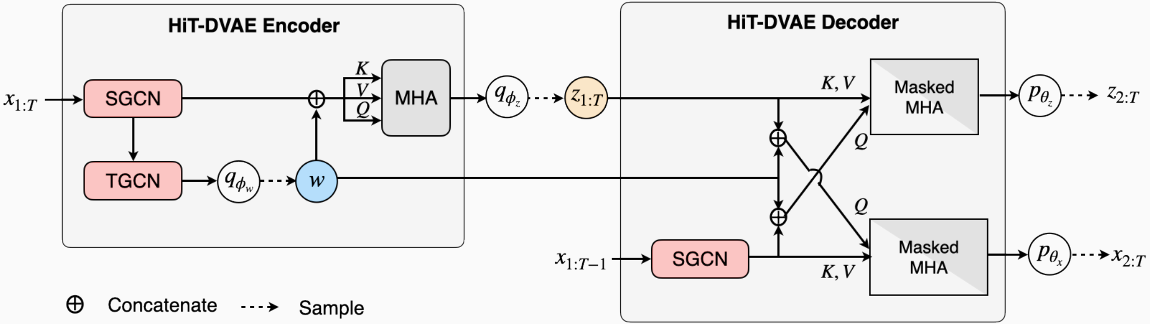
Simon Leglaive

Francesc Moreno

Laurent Girin

Introducing HIT-DVAE[12]

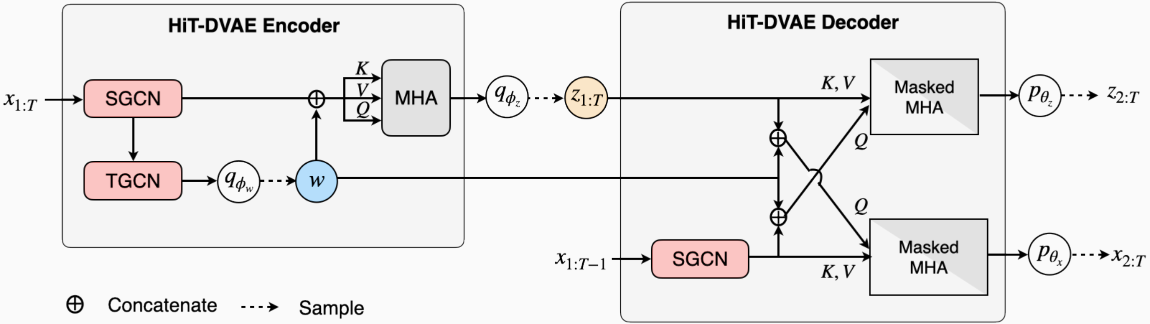
Hierarchical Transformer DVAE – two levels of latent variables: static w and dynamic $z_{1:T}$.



[12] Bie, X., *et. al.*, (2023), Pre-print.

Introducing HIT-DVAE[12]

Hierarchical Transformer DVAE – two levels of latent variables: static w and dynamic $z_{1:T}$.

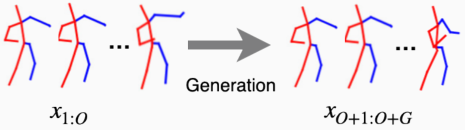


There is something odd in the way Q, K, V are used...

[12] Bie, X., *et. al.*, (2023), Pre-print.

HIT-DVAE: Modified transformer architecture

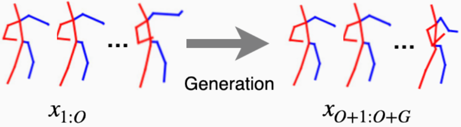
Task: human motion modeling/generation. Predicting $\boldsymbol{x}_{O+1}$ from $\boldsymbol{x}_{1:O}$ is very easy.[13]



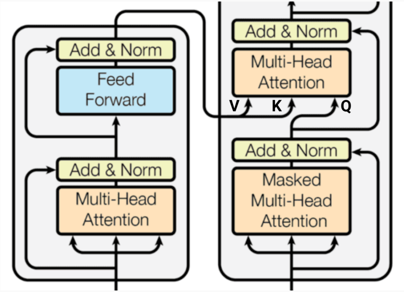
[13] Guo., W., *et. al.*, (2021), IEEE WACV.

HIT-DVAE: Modified transformer architecture

Task: human motion modeling/generation. Predicting x_{O+1} from $x_{1:O}$ is very easy.[13]



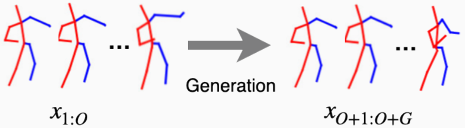
Standard query/residual connection:



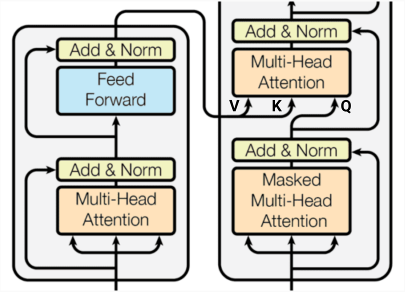
[13] Guo., W., *et. al.*, (2021), IEEE WACV.

HIT-DVAE: Modified transformer architecture

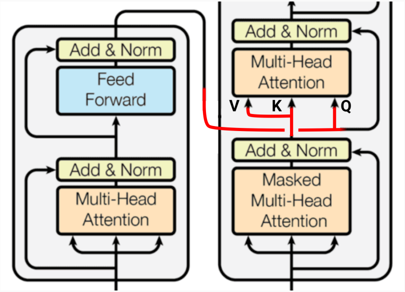
Task: human motion modeling/generation. Predicting x_{O+1} from $x_{1:O}$ is very easy.[13]



Standard query/residual connection:

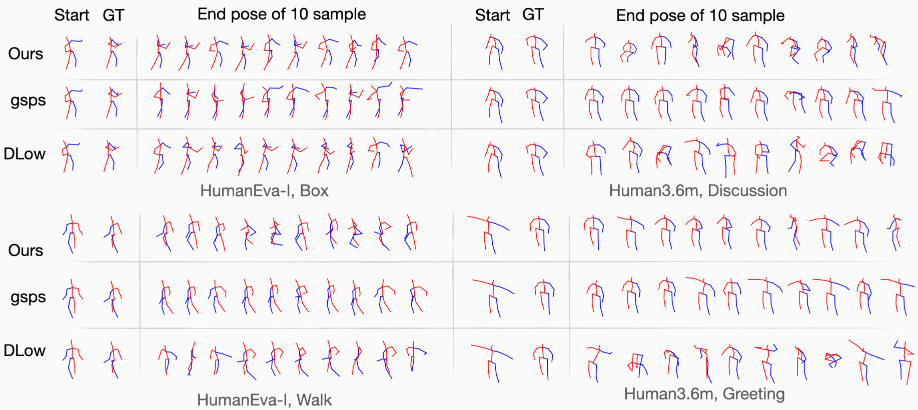


HIT-DVAE query/residual connection:



[13] Guo., W., *et. al.*, (2021), IEEE WACV.

HIT-DVAE: Results



Good trade-off between diversity, quality and accuracy. Also useful for audio modeling.[14]

[14] Lin, X., *et. al.*, (2023), IEEE ICASSP.

DVAEs can model mono-modal sequences.

- **What about multiple modalities?**

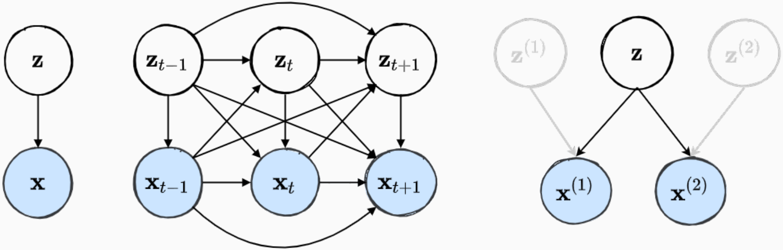
Samir Sadok

Simon Leglaive

Renaud Séguier

Laurent Girin

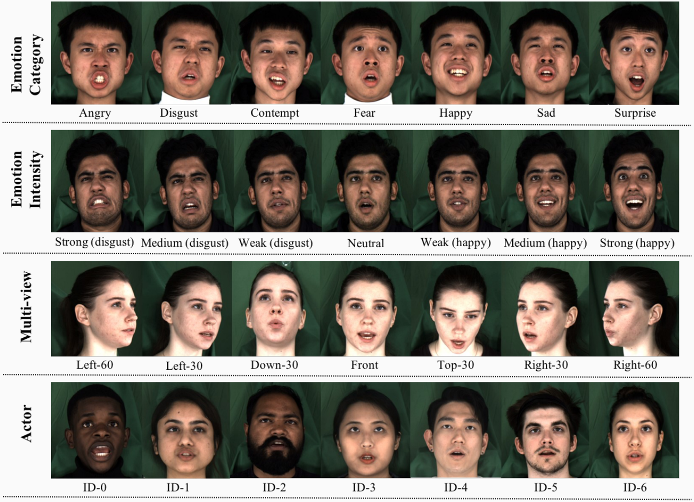
In other words:



VAEs (left) can be multi-modal[15] (right). Can DVAEs (middle) be multi-modal too?

[15] Sutter, T. M., *et. al.*, (2021), ICLR.

Task: emotional audio-visual speech modeling.[16]

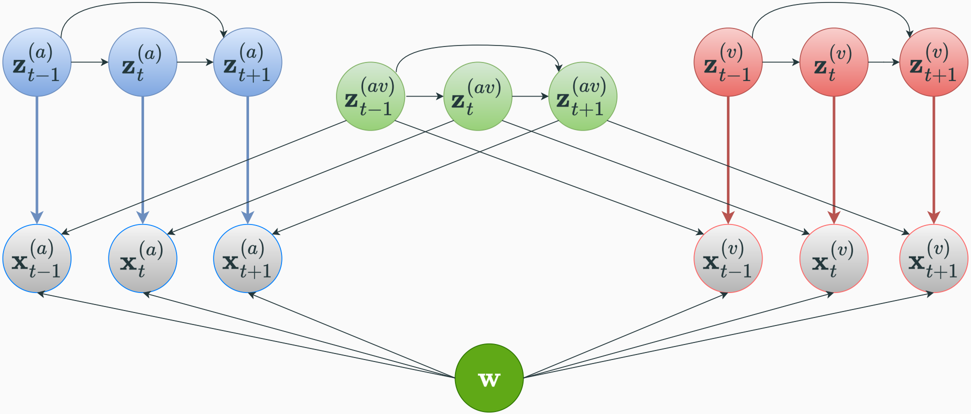


What should we model?

- Static AV (ID, emotion)
- Dynamic AV (lip-audio corr.)
- Dynamic A (other audio features)
- Dynamic V (eye AUs).

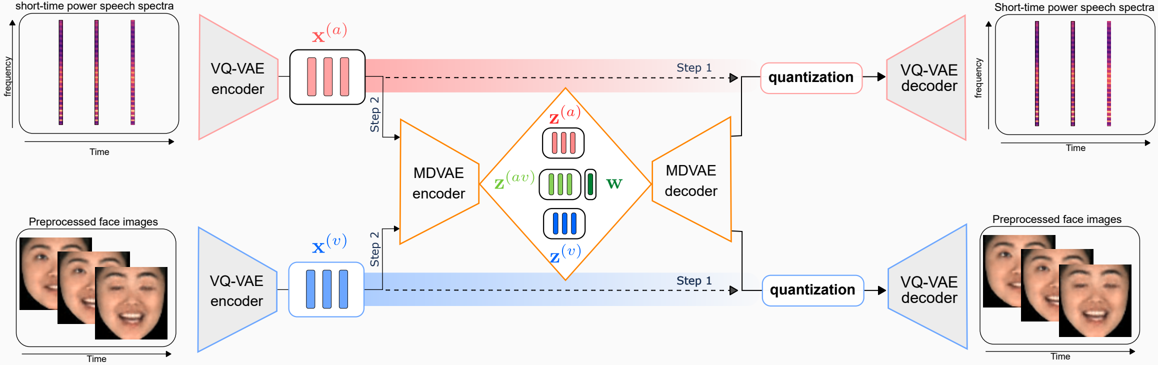
[16]Wang, K., *et. al.*, (2020), ECCV.

Introducing MDVAEs[17]



[17] Sadok, S., *et. al.*, (2023), Under review Neural Networks.

The VQ-MDVAE Architecture



(i) Quantize auditory and visual features. (ii) Use MDVAE to model the quantized features.

MDVAE: Some fun things

Let's see a couple of videos on:

- latent variable transfer,
- latent variable interpolation.

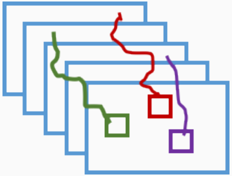
Earlier, we combined (static) VAEs with mixing latents.

- **Is it possible/useful with DVAEs?**

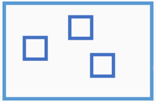
Xiaoyu Lin

Laurent Girin

Motivating application: unsupervised multiple object tracking



Tracklets @ $t-1$

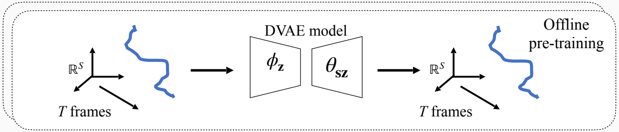


Detections @ t

Desired result

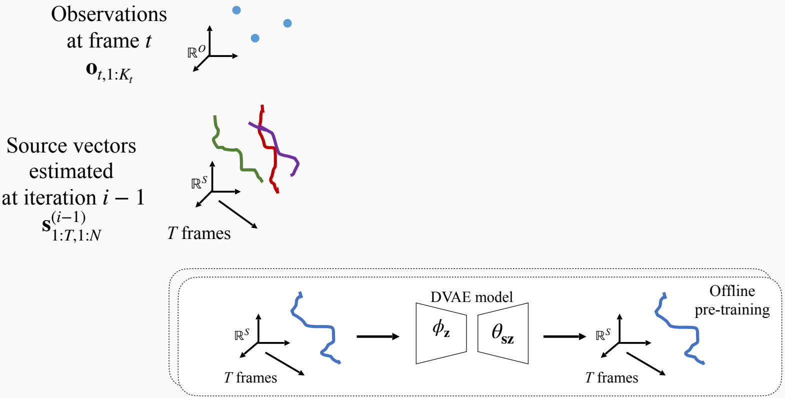
Question: Can we model the non-linear dynamics with a DVAE, and have an assignment mechanism within the same ML formulation?

Introducing MixDVAE[18]



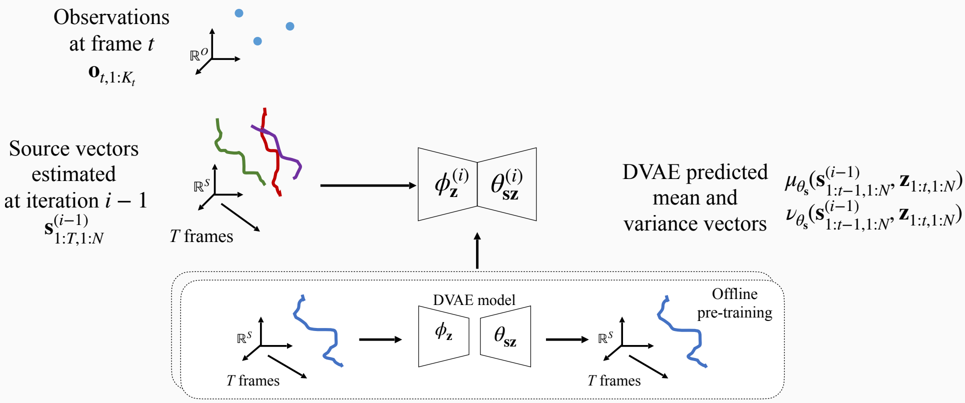
[18] Lin, X., *et. al.*, (2023), Under review.

Introducing MixDVAE[18]



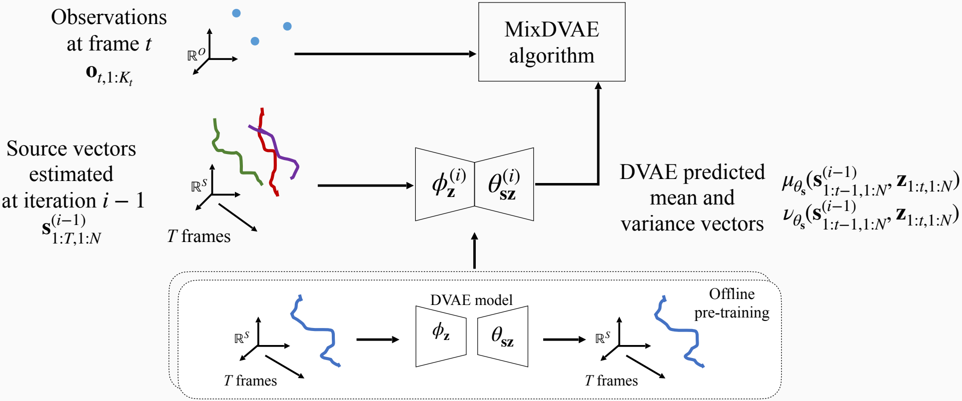
[18] Lin, X., *et. al.*, (2023), Under review.

Introducing MixDVAE[18]



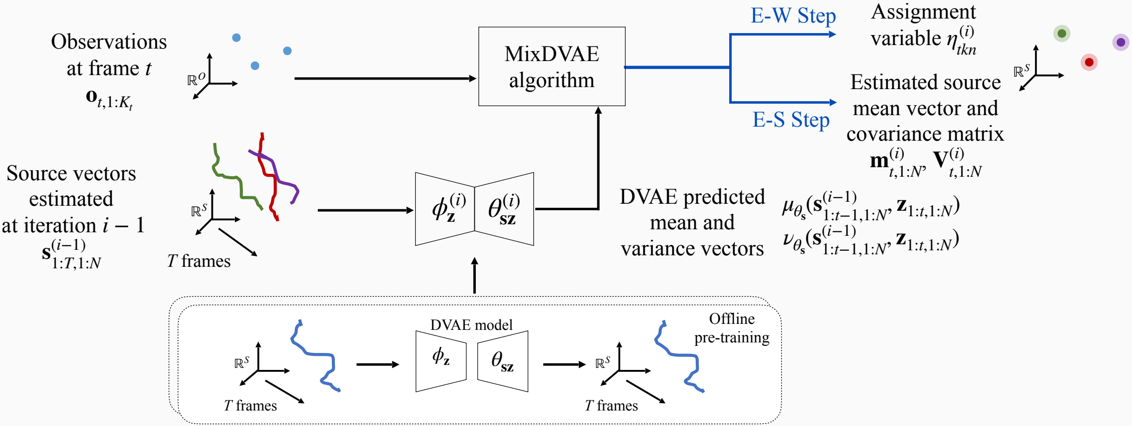
[18] Lin, X., *et. al.*, (2023), Under review.

Introducing MixDVAE[18]



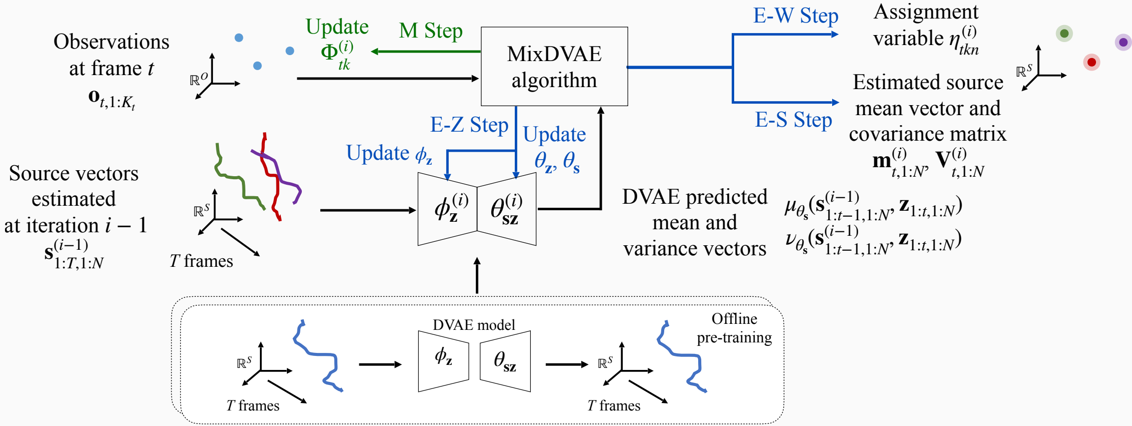
[18] Lin, X., *et. al.*, (2023), Under review.

Introducing MixDVAE[18]



[18] Lin, X., *et. al.*, (2023), Under review.

Introducing MixDVAE[18]



[18] Lin, X., *et. al.*, (2023), Under review.

Quick discussion & results

GMM update (K observations):

$$\boldsymbol{\mu}_n = \sum_k \underbrace{\eta_{kn}}_{\text{assign.}} \boldsymbol{o}_k$$

Quick discussion & results

GMM update (K observations):

$$\boldsymbol{\mu}_n = \sum_k \underbrace{\eta_{kn}}_{\text{assign.}} \boldsymbol{o}_k$$

Kalman update (sequence of T obs):

$$\boldsymbol{s}_t = \underbrace{\boldsymbol{P}_t \boldsymbol{o}_t}_{\text{update}} + \underbrace{\boldsymbol{T}_t \boldsymbol{s}_{t-1}}_{\text{prediction}}$$

Quick discussion & results

GMM update (K observations):

$$\boldsymbol{\mu}_n = \sum_k \underbrace{\eta_{kn}}_{\text{assign.}} \boldsymbol{o}_k$$

Kalman update (sequence of T obs):

$$\boldsymbol{s}_t = \underbrace{\boldsymbol{P}_t \boldsymbol{o}_t}_{\text{update}} + \underbrace{\boldsymbol{T}_t \boldsymbol{s}_{t-1}}_{\text{prediction}}$$

MixDVAE update is a combination:

$$\boldsymbol{s}_{tn} = \underbrace{\boldsymbol{P}_t \sum_k \eta_{kn} \boldsymbol{o}_{tk}}_{\text{assig. \& update}} + \underbrace{\boldsymbol{T}_t(\boldsymbol{s}_{1:t-1})}_{\text{non-lin. prediction}}$$

Quick discussion & results

GMM update (K observations):

$$\boldsymbol{\mu}_n = \sum_k \underbrace{\eta_{kn}}_{\text{assign.}} \boldsymbol{o}_k$$

Kalman update (sequence of T obs):

$$\boldsymbol{s}_t = \underbrace{\boldsymbol{P}_t \boldsymbol{o}_t}_{\text{update}} + \underbrace{\boldsymbol{T}_t \boldsymbol{s}_{t-1}}_{\text{prediction}}$$

MixDVAE update is a combination: $\boldsymbol{s}_{tn} = \underbrace{\boldsymbol{P}_t \sum_k \eta_{kn} \boldsymbol{o}_{tk}}_{\text{assig. \& update}} + \underbrace{\boldsymbol{T}_t(\boldsymbol{s}_{1:t-1})}_{\text{non-lin. prediction}}$

- Results in unsupervised MOT and in semi-blind source separation.
- Work in progress: fine-tuning, complexity, learning from noise, ...

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]		
Dynamic			

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]	CVAE [Sadeghi'20] MVAE [Sutter'21]	
Dynamic			

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]	CVAE [Sadeghi'20] MVAE [Sutter'21]	VAE-MM [Sadeghi'20] MIN-VAE [Sadeghi'21]
Dynamic			

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]	CVAE [Sadeghi'20] MVAE [Sutter'21]	VAE-MM [Sadeghi'20] MIN-VAE [Sadeghi'21]
Dynamic	DVAE [Girin'21] Sw-VAE [Sadeghi'21]		

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]	CVAE [Sadeghi'20] MVAE [Sutter'21]	VAE-MM [Sadeghi'20] MIN-VAE [Sadeghi'21]
Dynamic	DVAE [Girin'21] Sw-VAE [Sadeghi'21]	VQ-MDVAE [Sadok'23]	

Summary

	Mono-modal	Multi-modal	Mixtures
Static	VAE [Kingma'14] VQ-VAE [van den Oord'17]	CVAE [Sadeghi'20] MVAE [Sutter'21]	VAE-MM [Sadeghi'20] MIN-VAE [Sadeghi'21]
Dynamic	DVAE [Girin'21] Sw-VAE [Sadeghi'21]	VQ-MDVAE [Sadok'23]	MixDVAE [Lin'23]

Conclusions & Future Work

- Variational inference provides a general framework for multi-modal unsupervised learning.
- It is specially suitable for low-data regimes.
- Recent models (VAE, DVAE, VQ-VAE) can be combined with more classical ones (mixture models, HMM) within the same probabilistic paradigm.
- The probabilistic/maximum likelihood paradigm provides a principled way for disentangling representations and interpreting them.

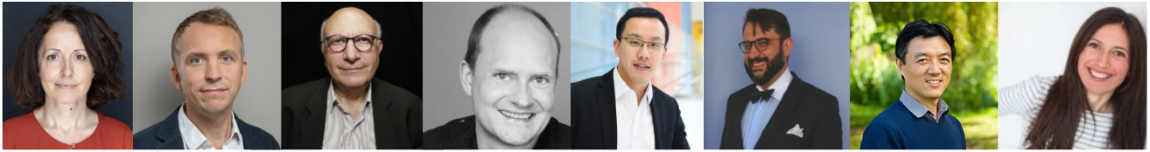
Open questions:

- Computational complexity is an issue when involving EM/VEM algorithms.
- These methods MUST be extended/rethought for other modalities.
- Understand and develop opportunities with other families (Diffusion, Flows, ...).

Social Robotics, Artificial Intelligence and Multimedia

Grenoble (FR) 19th-23rd, February, 2024. **No fee!**

Prof. Hatice Gunes, University of Cambridge
Prof. Gabriel Skantze, KTH Stockholm
Prof. Raja Chatila, Sorbonne Université
Prof. Marc Hanheide, University of Lincoln
Prof. Xuesu Xiao, George Mason University
Prof. Antonios Gasteratos, Democritus U. of Thrace
Prof. Wenwu Wang, University of Surrey
Dr. Vasiliki Charisi, JRC European Comission



Thanks...

for bearing with me!

to my colleagues & collaborators!

for your challenging questions
& interesting discussion.

We are also interested in meta-learning, reinforcement learning, domain adaptation, etc.