

Abstract

We address the problem of learning informative latent representations in the context of variational autoencoders. To do this, we

- use a hierarchical prior to avoid the over-regularisation resulting from a standard normal prior distribution.
- formulate the learning problem as a constrained optimisation problem.
- introduce a graph-based interpolation method to evaluate the learned latent representation.

Variational Autoencoders as a Constrained Optimisation Problem

Rezende and Viola (2018) reformulate the VAE objective as the Lagrangian

$$\mathcal{L}(\theta, \phi; \lambda) \equiv \mathbb{E}_{p_D(\mathbf{x})} \left[\underbrace{\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_0(\mathbf{z}))}_{\text{optimisation objective}} + \lambda \underbrace{(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\mathcal{C}_\theta(\mathbf{x}, \mathbf{z})] - \kappa^2)}_{\text{inequality constraint}} \right]$$

of a constrained optimisation problem

$$\underbrace{\min_{\theta} \max_{\lambda} \min_{\phi}}_{\text{E-step}} \mathcal{L}(\theta, \phi; \lambda) \quad \text{s.t.} \quad \lambda \geq 0.$$

Here, $\mathcal{C}_\theta(\mathbf{x}, \mathbf{z})$ is defined as the reconstruction-error-related term in $-\log p_\theta(\mathbf{x}|\mathbf{z})$. Thus, $\min_\theta \mathcal{L}$ and $\max_\lambda \min_\phi \mathcal{L}$ can be interpreted as M- and E-step, respectively, of the original EM algorithm for training VAEs. Optimisation is performed by a quasi-gradient ascent/descent algorithm (GECO):

$$\lambda_t = \lambda_{t-1} \cdot \exp(\nu \cdot (\mathcal{C}_t - \kappa^2)) \quad \text{and} \quad (\theta_t, \phi_t) = (\theta_{t-1}, \phi_{t-1}) - \eta_t \partial_{(\theta, \phi)} \mathcal{L},$$

where $\Delta \lambda_t \cdot \partial_\lambda \mathcal{L} \geq 0$ and ν is the update's learning rate. We optimise the ELBO iff $\lambda = 1$; or if $0 \leq \lambda < 1$, a lower bound on the ELBO (see β -VAE formulation (Higgins et al., 2017)).

Hierarchical Priors for Learning Informative Latent Representations

The optimal empirical Bayes prior is the aggregated posterior distribution $p^*(\mathbf{z}) = \mathbb{E}_{p_D(\mathbf{x})} [q_\phi(\mathbf{z}|\mathbf{x})]$. In order to express it, we use a hierarchical prior/two-layer stochastic model

$$p_0(\mathbf{z}) \equiv p_\Theta(\mathbf{z}) = \int p_\Theta(\mathbf{z}|\zeta) p(\zeta) d\zeta$$

and learn the parameters by applying an importance-weighted lower bound on $\mathbb{E}_{p^*(\mathbf{z})} [\log p_\Theta(\mathbf{z})] \geq \mathbb{E}_{p_D(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\mathcal{L}_{\text{IW}}(\Theta, \Phi; \mathbf{z})]$ (Burda et al., 2016):

$$\begin{aligned} \mathbb{E}_{p_D(\mathbf{x})} [\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\Theta(\mathbf{z}))] &\leq \mathbb{E}_{p_D(\mathbf{x})} [\mathcal{F}(\phi, \Theta, \Phi; \mathbf{x})] \\ &\equiv \mathbb{E}_{p_D(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\underbrace{\log q_\phi(\mathbf{z}|\mathbf{x}) - \mathbb{E}_{\zeta_{1:K} \sim q_\Theta(\zeta|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^K p_\Theta(\mathbf{z}, \zeta_k) \right]}_{\mathcal{L}_{\text{IW}}(\Theta, \Phi; \mathbf{z})} \right]. \end{aligned}$$

As a result, we arrive at the Lagrangian objective

$$\mathcal{L}_{\text{VHP}}(\theta, \phi, \Theta, \Phi; \lambda) \equiv \mathbb{E}_{p_D(\mathbf{x})} \left[\mathcal{F}(\phi, \Theta, \Phi; \mathbf{x}) + \lambda (\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\mathcal{C}_\theta(\mathbf{x}, \mathbf{z})] - \kappa^2) \right],$$

where the constrained optimisation problem is formulated as

$$\underbrace{\min_{\Theta, \Phi}}_{\text{empirical Bayes}} \underbrace{\max_{\lambda}}_{\text{E-step}} \underbrace{\min_{\phi}}_{\text{E-step}} \mathcal{L}_{\text{VHP}}(\theta, \phi, \Theta, \Phi; \lambda) \quad \text{s.t.} \quad \lambda \geq 0.$$

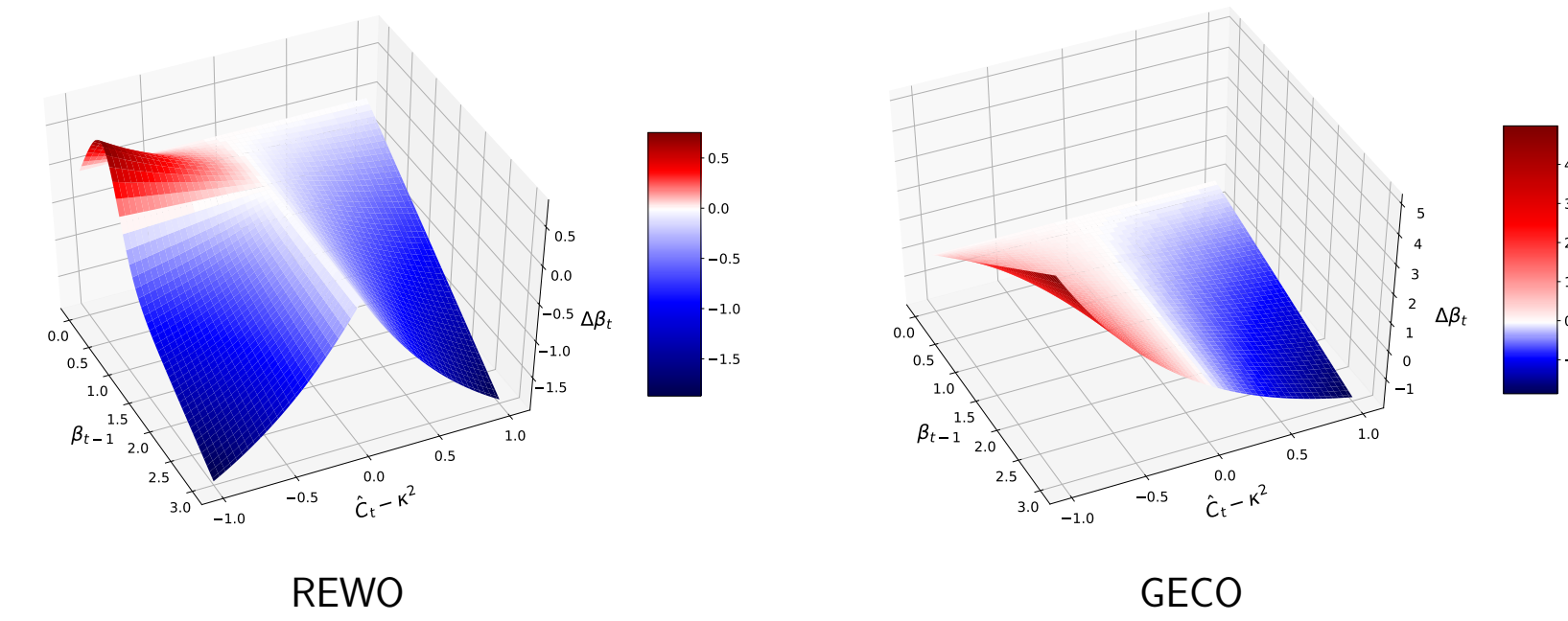
This leads to the following double-loop method: (i) update the upper bound (empirical Bayes) via (Θ, Ψ) ; (ii) solve the constrained optimisation problem w.r.t. (θ, λ, ψ) .

Optimisation: to be in line with previous literature and to facilitate the comparison with the original VAE framework, we use the β -parametrisation: $\beta = \frac{1}{\lambda}$.

Our goal is to obtain a tight lower bound on the log-likelihood. This holds when $\beta = 1$ (ELBO). To guarantee that the optimisation process finishes at $\beta = 1$ —provided the constraint is fulfilled—we propose the following update:

$$\beta_t = \beta_{t-1} \cdot \exp[\nu \cdot f_\beta(\beta_{t-1}, \mathcal{C}_t - \kappa^2; \tau) \cdot (\mathcal{C}_t - \kappa^2)],$$

where $f_\beta(\beta, \delta; \tau) \equiv (1 - H(\delta)) \cdot \tanh(\tau \cdot (\beta - 1)) - H(\delta)$; H is the Heaviside function; and τ is a slope parameter.



Comparison of β -update schemes: $\Delta \beta_t = \beta_t - \beta_{t-1}$ as a function of β_{t-1} and $\mathcal{C}_t - \kappa^2$ for $\nu = 1$ and $\tau = 3$.

We experienced that the double-loop method behaves as a layer-wise pre-training. Thus, we propose an algorithm (REWO) that separates training into two phases:

- Initial phase: to enforce a reconstruction optimisation, we start with $\beta \ll 1$ and optimise \mathcal{L}_{VHP} w.r.t. (θ, ϕ) , until $\mathcal{C}_t < \kappa^2$.
- Main phase: after $\mathcal{C}_t < \kappa^2$ is fulfilled, we optimise \mathcal{L}_{VHP} w.r.t. $(\Theta, \Phi, \theta, \phi)$ and update β .

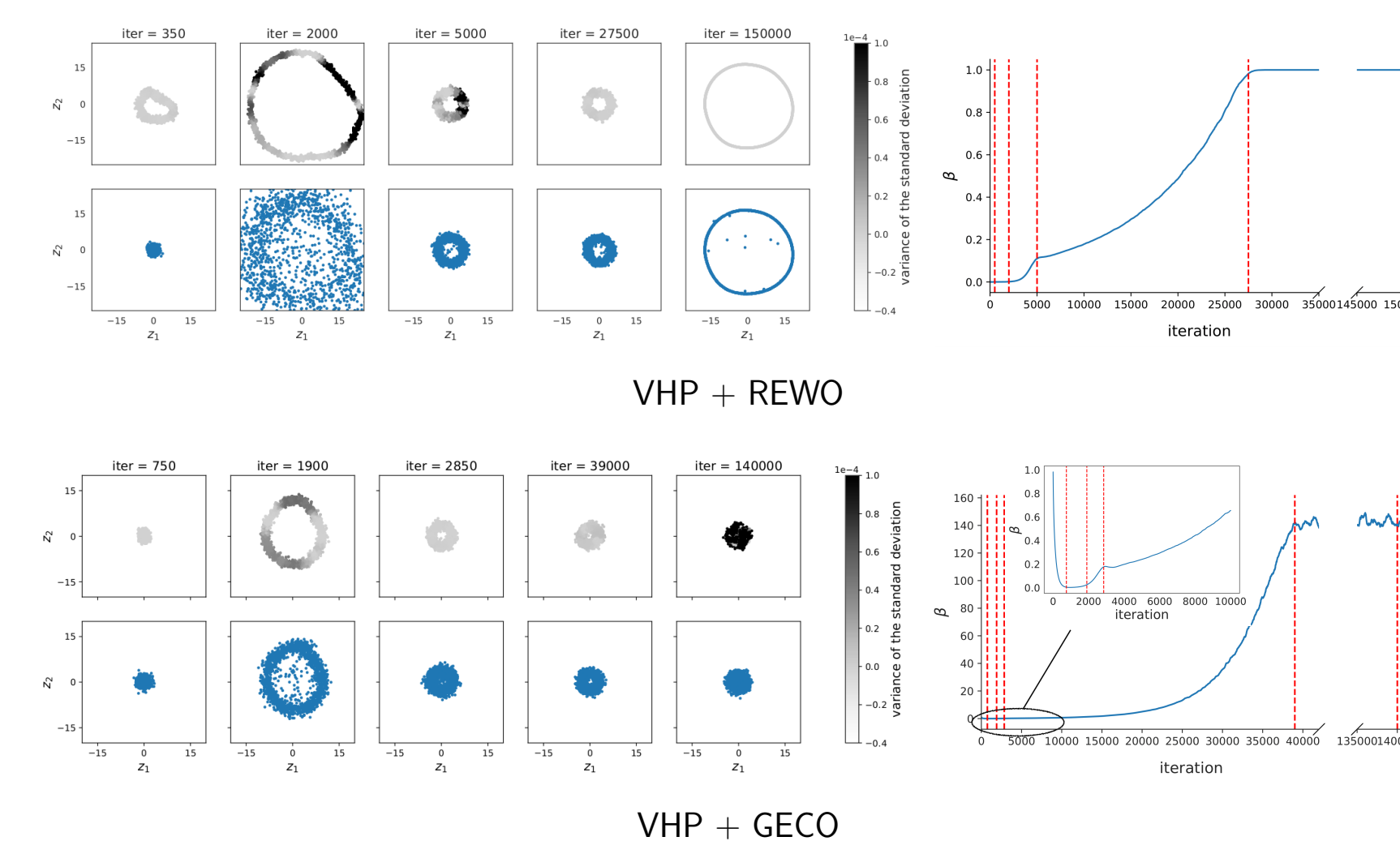
Graph-Based Interpolation

The nodes of the graph $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ are obtained by randomly sampling N samples from the prior distribution:

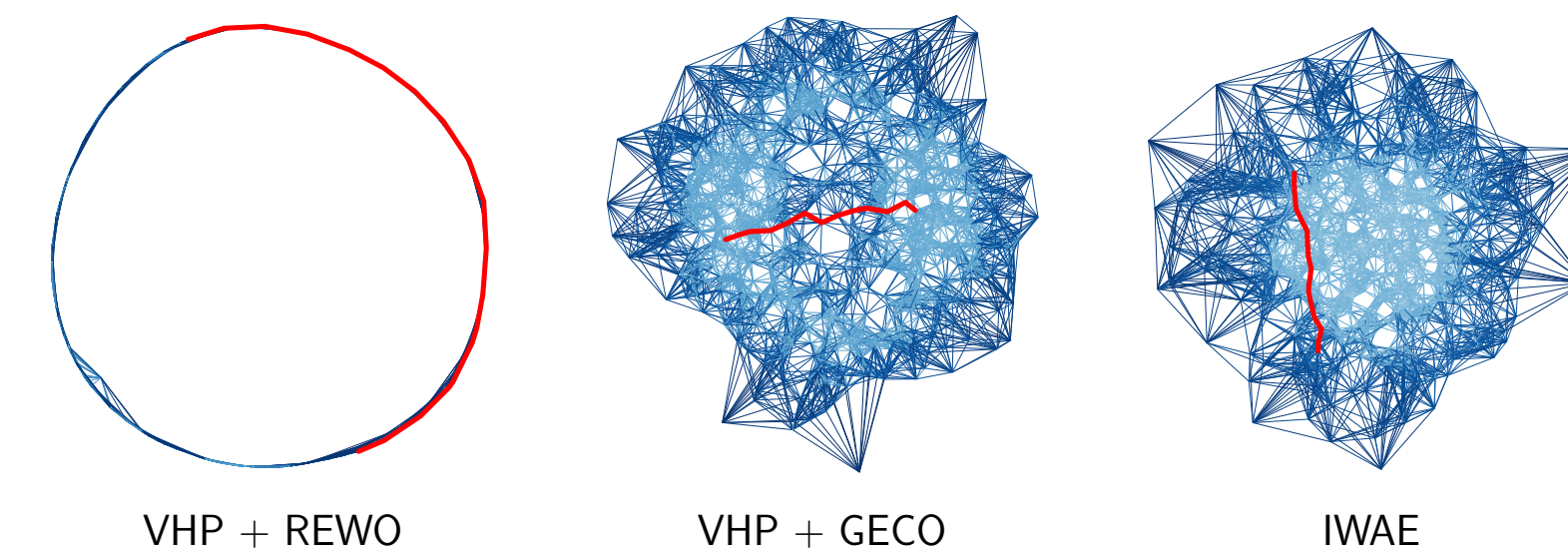
$$\mathbf{z}_n, \zeta_n \sim p_\Theta(\mathbf{z}|\zeta) p(\zeta), \quad n = 1, \dots, N.$$

The graph is constructed by connecting each node by undirected edges to its k-nearest neighbours. The edge weights are the Euclidean distances (latent space) between the node pairs.

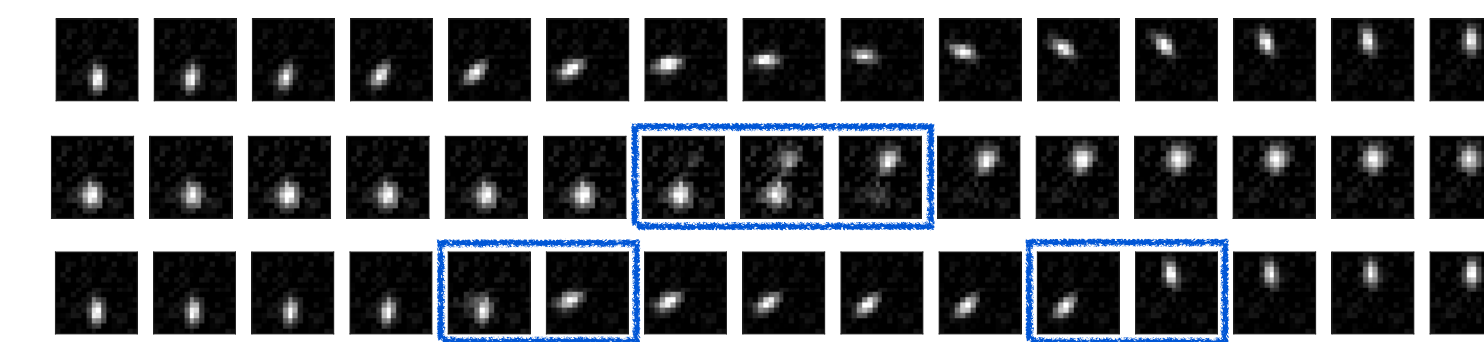
Artificial Pendulum



Latent representation of the pendulum data at different iteration steps when optimising $\mathcal{L}_{\text{VHP}}(\theta, \phi, \Theta, \Phi; \beta)$ with REWO and GECO, respectively. The top row shows the approximate posterior; the greyscale encodes the variance of its standard deviation. The bottom row shows the hierarchical prior.



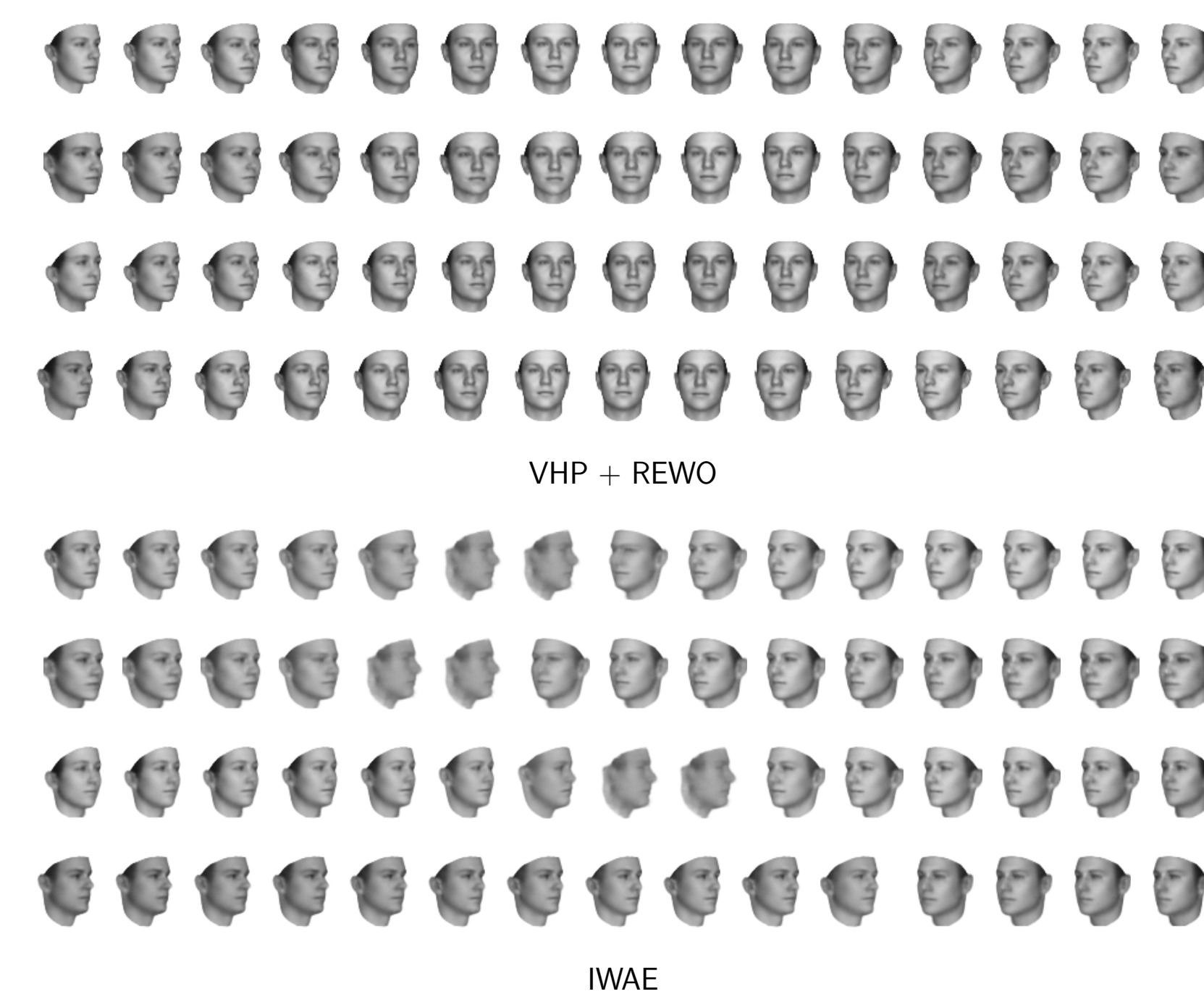
Graph-based interpolation of the pendulum movement. The graph is based on the respective prior. The red curves depict the interpolations, the bluescale indicates the edge weight.



top: VHP + REWO, middle: VHP + GECO, bottom: IWAE

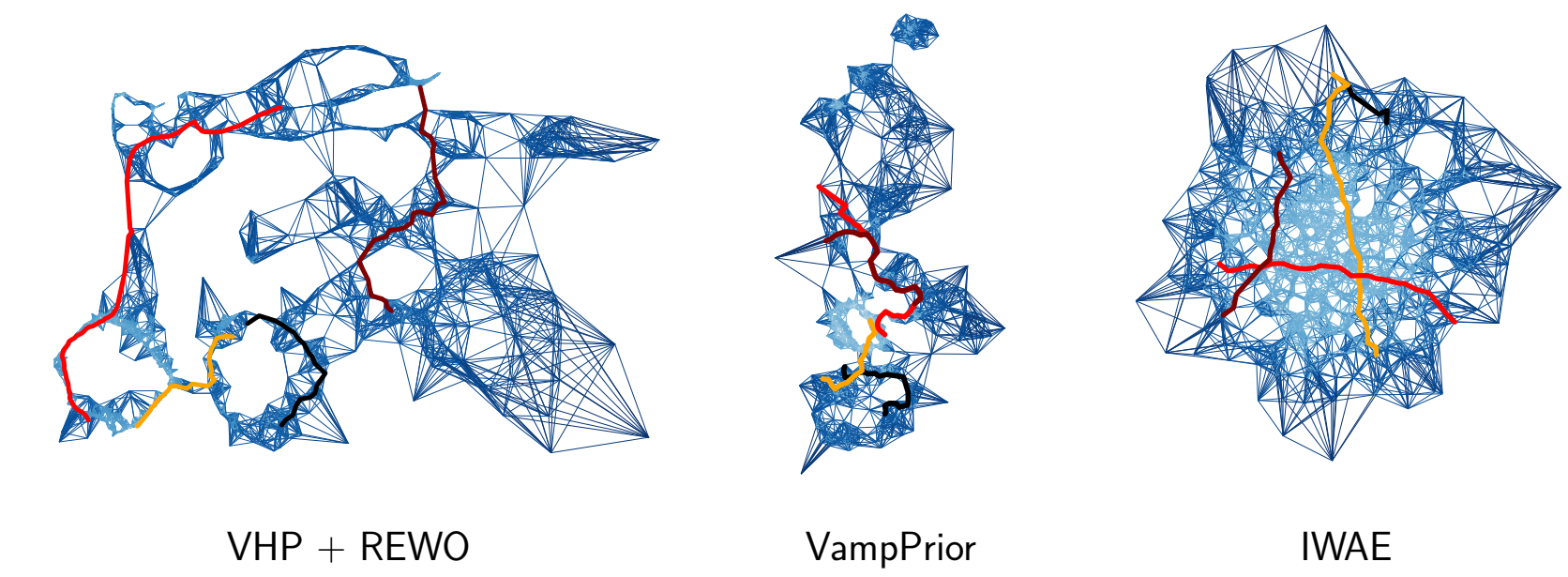
Pendulum reconstructions of the graph-based interpolations (red curve). Discontinuities are marked by blue boxes.

3D Faces

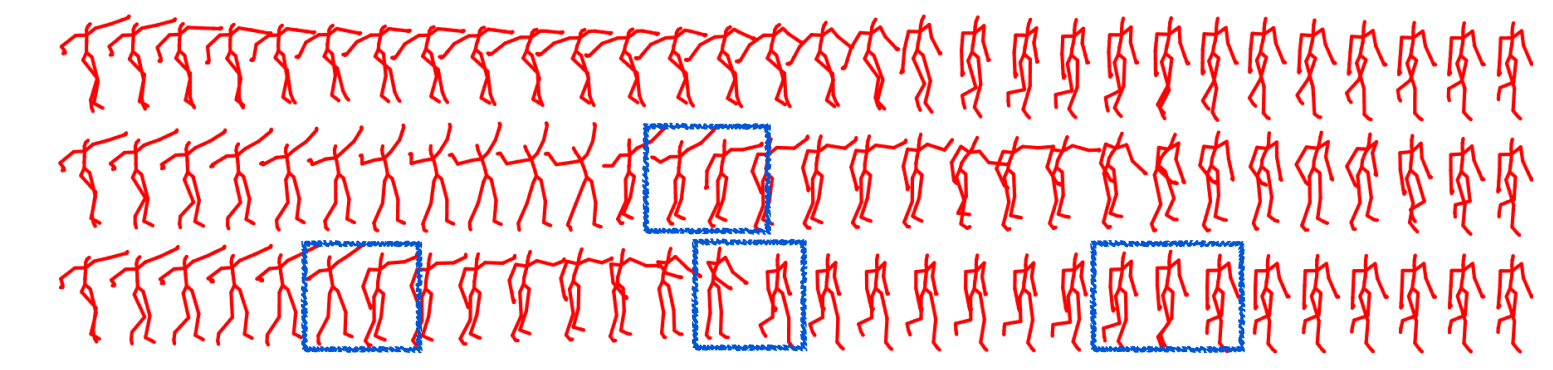


Graph-based interpolations along the learned 32-dimensional latent manifold. The graph is based on samples from the respective prior distribution.

Human Motion

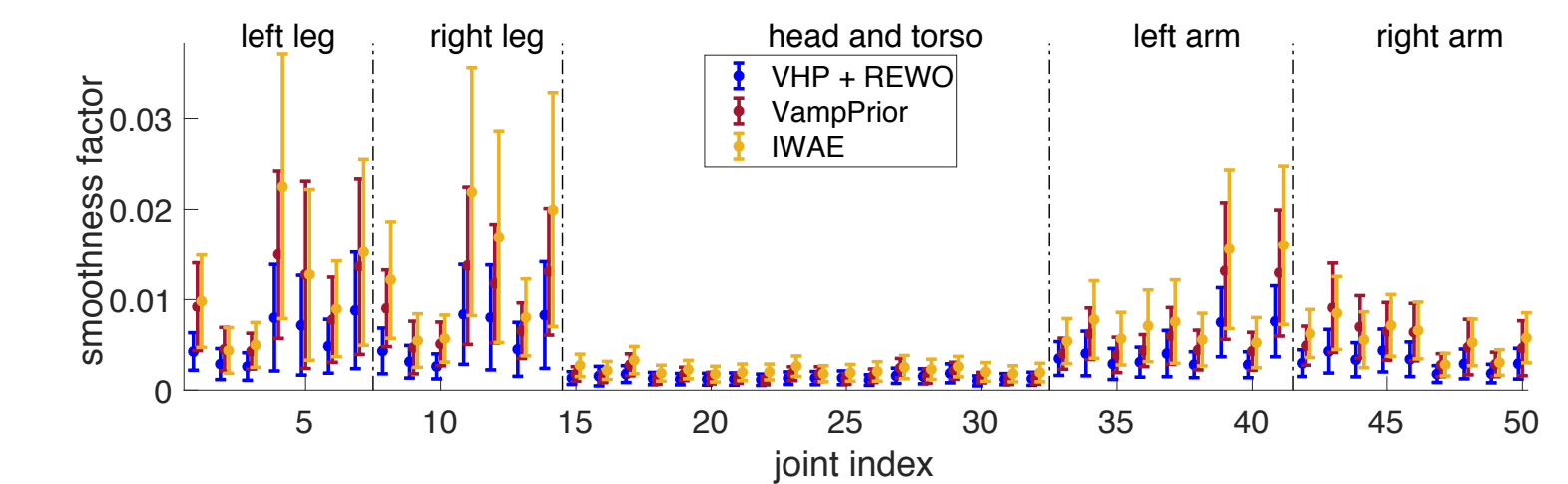


Graph-based interpolation of human motions. The graphs are based on the (learned) prior distributions. The bluescale indicates the edge weight. The coloured lines represent four interpolated movements.



top: VHP + REWO, middle: VampPrior, bottom: IWAE

Human-movement reconstructions of the graph-based interpolations (red curve). Discontinuities are marked by blue boxes.



Smoothness measure of the human-movement interpolations. For each joint, the mean and standard deviation of the smoothness factor are displayed. Smaller values correspond to smoother movements.

MNIST, Fashion-MNIST, & OMNIGLOT

Negative test log-likelihood estimated with 5,000 importance samples

	DYNAMIC MNIST	STATIC MNIST	FASHION- MNIST	OMNIGLOT
VHP + REWO	78.88	82.74	225.37	101.78
VHP + GECO	95.01	96.32	234.73	108.97
VAMPprior	80.42	84.02	232.78	101.97
IWAE (L=1)	81.36	84.46	226.83	101.57
IWAE (L=2)	80.66	82.83	225.39	101.83

References

- Burda, Y. et al. (2016). Importance weighted autoencoders. *ICLR*.
Higgins, I. et al. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*.
Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *ICML*.
Rezende, D. J. and Viola, F. (2018). Taming VAEs. *CoRR*.
Tomczak, J. and Welling, M. (2018). VAE with a VampPrior. *AISTATS*.

Contact

alexej.klushyn@argmax.ai
botond.cseke@argmax.ai

