[07/01/22 GTS Research Seminar]

Missing Data Imputation and Acquisition with Deep Hierarchical Models and Hamiltonian Monte Carlo

Ignacio Peis[1], Chao Ma[2,3], José Miguel Hernández-Lobato[2]

[1]Dept. of Signal Theory and Communications, Universidad Carlos III de Madrid
[2]Dept. of Engineering, University of Cambridge
[3]Microsoft Research Cambridge

Introduction

Challenges

- Improve approximate inference in advanced VAEs
- Improve missing data imputation
- Improve predictions under missing data condition
- Improve active information acquisition
- Deal with partial, mixed-type data

Introduction

Contributions

- Improved Hierarchical VAE for mixed-type partial data.
- Improved inference via Hamiltonian Monte Carlo with automatic hyperparameter optimization.
- Improved active learning with novel sampling-based active information acquisition technique.

Missing Data Imputation and Acquisition with Deep Hierarchical Models and Hamiltonian Monte Carlo

Ignacio Peis[1] **Chao Ma**[2,3] **José Miguel Hernández-Lobato**[2]
[1]Universidad Carlos III de Madrid [2]University of Cambridge [3]Microsoft Research Cambridge
ipeis@tsc.uc3m.es
{cm905,jmh233}@cam.ac.uk

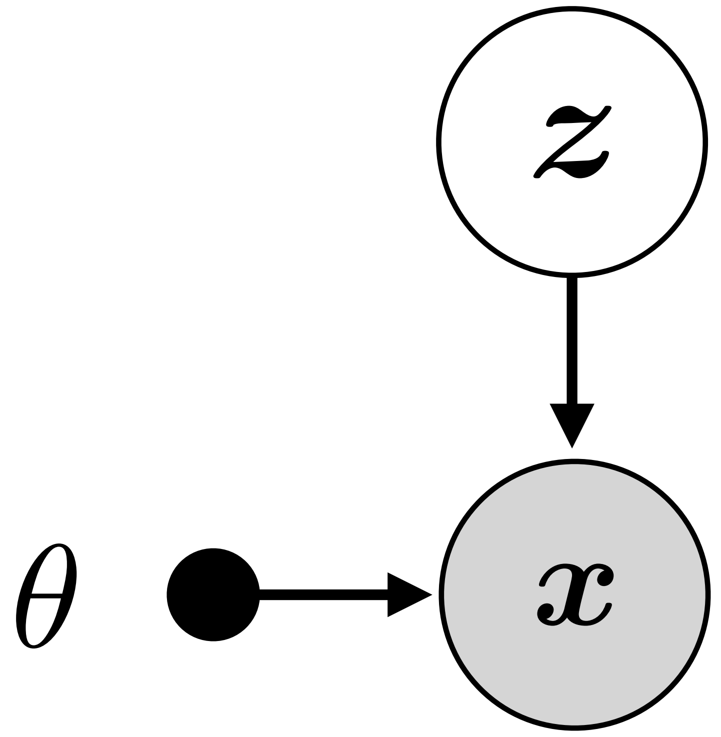
[Preprint] [Code]

Variational Autoencoders

Definition[1]

- Generative, explicit models with intractable probability.
- **Goal**: optimize parameters to maximize the marginal likelihood:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$



- **Intractable**, due to the complexity added by NNs.
- Posterior distribution:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z})}{p_\theta(\boldsymbol{x})}$$

[1] Kigma et at., 2013

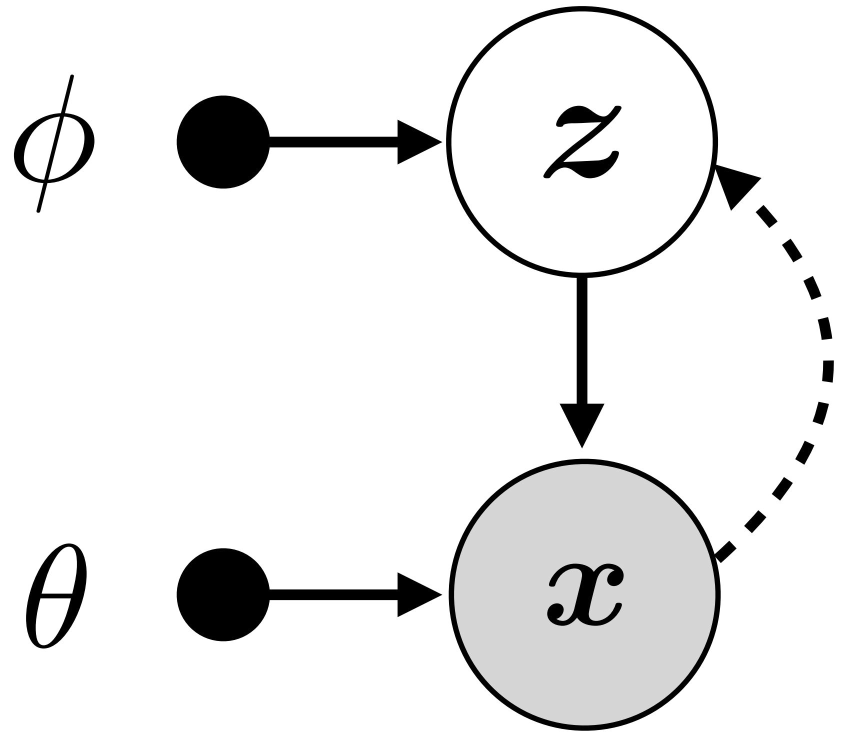
Variational Autoencoders

Amortized Variational Inference

- Optimize a Gaussian approximation of the posterior

$$D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})\right)$$

- Minimizing this KL is equivalent to maximizing the Evidence Lower Bound



$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})},$$

- Alternatively:

Encoder

$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right)$$

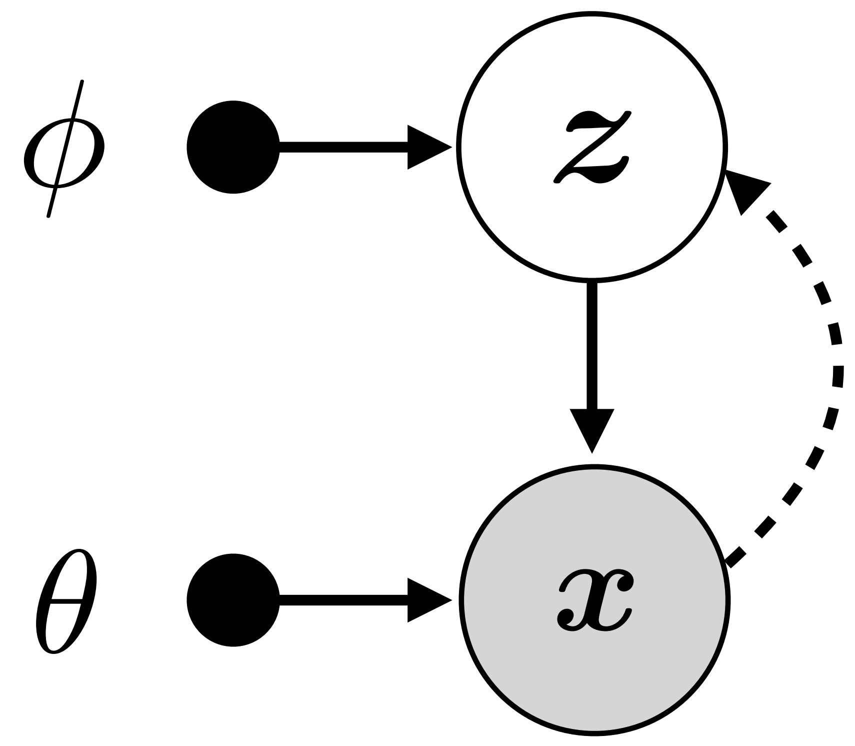
Decoder

$$\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Variational Autoencoders

Training

- For each batch:
    1. Encode to parameters of the approximate posterior.
    2. Sample from $q_\phi(\boldsymbol{z}|\boldsymbol{x})$.
    3. Obtain the Monte Carlo approximation of the ELBO.
    4. Optimization step on θ and ϕ.

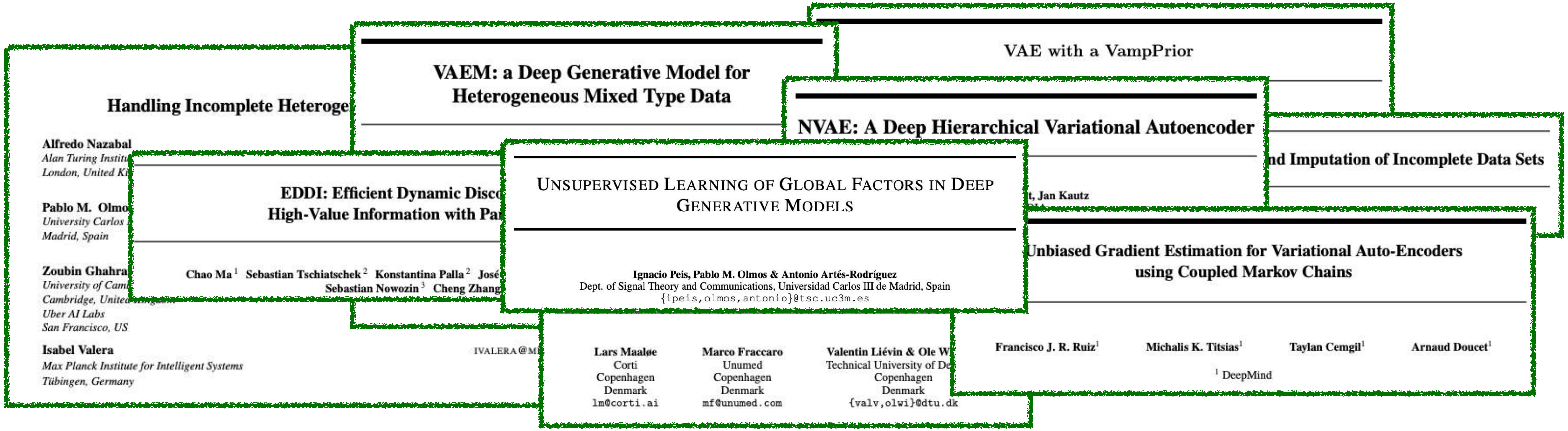


$$\nabla_{(\theta,\phi)}\left(\frac{1}{B}\sum_{i=1}^{B}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right)\right]\right)$$

Variational Autoencoders

Recent advances

- Handling incomplete and heterogeneous data.
- Improving approximate inference VS increasing flexibility of the prior.



Hierarchical Variational Autoencoders

Definition*

- Add flexibility to the prior with an autoregressive path of latent variables

$$ELBO(\mathbf{x}) = \mathbb{E}_{Q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z}_1) - KL[q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1|\mathbf{z}_2)] - KL[q(\mathbf{z}_2|\mathbf{z}_1)||p(\mathbf{z}_2)]\right]$$

$$q(\mathbf{z}_2|\mathbf{z}_1) \approx p(\mathbf{z}_2) \approx \mathcal{N}(0, 1)$$

- **Inductive bias**: hierarchical flow of information.
- **Potential pitfalls:** *posterior collapse.*

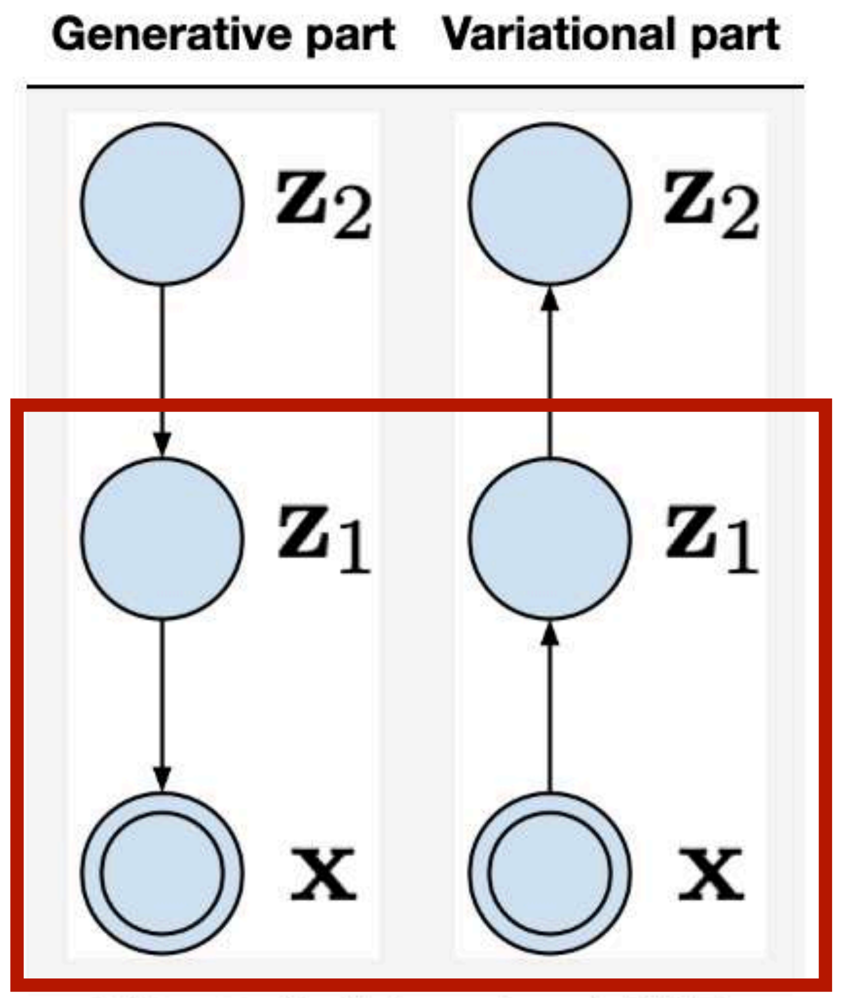


Figure 4. A two-level VAE.

* https://jmtomczak.github.io/blog/9/9_hierarchical_lvm_p1.html

Hierarchical Variational Autoencoders

Definition

- To avoid posterior collapse, the variational networks learn a residual difference of the posterior from a deterministic bottom-up path[1,2,3].

$$q(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mu_i + \Delta\mu_i(\mathbf{x}), \sigma_i^2 \Delta\sigma_i^2)$$

- The generative and variational posterior are tightly connected:

$$KL\left(q\left(z^i \mid x\right) \| p\left(z^i\right)\right) = \frac{1}{2}\left(\frac{\Delta\mu_i^2}{\sigma_i^2} + \Delta\sigma_i^2 - \log\Delta\sigma_i^2 - 1\right)$$

Figure 1: 256×256-pixel samples generated by NVAE, trained on CelebA HQ [28].

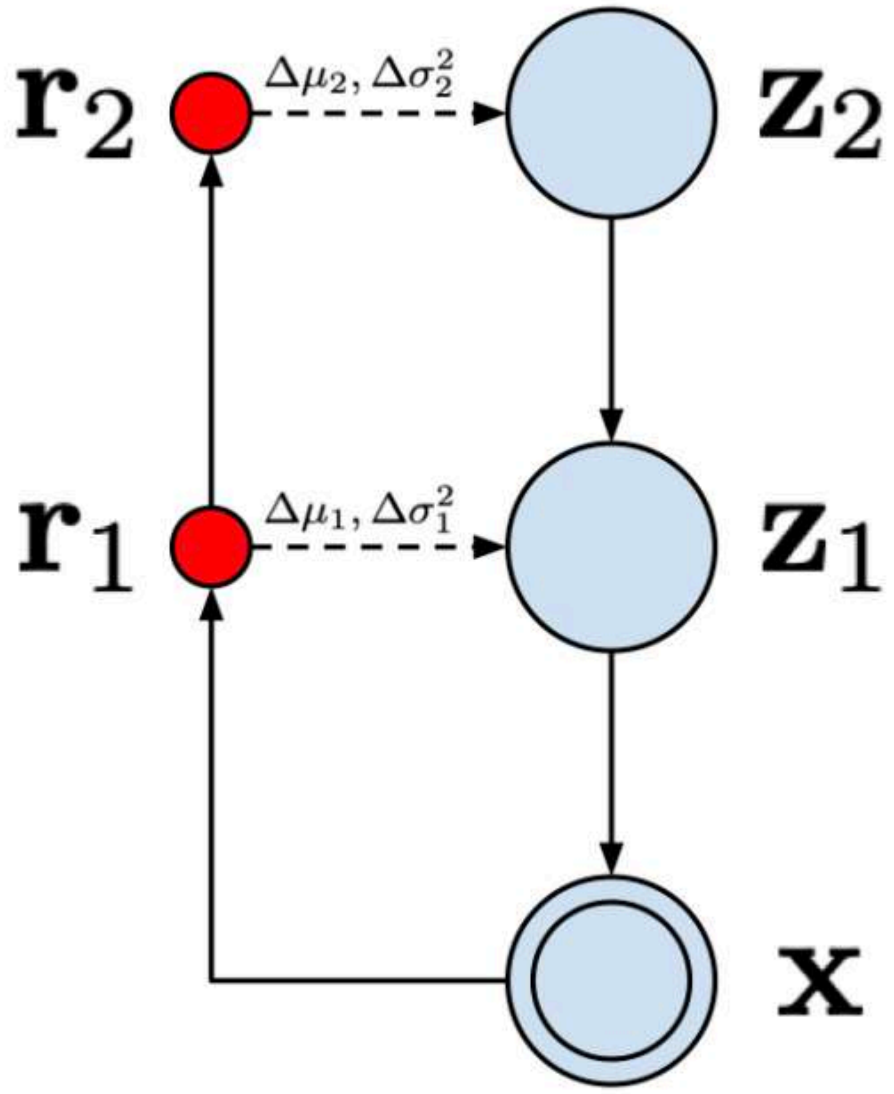
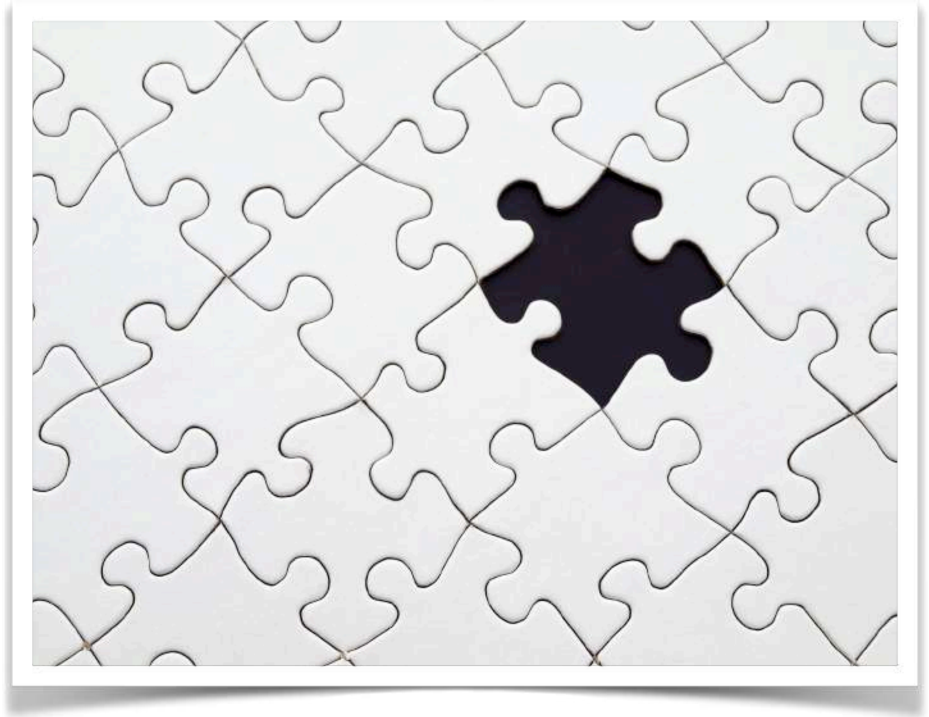


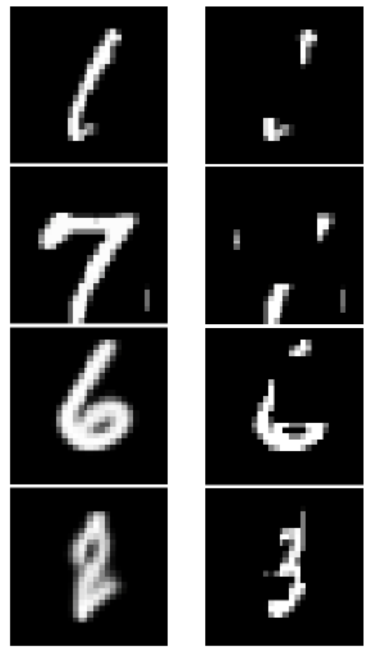
Figure 5. A top-down VAE.

[1] Vahdat et at., 2020 [2] Maaløe et at., 2019 [3] Child, 2019

Variational Autoencoders

VAEs for partial, heterogeneous data



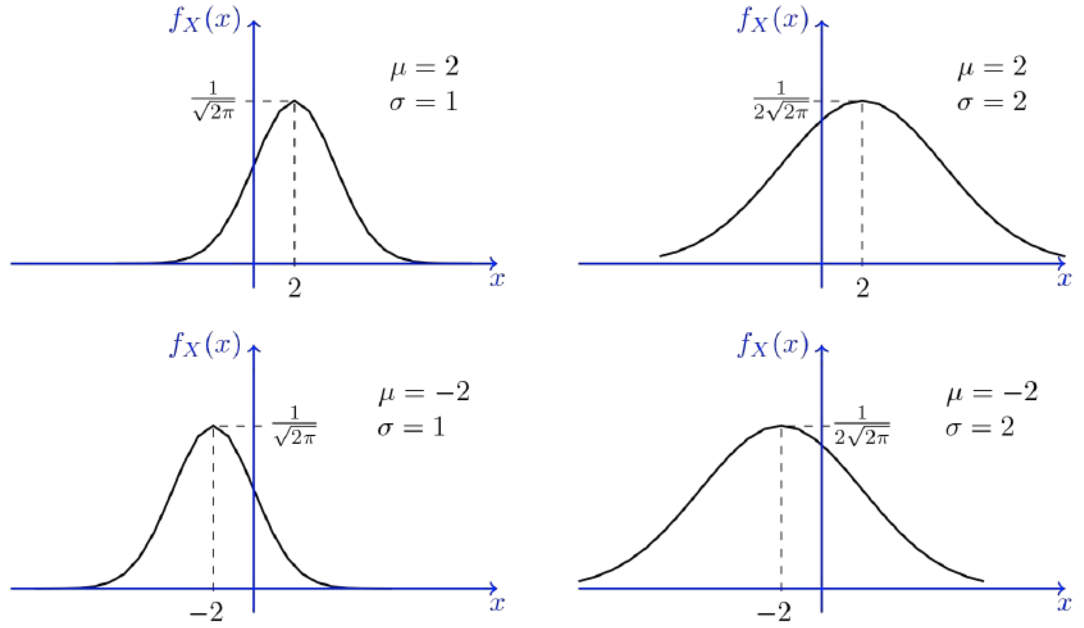


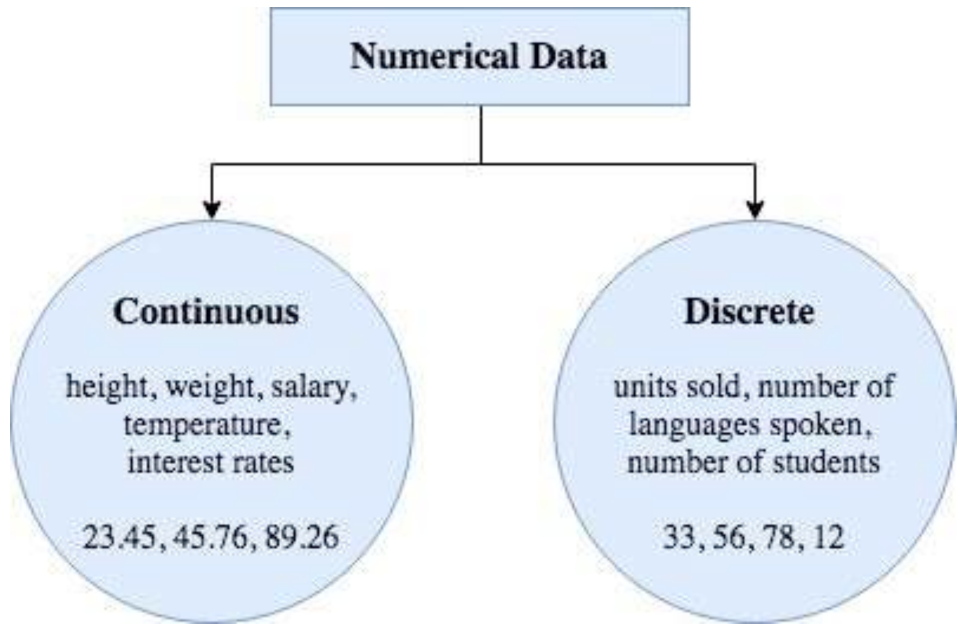
- Factorization over dimensions[1,2]:

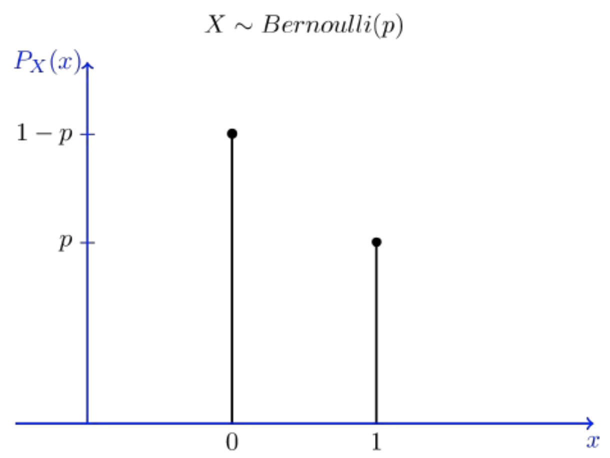
$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{d=1}^{D} \mathbb{I}(x_d \in \boldsymbol{x}_O) \log p_\theta(x_d|\boldsymbol{z})\right] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}_O)||p(\boldsymbol{z}))$$

- Naïve approach for heterogeneous[1]: use a different likelihood per dimension.

- **Problem**: unbalanced likelihoods.







[1] Nazabal et at., 2020 [2] Mattei et at., 2019

Variational Autoencoders

VAEM for partial heterogeneous data

- **Solution**[1]: learn first D marginal VAEs (θ_d, γ_d):

$$\mathcal{L}_d(x_d; \{\theta_d, \gamma_d\}) = \mathbb{I}(x_d \in \boldsymbol{x}_o)\mathbb{E}_{q_{\gamma_d}(z_d|x_d)} \log \frac{p_{\theta_d}(x_d, z_d)}{q_{\gamma_d}(z_d|x_d)}$$

And and a joint dependency VAE (θ, ϕ) on the marginally encoded data:

$$\mathcal{L}(\boldsymbol{z}) = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{z})}\left[\sum_{d=1}^{D} \mathbb{I}(z_d \in \boldsymbol{z}_O)\mathbb{E}_{q_{\gamma_d}(z_d|x_d)}\left[\log p_\theta(z_d|\boldsymbol{h})\right]\right] - D_{KL}(q_\phi(\boldsymbol{h}|\boldsymbol{z}_O)||p(\boldsymbol{h}))$$

- The marginal encodings z_d are Gaussian, thus, the likelihoods are balanced.
- Interdependencies between heterogenous variables are better captured by the model.

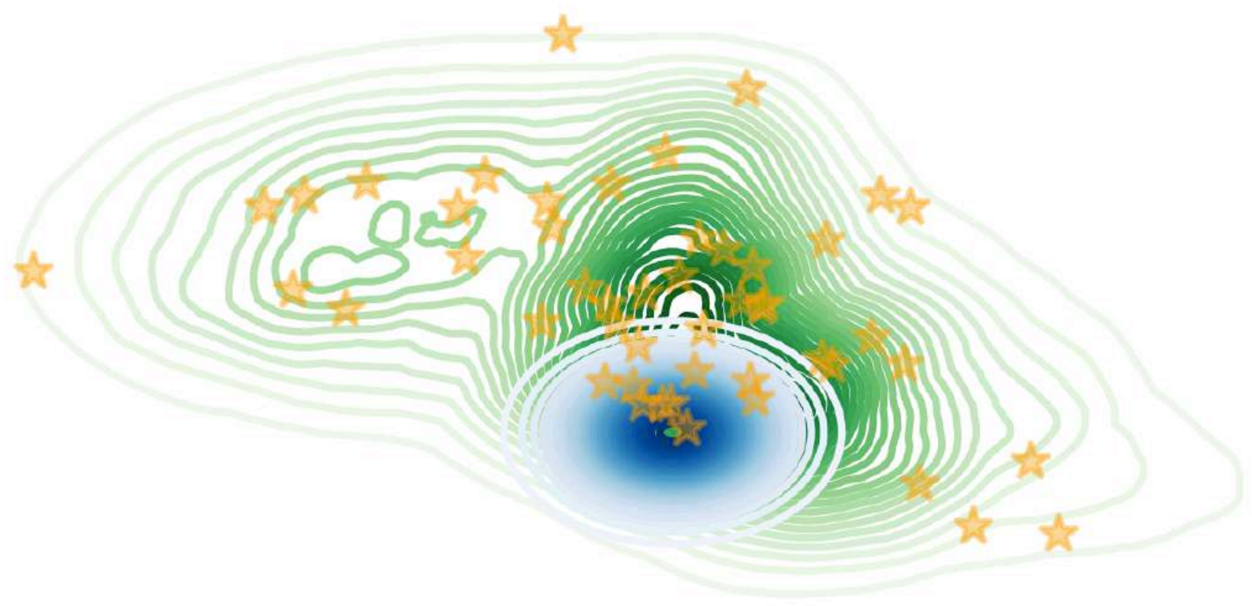
[1] Ma et at., 2020

Approximate Inference in VAEs

Variational Inference

- VAEs are **restricted** by purely Gaussian approximations of the true posterior.

[1]True posterior (green) vs Approximate posterior (blue)



[2]Samples from the true posterior (orange)

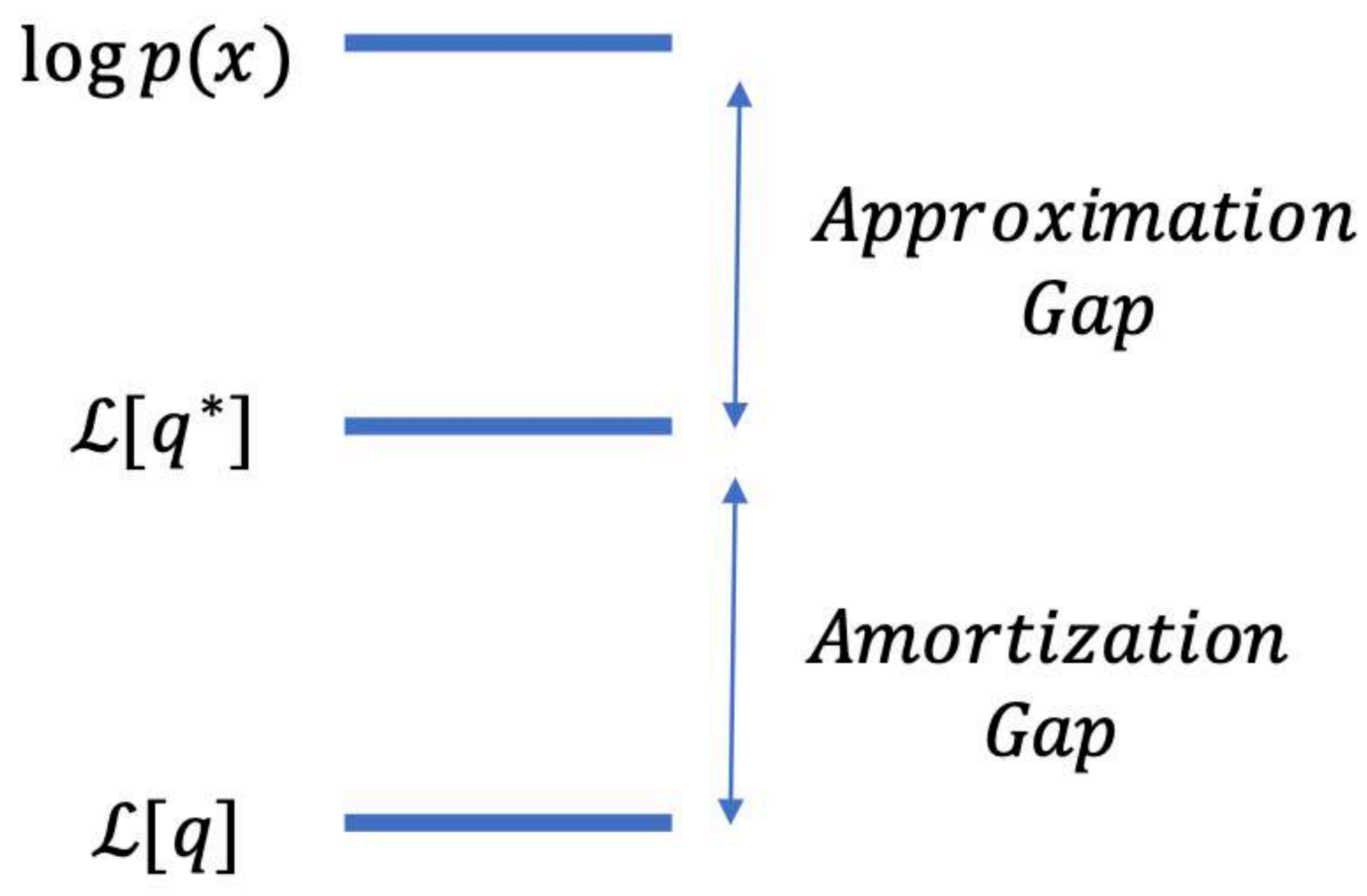


Figure 1. Gaps in Inference

[1] Cremer et at., 2018 [2] Peis et at., 2022

Bayesian active information acquisition

Encoder-based

- Reward function as an expected gain of information[1]:

$$R(i, \mathbf{x}_O) = \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_O)} D_{\mathrm{KL}}\left[p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_O) \,\|\, p(\mathbf{x}_\phi|\mathbf{x}_O)\right]$$

- **Intractable**. Solved by transforming into $\mathbf{z}$ space[2]:

$$\begin{aligned}\hat{R}(i, \mathbf{x}_o) = &\mathbb{E}_{\mathbf{x}_i \sim \hat{p}(\mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[q(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_o)\right] - \\ &\mathbb{E}_{\mathbf{x}_\phi, \mathbf{x}_i \sim \hat{p}(\mathbf{x}_\phi, \mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[q(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o)\right]\end{aligned}$$

- Extension for the VAEM model[3]:

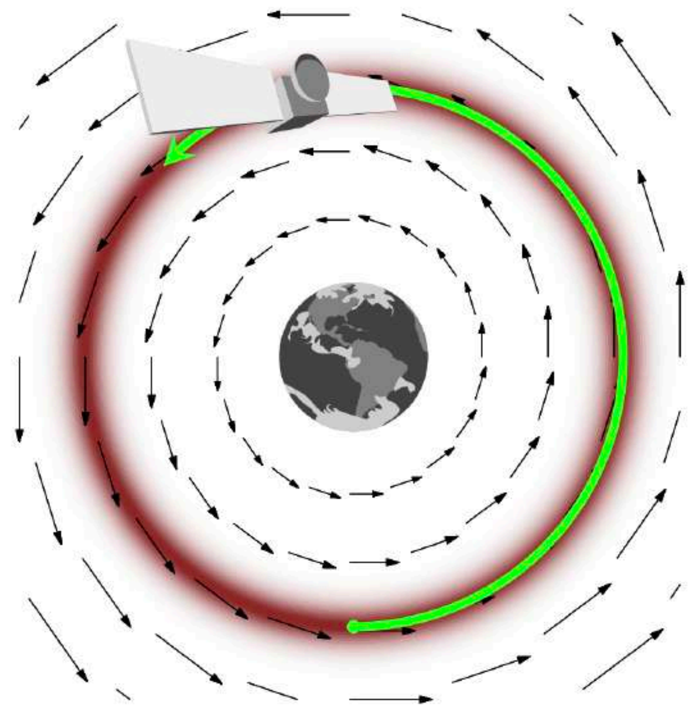
$$\begin{aligned}\hat{R}_I(\mathbf{x}_i, \mathbf{x}_O) = &\mathbb{E}_{p_\theta(\mathbf{x}_i, \mathbf{z}_i, \mathbf{z}_O|\mathbf{x}_O)} \{\mathbb{KL}\left[q_\lambda(\mathbf{h}|\mathbf{z}_i, \mathbf{z}_O)||q_\lambda(\mathbf{h}|\mathbf{z}_O)\right] - \\ &\mathbb{E}_{p_\theta(\mathbf{x}_\phi, \mathbf{z}_\Phi, |\mathbf{x}_O)} \mathbb{KL}\left[q_\lambda(\mathbf{h}|\mathbf{z}_\Phi, \mathbf{z}_i, \mathbf{z}_O)||q_\lambda(\mathbf{h}|\mathbf{z}_\Phi, \mathbf{z}_O)\right]\}\end{aligned}$$

- **Restriction**: this approximation metric still relies on the Gaussian approximation of the encoder.

[1] Bernardo et at., 1979 [2] Ma et at., 2018 [2] Ma et at., 2020

Hamiltonian Monte Carlo[1,2]

Definition

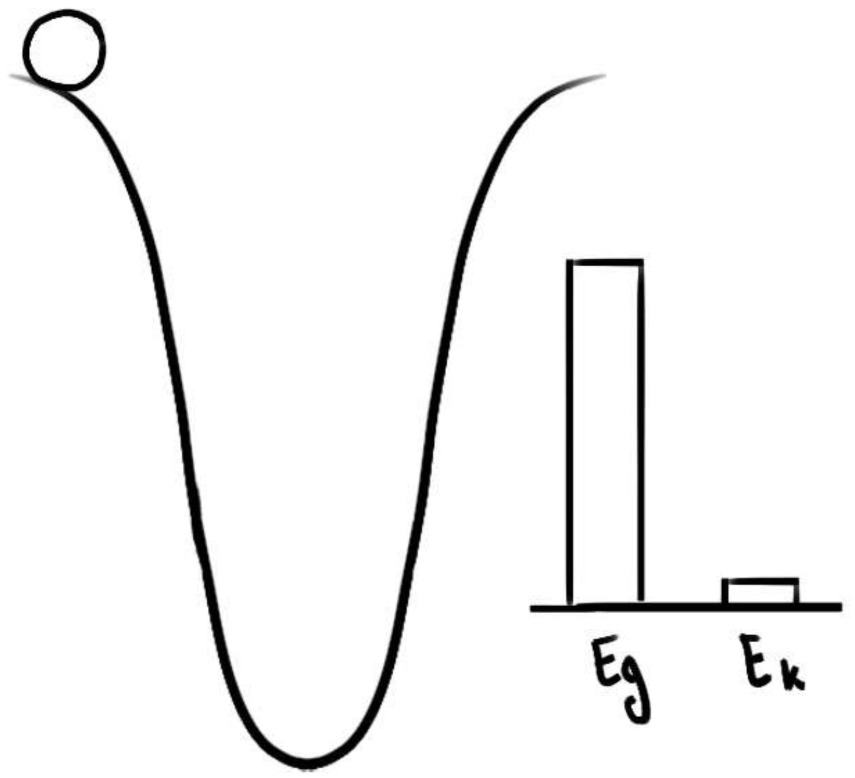


- Sample from complex distributions via unnormalised targets $p(z) = \frac{1}{Z}\boxed{p^*(z)}$
- Highly efficient exploration using differential geometry and conservative dynamics
    1. Expand to phase space with momentum variable:

$$(\boldsymbol{z}) \rightarrow (\boldsymbol{z}, \boldsymbol{r}) \qquad\qquad p(\boldsymbol{z}, \boldsymbol{r}) = p(\boldsymbol{r}|\boldsymbol{z})p(\boldsymbol{z})$$

$$H(\boldsymbol{z}, \boldsymbol{r}) = -\log p(\boldsymbol{z}, \boldsymbol{r}) = -\log p(\boldsymbol{r}|\boldsymbol{z}) - \log p(\boldsymbol{z}) = K(\boldsymbol{r}, \boldsymbol{z}) + V(\boldsymbol{z})$$

$$H(\boldsymbol{z}, \boldsymbol{r}) = -\log p^*(\boldsymbol{z}) + \frac{1}{2}\boldsymbol{r}^T \boldsymbol{M}^{-1}\boldsymbol{r}.$$



    2. Hamiltonian equations

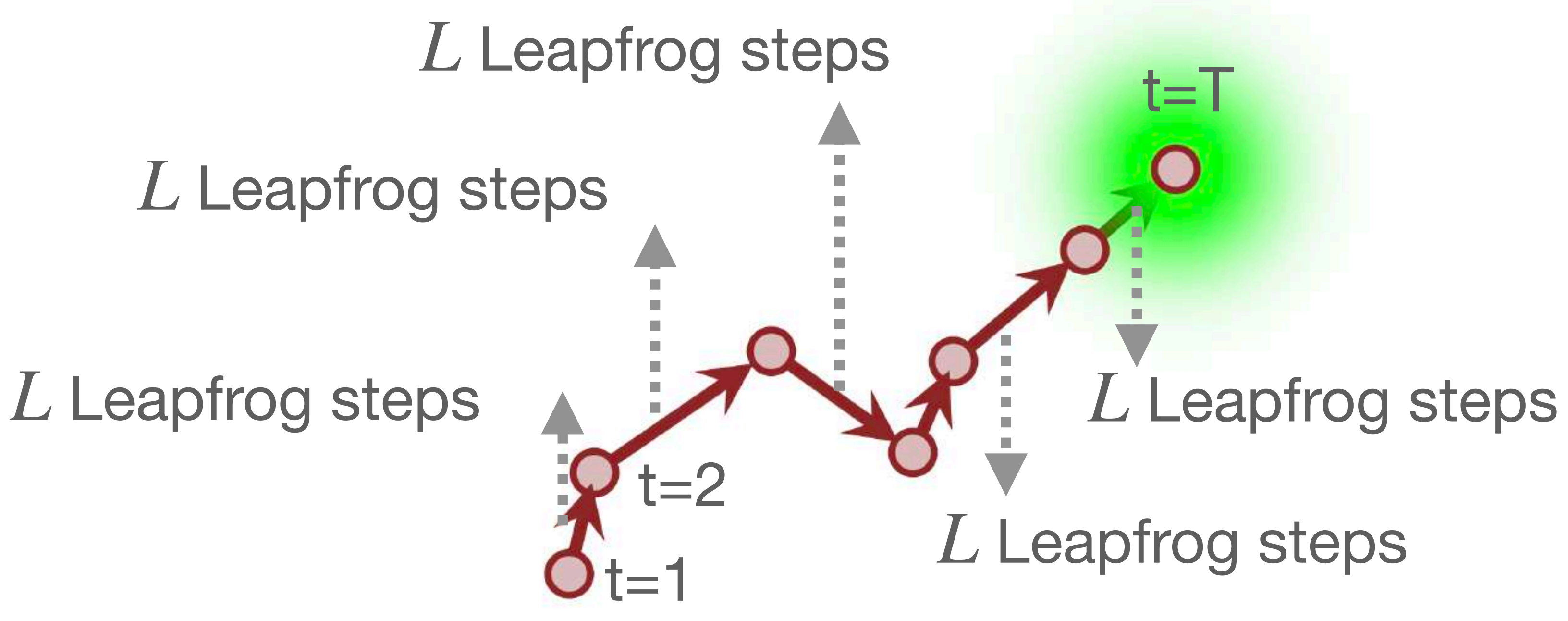
$$\frac{d\boldsymbol{z}}{dt} = +\frac{\partial H}{\partial \boldsymbol{r}} = \frac{\partial K}{\partial \boldsymbol{r}}$$

$$\frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{z}} = -\frac{\partial K}{\partial \boldsymbol{z}} - \frac{\partial V}{\partial \boldsymbol{z}}$$

[1] Betancourt, 2017 [1] Neal., 2017

Hamiltonian Monte Carlo[1]

HMC in practice



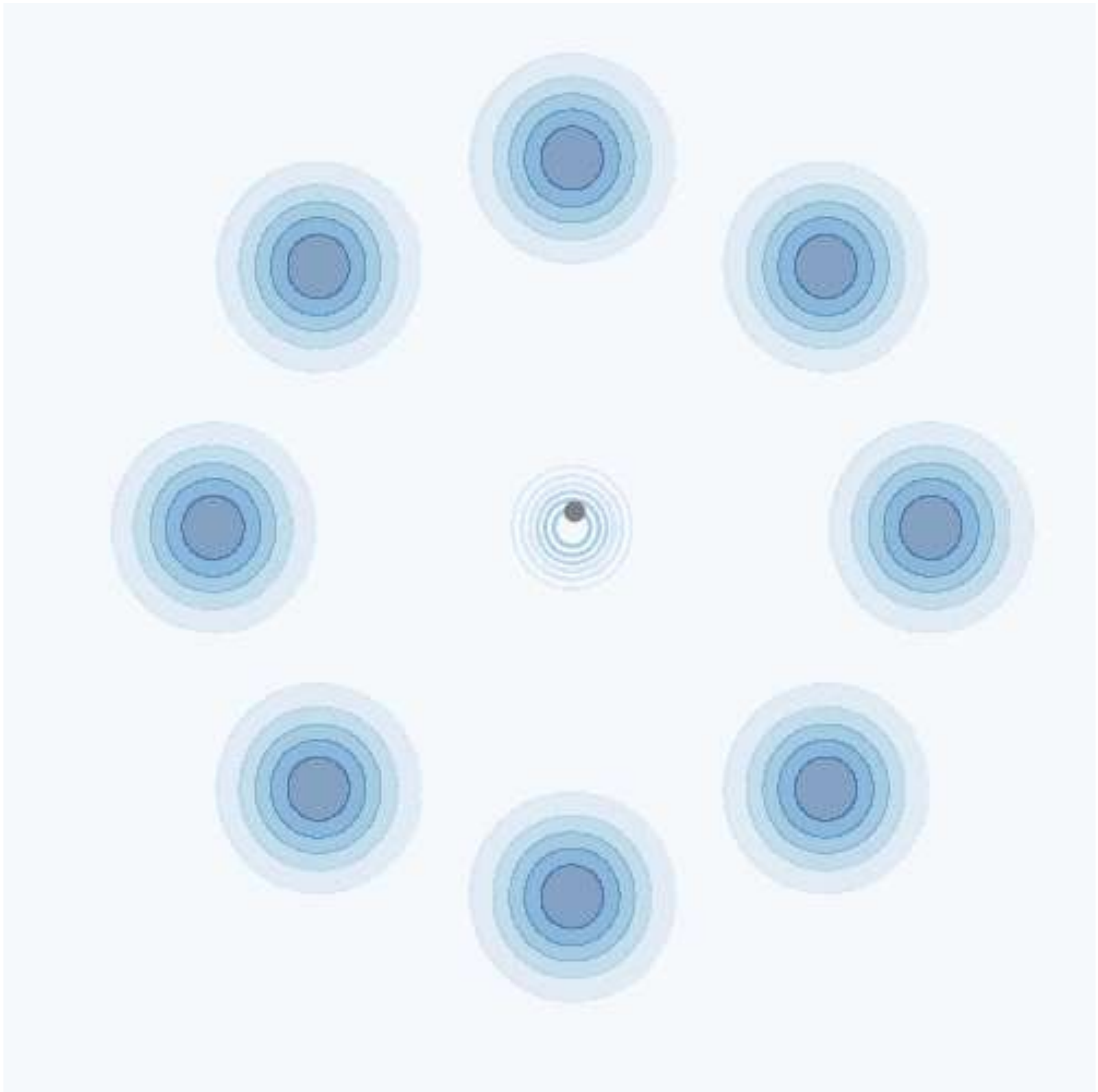
- Leapfrog integrator for Hamiltonian equations.
- Discrete trajectories (*chains*) of T updates.
- From an initial proposal, each update consists on L cyclic Leapfrog steps:

$$\boldsymbol{r}_{l+\frac{1}{2}} = \boldsymbol{r}_l + \frac{1}{2}\boldsymbol{\phi} \odot \nabla_{\boldsymbol{z}_l} \log p^*(\boldsymbol{z}_l)\,,$$

$$\boldsymbol{z}_{l+1} = \boldsymbol{z}_k + \boldsymbol{r}_{l+\frac{1}{2}} \odot \boldsymbol{\phi} \odot \frac{1}{\boldsymbol{M}}\,,$$

Hyperparameters

$$\boldsymbol{r}_{l+1} = \boldsymbol{r}_{l+\frac{1}{2}} + \frac{1}{2}\boldsymbol{\phi} \odot \nabla_{\boldsymbol{z}_{l+1}} \log p^*(\boldsymbol{z}_{l+1}),$$



- Ending in a new proposal $(\boldsymbol{z}', \boldsymbol{r}')$, which is accepted with probability:

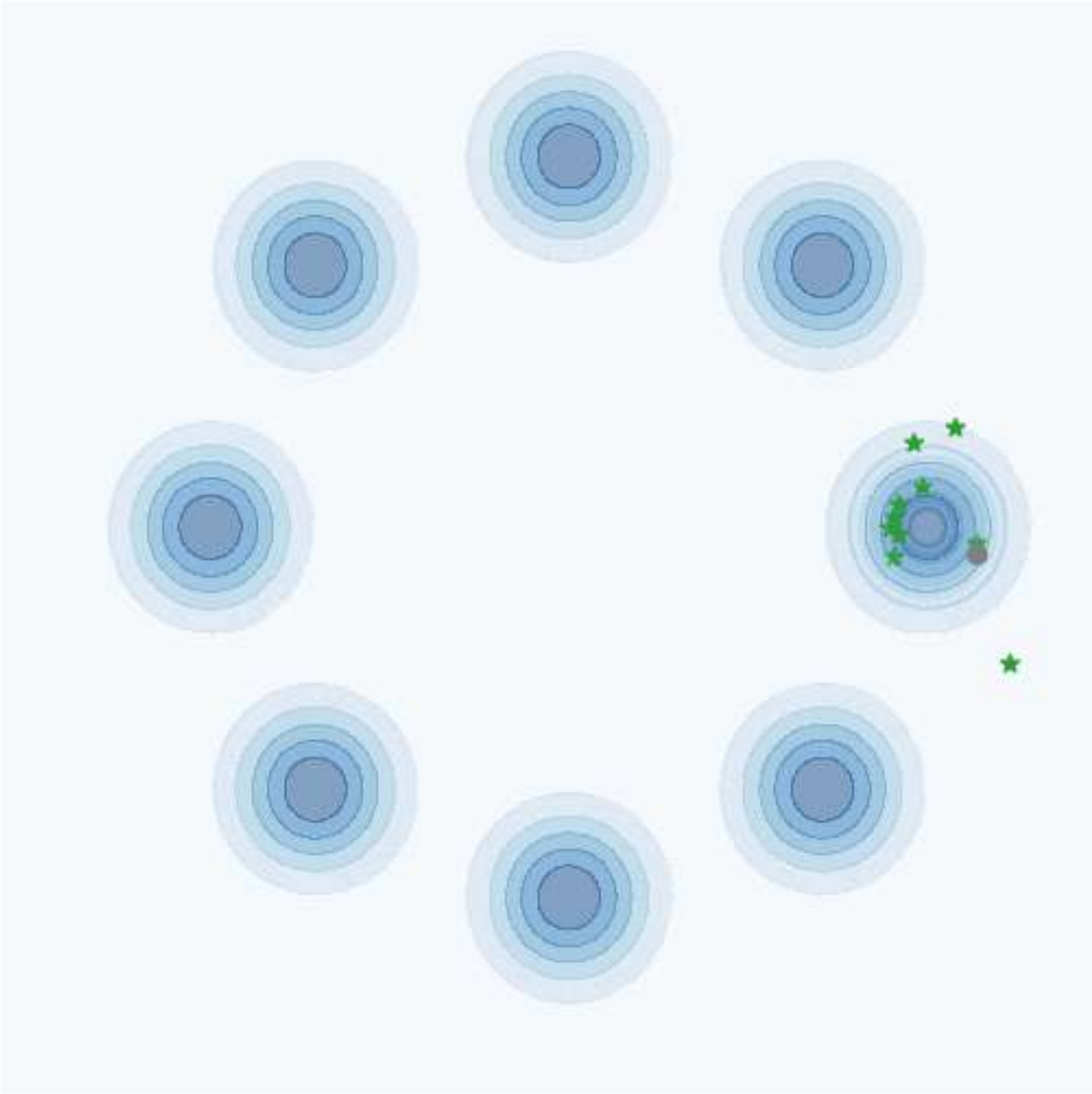
$$min\left[1, \exp(-H(\boldsymbol{z}', \boldsymbol{r}') + H(\boldsymbol{z}, \boldsymbol{r}))\right]$$

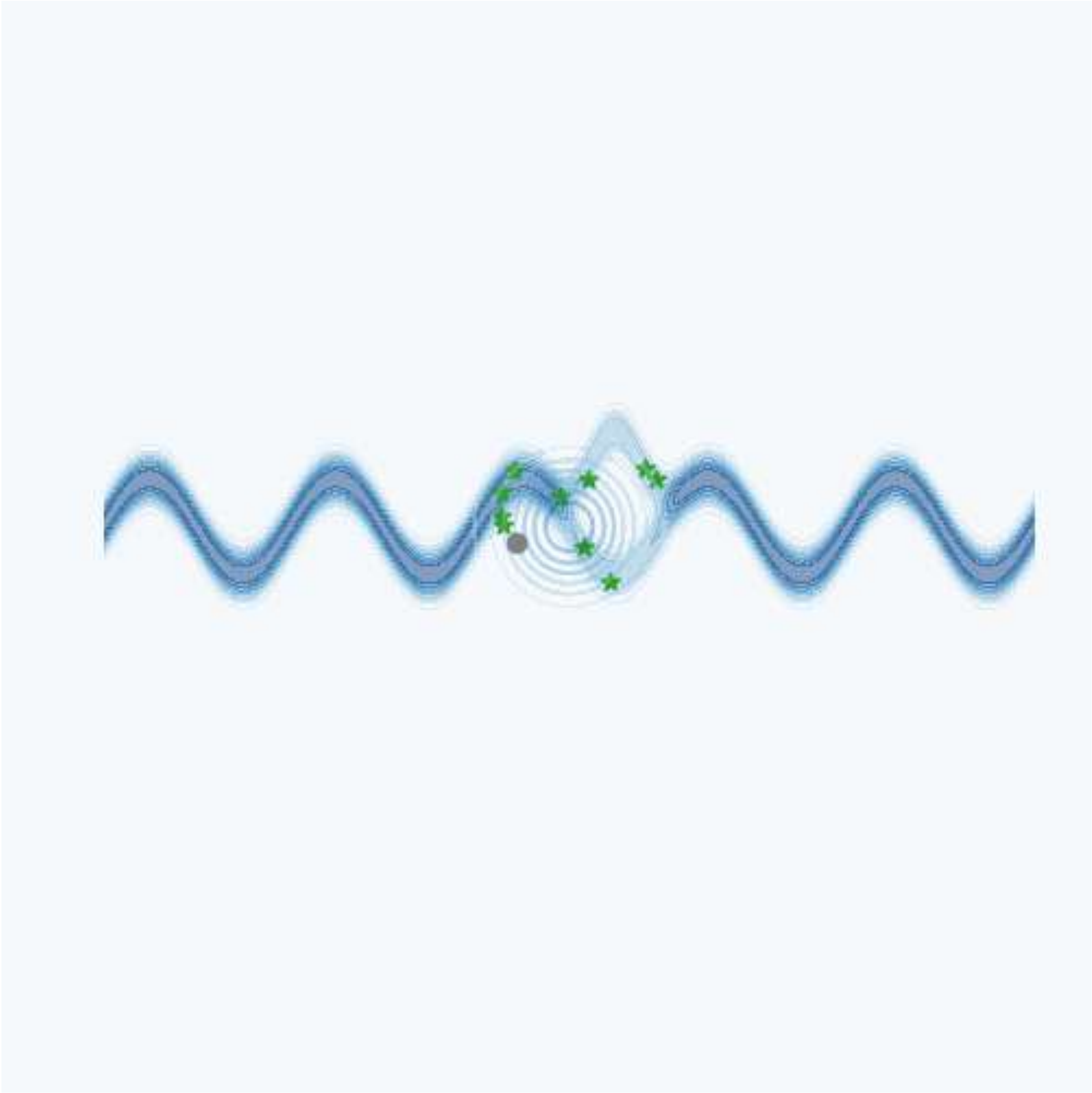
[1] Betancourt et at., 2017

Training Hamiltonian Monte Carlo [code]

ipeis

Effect of the hyperparameter choice





Training Hamiltonian Monte Carlo [code]

Gradient-based Optimization[1]

- Tuning the **hyperparameters** via Variational Inference:

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \ \mathbb{E}_{q_\phi^{(T)}(\boldsymbol{z})} [\log p^*(\boldsymbol{z})] \cancel{+ H\left[q_\phi^{(T)}(\boldsymbol{z})\right]}$$

Implicit

- Add an **inflation** parameter, $\mathbf{s}$, for scaling the proposal $q^{(0)}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{s}^2 \boldsymbol{\Sigma}_0)$

$$\boldsymbol{s}^* = \underset{s}{\operatorname{argmin}} \ d(q_\phi^{(T)}(\boldsymbol{z}), p(\boldsymbol{z}))$$

- Sliced Kernelized Stein Discrepancy[2] (SKSD) measures discrepancy between $p(\mathbf{z})$ and $q(\mathbf{z})$ using:
  - ✓ Samples of the approximated distribution
  - ✓ Gradients of the true target

$$\boldsymbol{s}^* = \underset{s}{\operatorname{argmin}} \ \mathrm{SKSD}(\boldsymbol{z}^{(T)}, \nabla_z \log p^*(\boldsymbol{z}))$$

  - ✓ Robust in high-dimensional spaces.

[1] Campbell et at., 2021 [2] Gong et at., 2020

Training Hamiltonian Monte Carlo [code]

Gradient-based Optimization

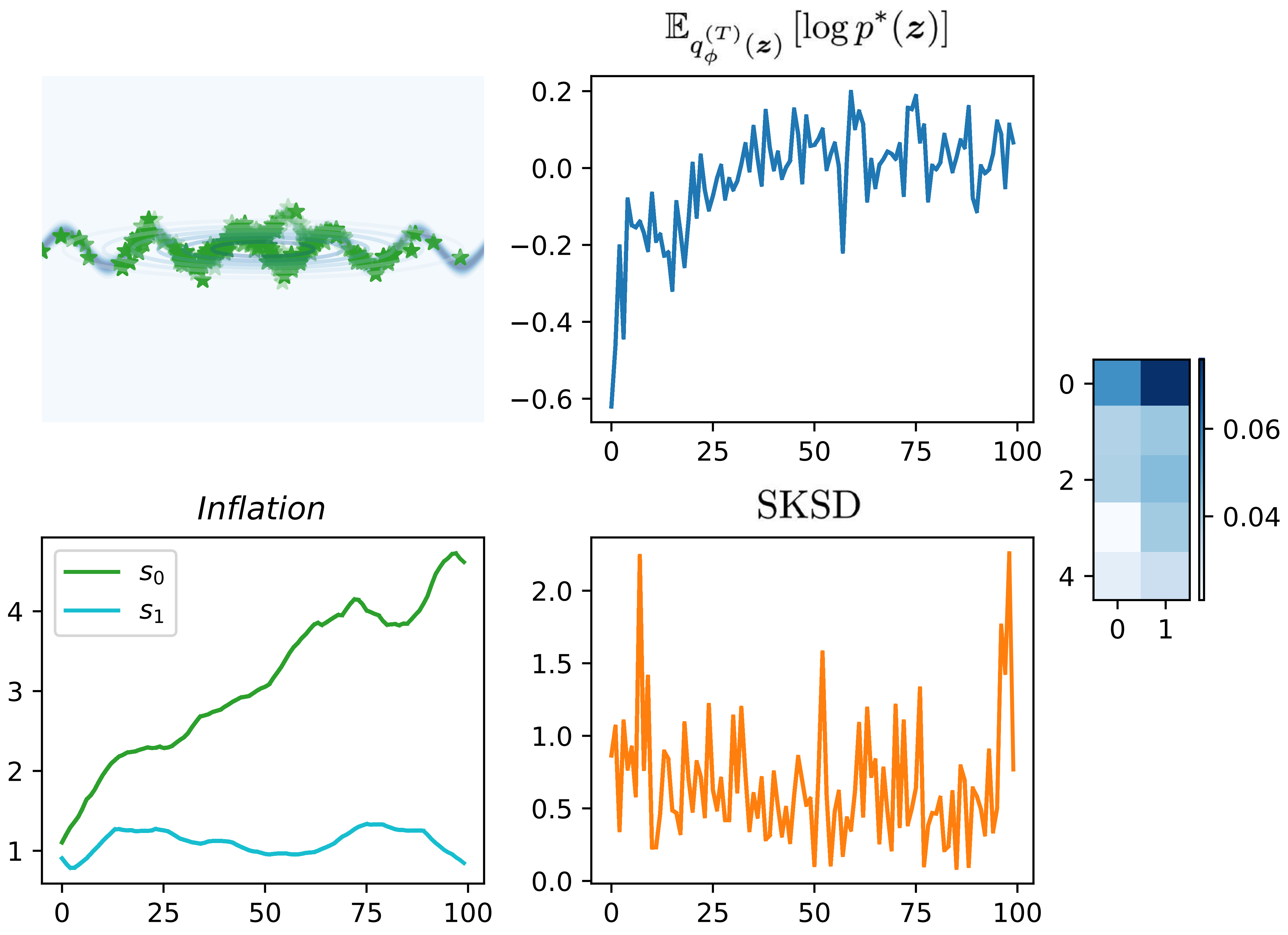
- For each step:

  1. Update HMC hyperparameters:

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \ \mathbb{E}_{q_\phi^{(T)}(\boldsymbol{z})} \left[\log p^*(\boldsymbol{z})\right]$$

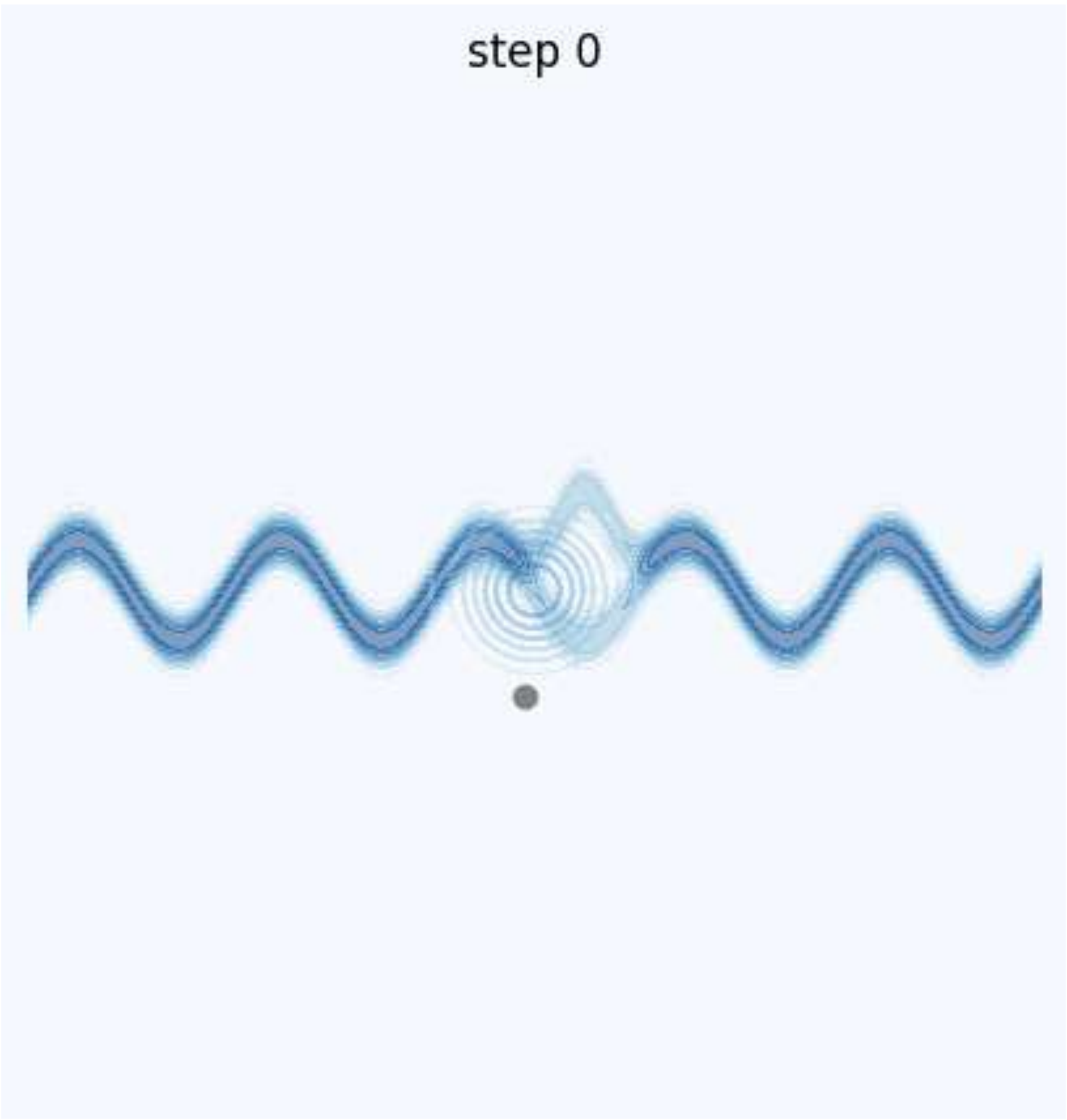
  2. Update scaling factor:

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmin}} \ \mathrm{SKSD}(\boldsymbol{z}^{(T)}, \nabla_z \log p^*(\boldsymbol{z}))$$

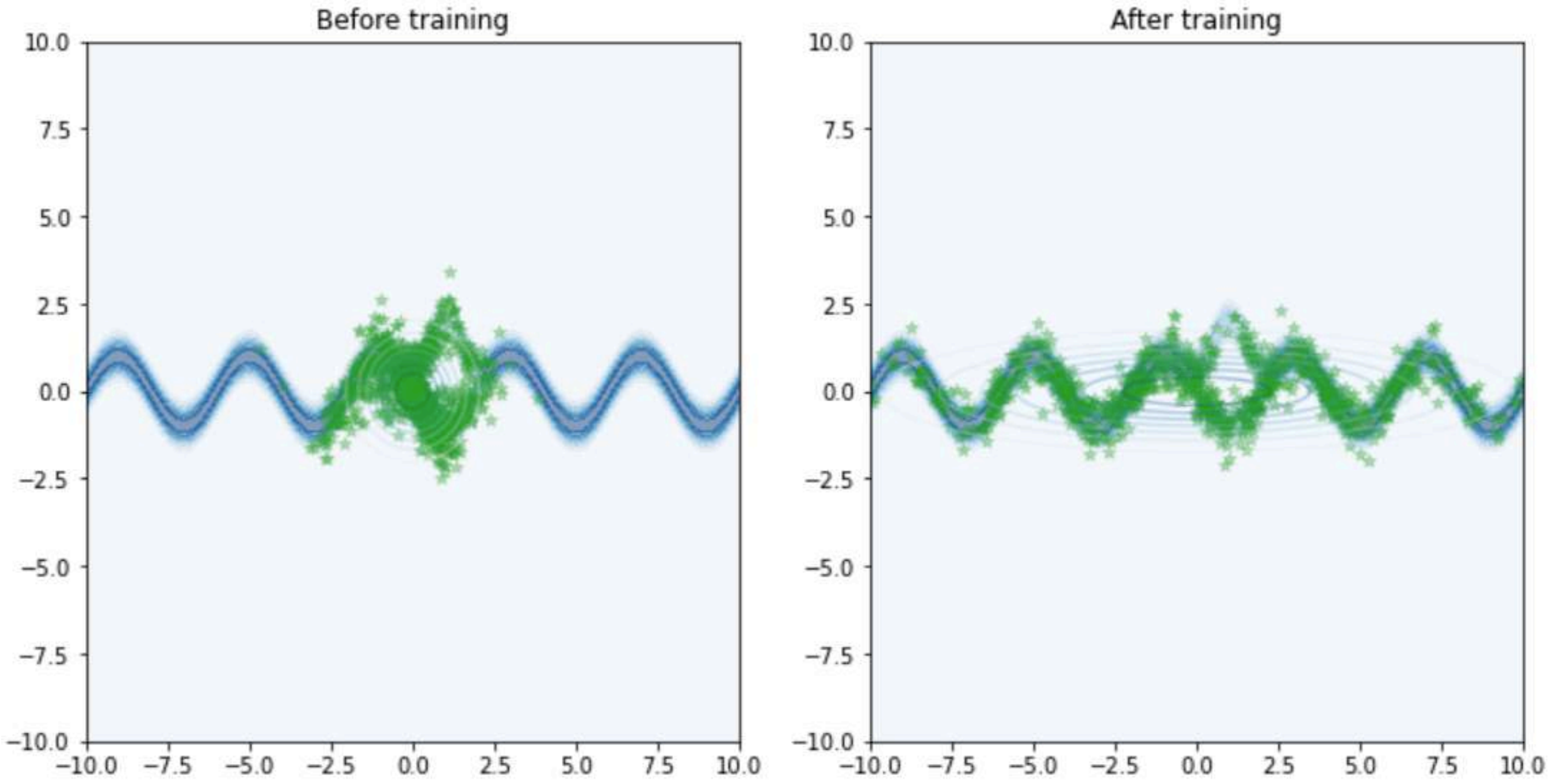


Training Hamiltonian Monte Carlo [code]

Gradient-based Optimization

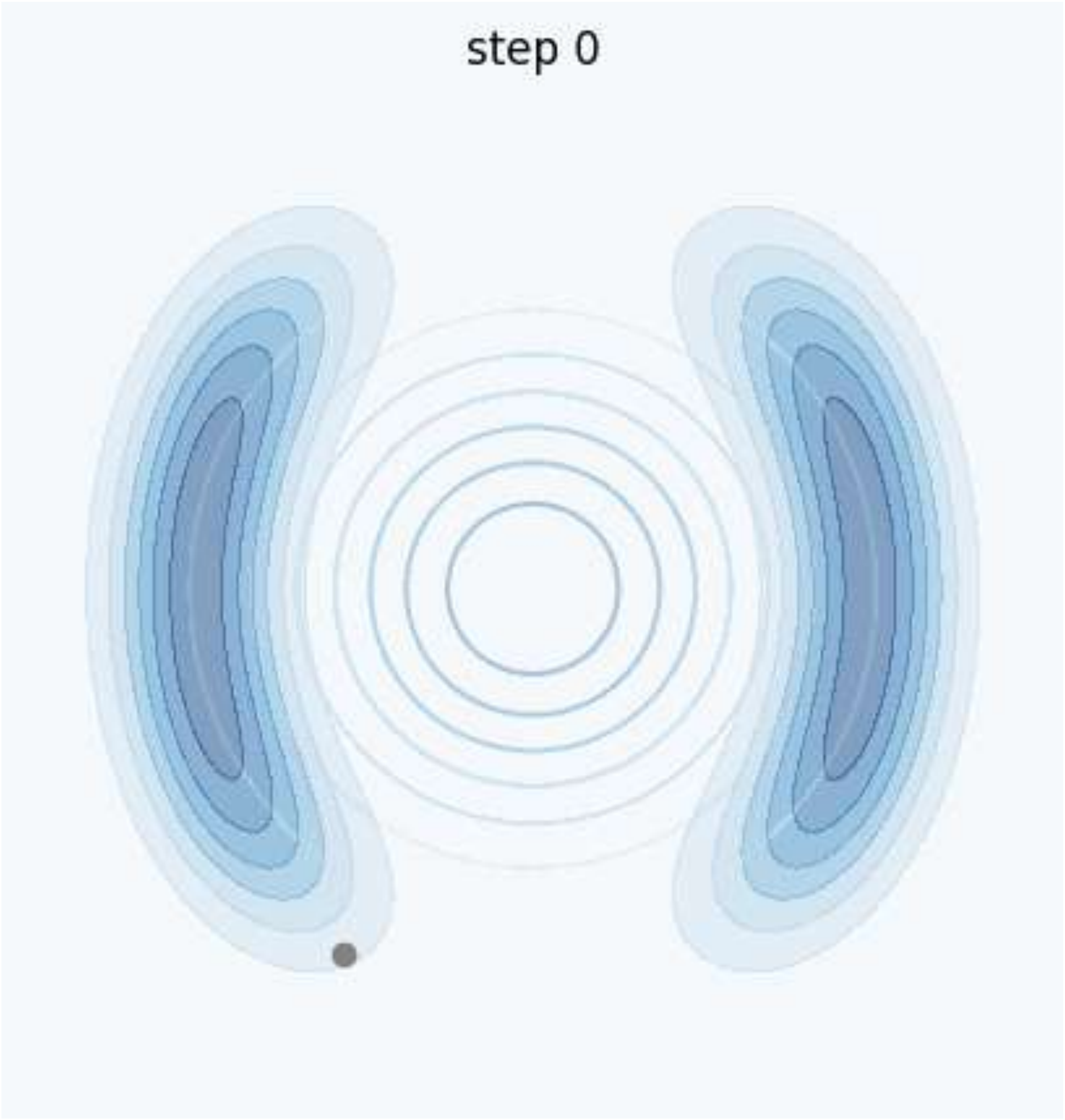


Optimization (gif)

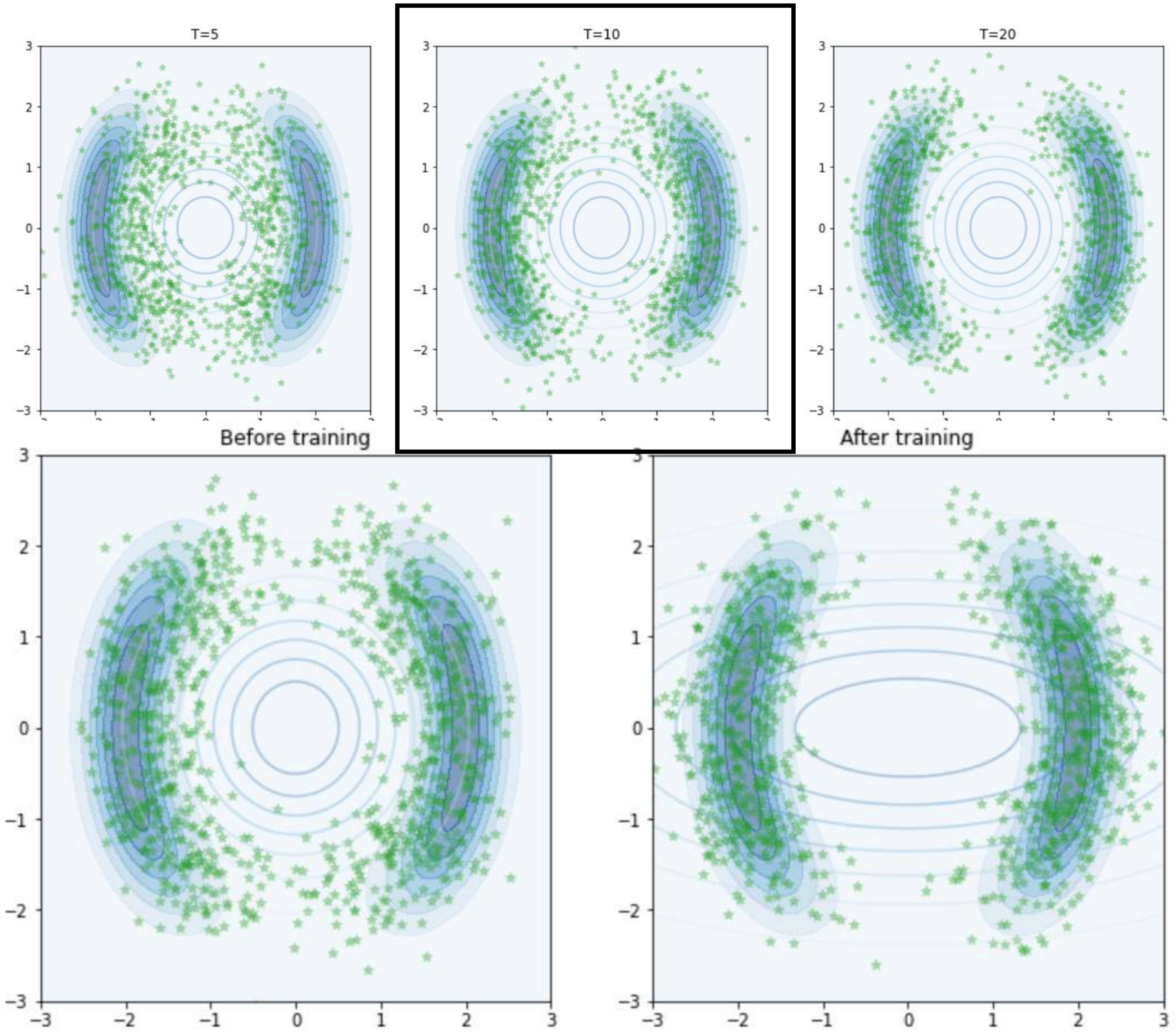


Training Hamiltonian Monte Carlo [code]

Gradient-based Optimization



Optimization (gif)



Hamiltonian Monte Carlo [code]

Application to VAEs[1,2]

- Approximation of $p(\mathbf{z}\,|\,\mathbf{x})$ improved with $q^{(T)}(\mathbf{z}\,|\,\mathbf{x})$, using as initial proposal $q^{(0)}(\mathbf{z}\,|\,\mathbf{x})$ given by the encoder.

Stage 1:

⋆ Pretrain VAE (θ, ψ) using ELBO.

Stage 2:

⋆ Keep training encoder parameters ψ using ELBO:

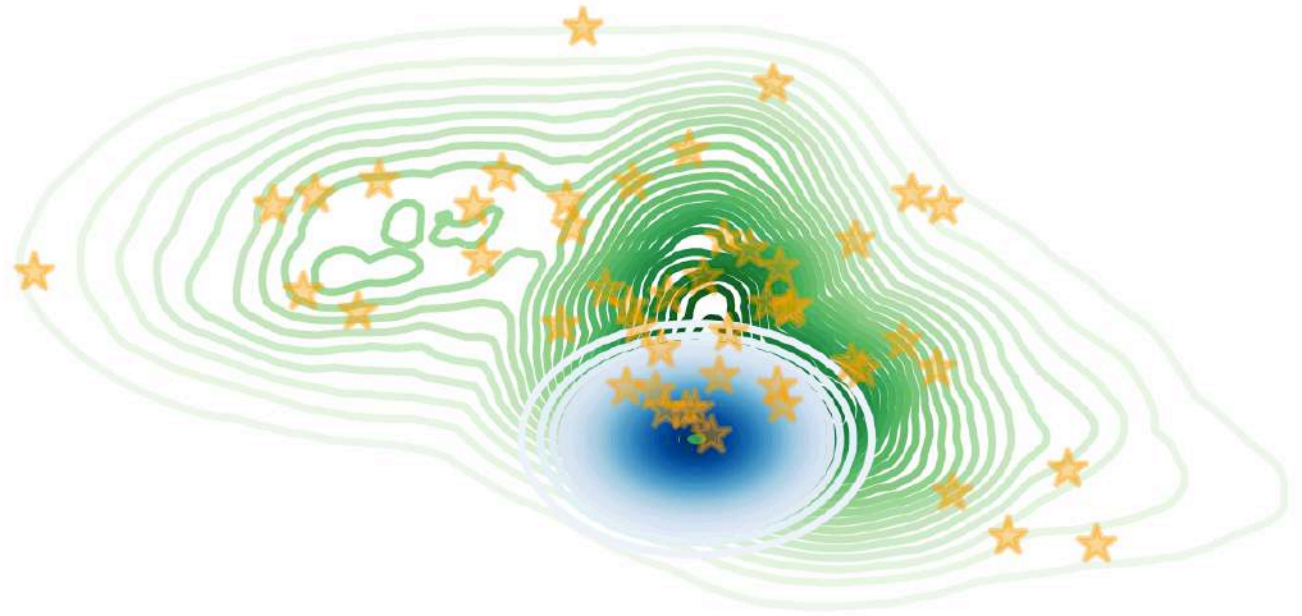
$$\psi^* \leftarrow \mathrm{Adam}_\psi\left(\log p_\theta\left(\boldsymbol{x} \mid \boldsymbol{z}^{(0)}\right) - D_{KL}\left(q_\psi^{(0)}(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})\right)\right)$$

⋆ Train decoder parameters θ and HMC hyperparameters ϕ:

$$(\theta, \phi)^* \leftarrow \mathrm{Adam}_{(\theta,\phi)}\left(\mathbb{E}_{q^{(T)}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{z}, \boldsymbol{x})\right]\right)$$

⋆ Train inflation parameter using:

$$\boldsymbol{s}^* \leftarrow \mathrm{Adam}_{\boldsymbol{s}}\left(\mathrm{SKSD}\left(\boldsymbol{z}^{(T)}, \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, \boldsymbol{x})\right)\right)$$



[2]Samples from the true posterior (orange)

Model	MNIST Scale	MNIST Mean	MNIST SE	Fashion MNIST Scale	Fashion MNIST Mean	Fashion MNIST SE
VAE	-	-85.08	0.22	-	-108.54	0.60
DReG-IWAE	-	-83.73	0.21	-	-104.48	0.58
maxELT $\alpha = 0$	1.0	-83.48	0.21	1.0	-104.08	0.58
maxELT $\alpha = 1$	1.0	-82.46	0.21	1.0	-103.57	0.58
maxELT $\alpha = 0$ SKSD	6.79	-81.91	0.20	5.58	-103.18	0.58
maxELT $\alpha = 1$ SKSD	3.90	-81.94	0.20	3.59	**-102.29**	0.57
Hoffman	-	**-81.74**	0.20	-	-103.04	0.58
Ruiz & Titsias	-	-82.45	0.21	-	-105.13	0.59
Salimans et al.	-	-81.94	-	-	-104.44	0.59
Caterini et al.	-	-82.62	-	-	-104.26	0.58

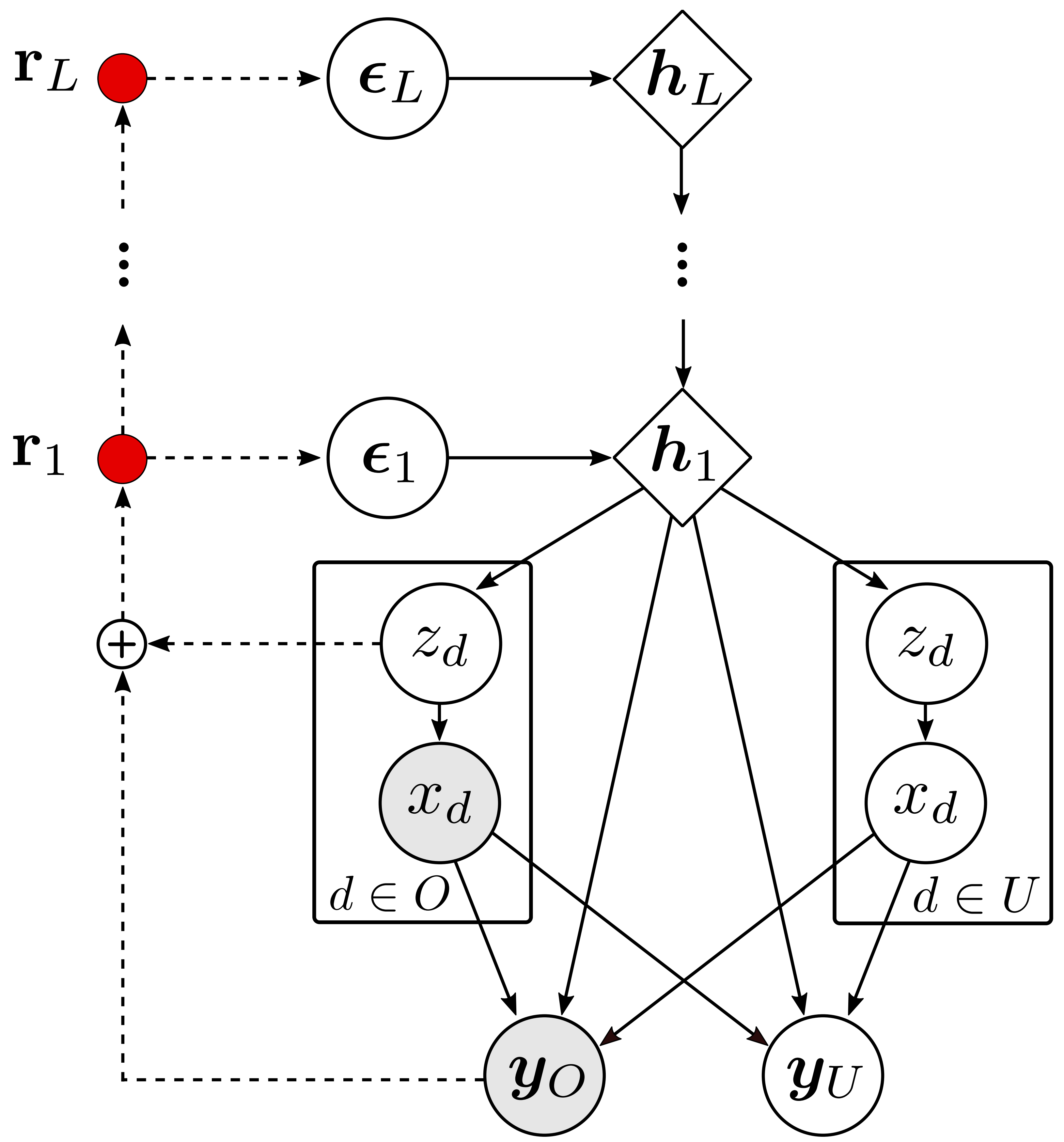
Approximated test $\log p(\mathbf{x})$

[1] Campbell et at., 2021 [2] Peis et at., 2022

HH-VAEM

The Hierarchical Hamiltonian VAE for Mixed-type incomplete data

- Generates data and predictions.
- Models heterogeneous, incomplete data.
- Flexible hierarchical latent space.
- Improved inference via tuned HMC.



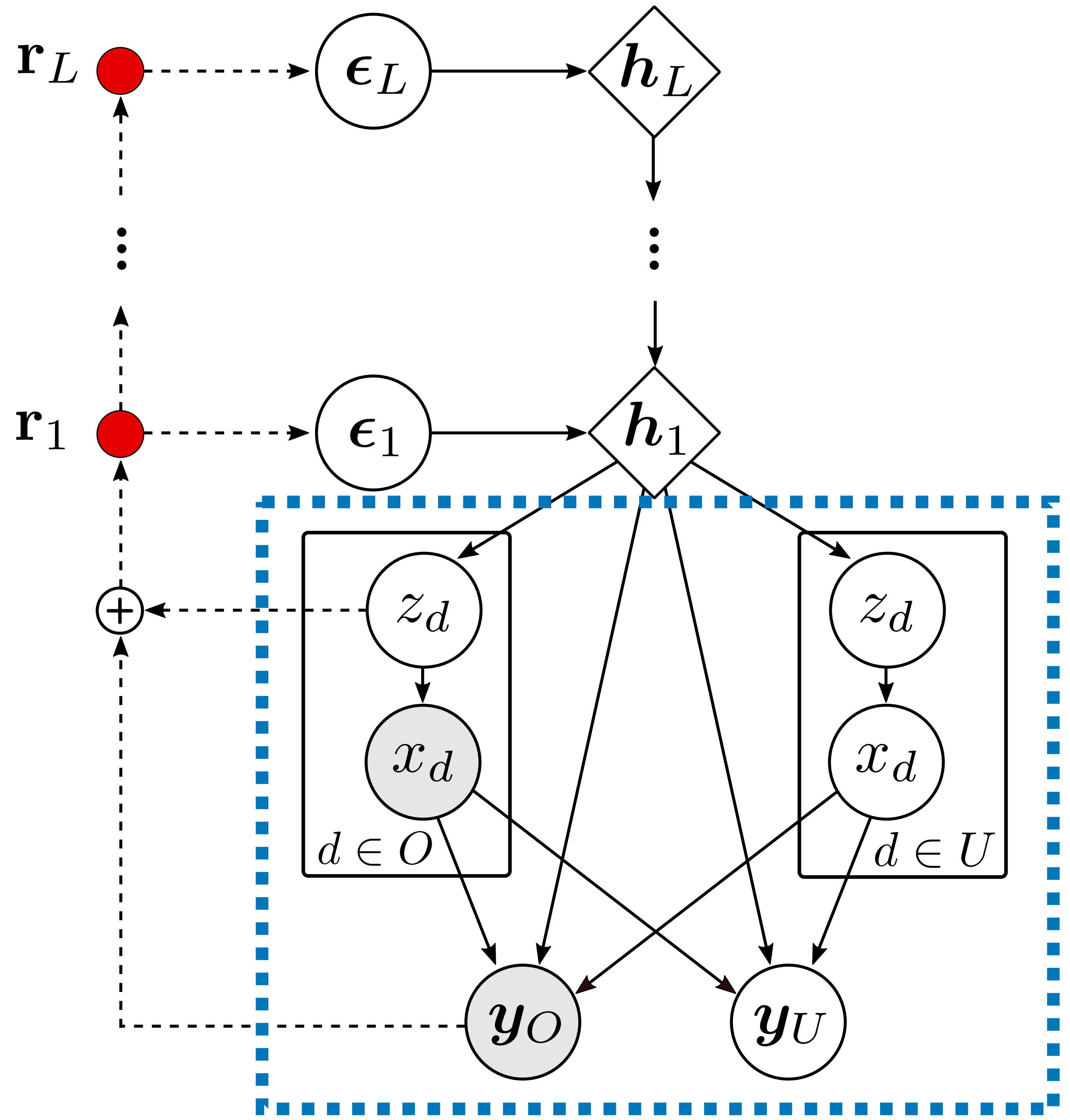
HH-VAEM

Mixed-type incomplete data

- Marginal VAEs (θ_d, γ_d) are pretrained independently on each dimension, with different likelihoods:

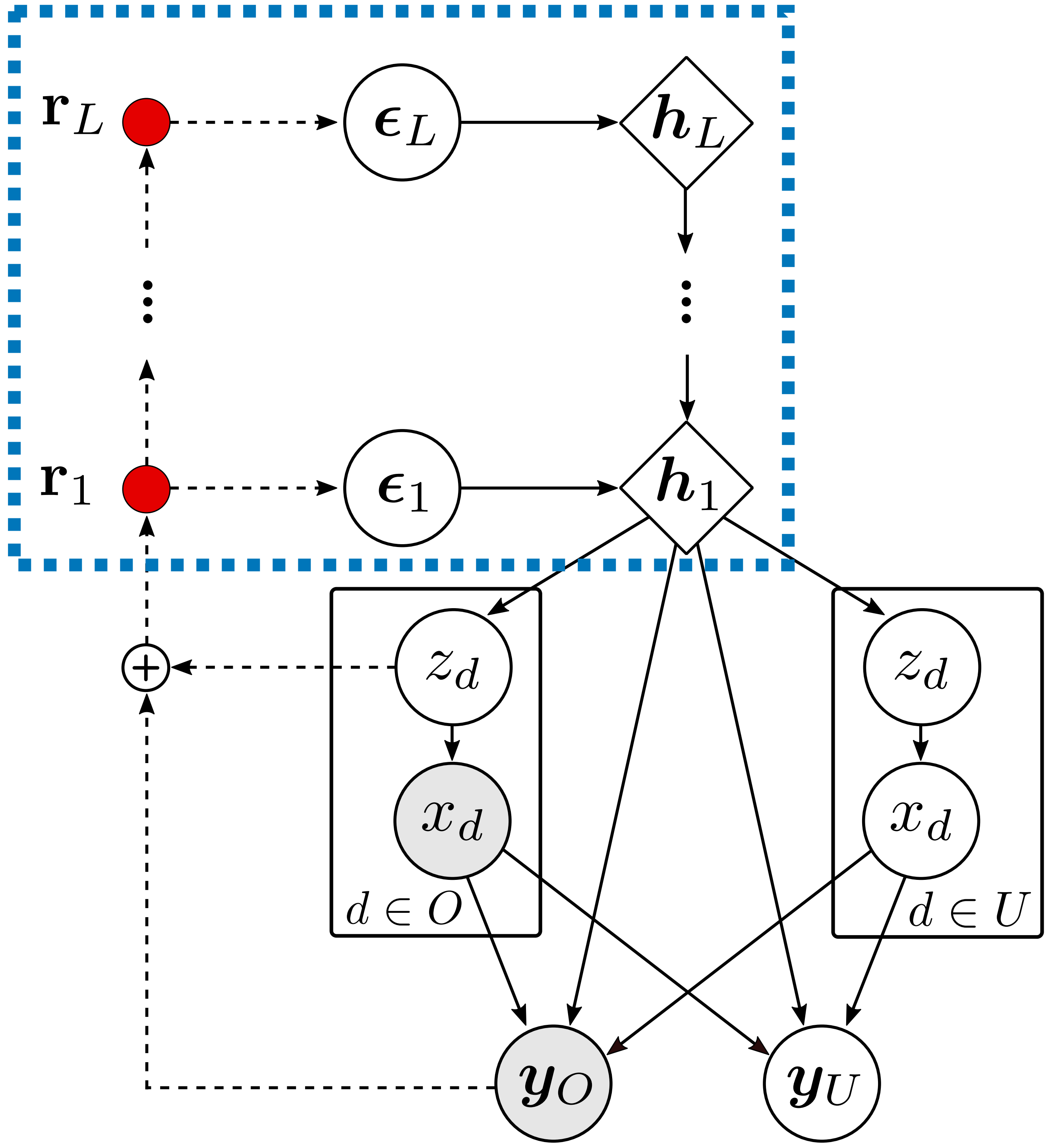
$$\mathcal{L}_d(x_d; \{\theta_d, \gamma_d\}) = \mathbb{I}(x_d \in \boldsymbol{x}_o)\mathbb{E}_{q_{\gamma_d}(z_d|x_d)} \log \frac{p_{\theta_d}(x_d, z_d)}{q_{\gamma_d}(z_d|x_d)}$$

- Dependency VAE over Gaussian factored dimensions allows dealing with partial heterogeneous data and capture dependencies from balanced likelihoods.



HH-VAEM

Hierarchical latent space



- Hierarchical latent space with L variables $\mathbf{h} = \{\mathbf{h}_1, \ldots, \mathbf{h}_L\}$.
- **Problem**[1]: HMC fails in densities with huge correlations (AR variables).

$$\nabla_{\boldsymbol{h}_{1:L}} \log p^*(\boldsymbol{h}) \uparrow\uparrow$$

- **Solution**:

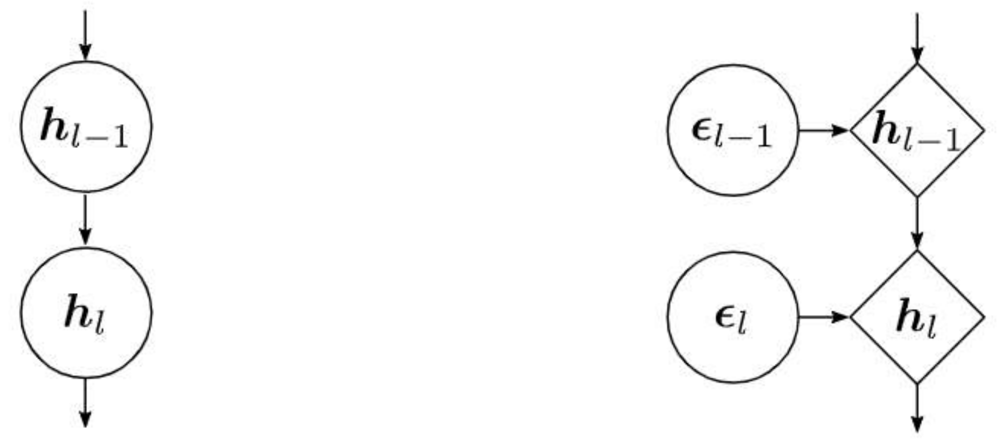
✓Reparameterization, deterministic hierarchy, relaxed posterior

$$\boldsymbol{h}_l = f_{\mu_l}(\boldsymbol{h}_{l+1}) + f_{\sigma_l}(\boldsymbol{h}_{l+1}) \cdot \boldsymbol{\epsilon}_l$$

NNs with parameters $\theta_{\mu_l} \to f_{\mu_l}$, $\theta_{\sigma_l} \to f_{\sigma_l}$. Then $\theta_l = \{\theta_{\mu_l}, \theta_{\sigma_l}\}$

✓Perform inference on $\epsilon = \{\epsilon_1, \ldots, \epsilon_1\}$ with standard Gaussian prior.

✓No need to increase complexity of the HMC method.



(a) AR hierarchy (b) Reparameterization

[1] Betancourt et at., 2015

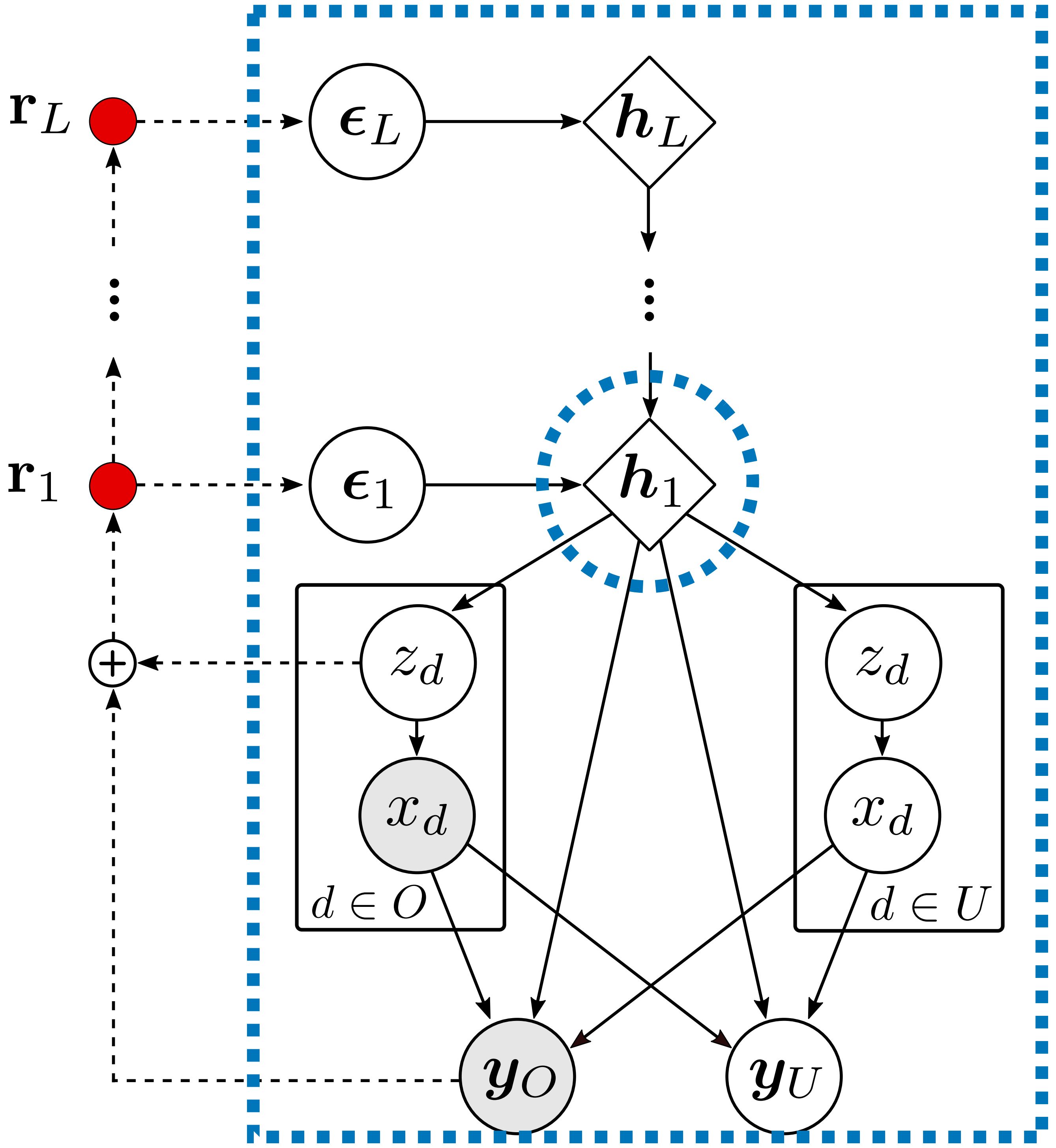
HH-VAEM

Generative model

- Generate from the superficial layer:

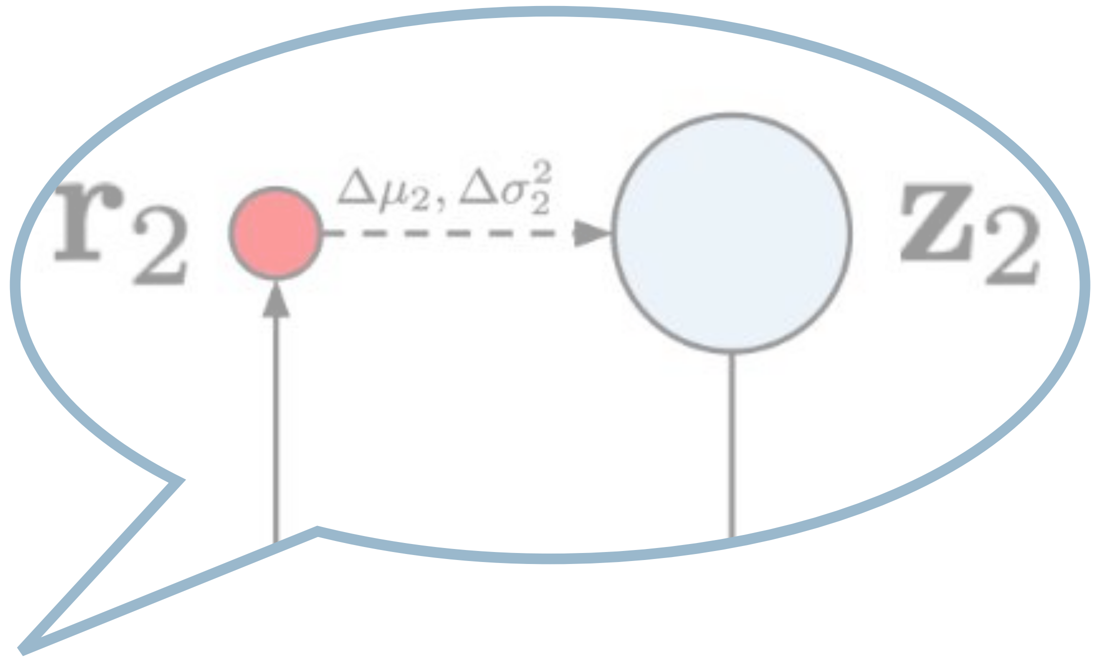
  1. Data: $p(\boldsymbol{z}|\boldsymbol{h}_1)$ as NN with parameters θ_z
  2. Predictions: $p(\boldsymbol{y}|\hat{\boldsymbol{x}}, \boldsymbol{h}_1)$ as NN with parameters θ_y

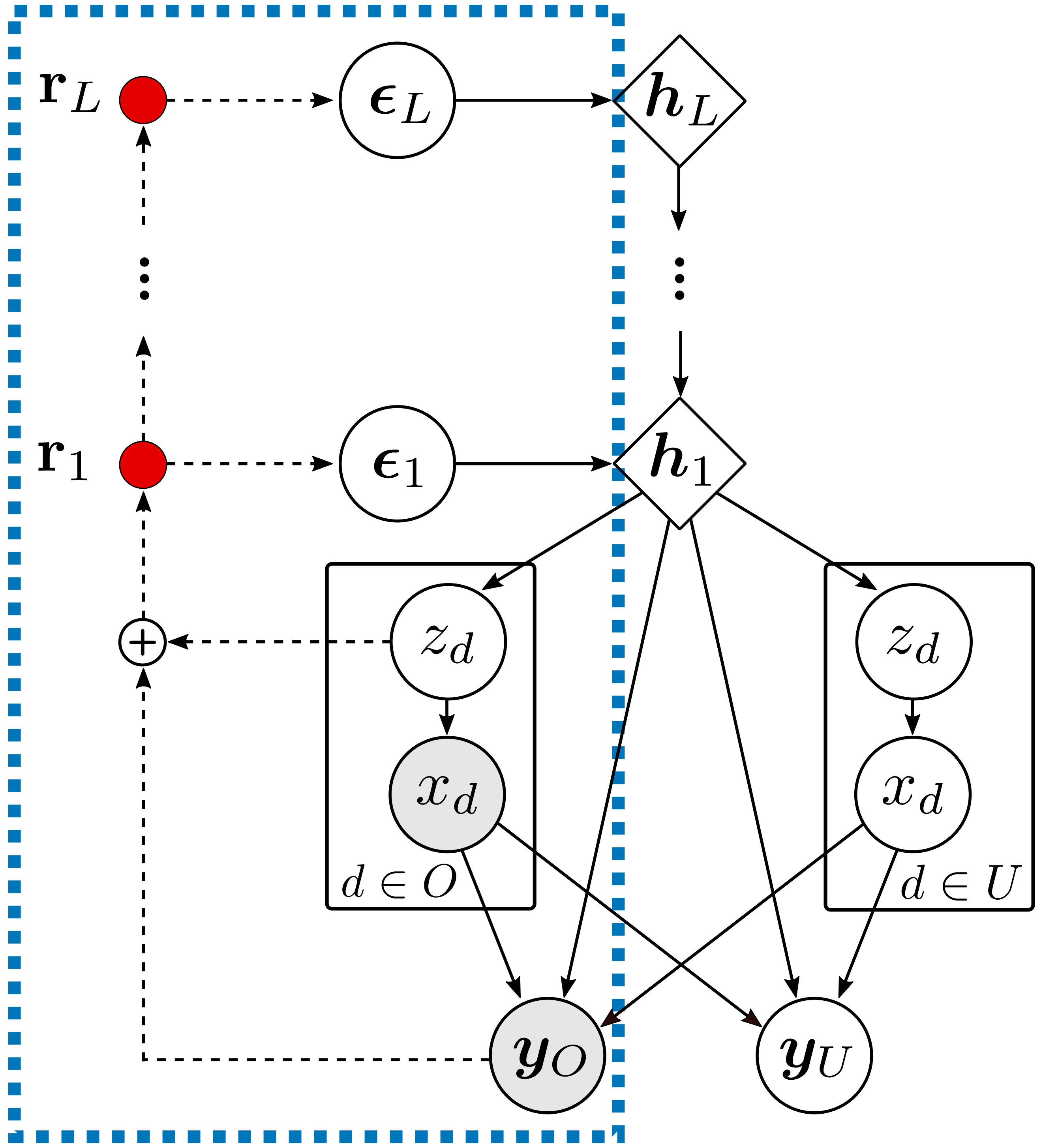
- $\hat{\boldsymbol{x}}$ is the concatenation $[\boldsymbol{x}_O, \hat{\boldsymbol{x}}_U]$ with the imputed missing part.
- The **predictor** parameters are jointly trained with the model.
- Generative parameters: $\theta = \{\theta_z, \theta_y, \theta_1, ...\theta_L\}$



HH-VAEM

Hierarchical encoder





- Bottom-up path, but no sharing parts needed! [1,2]

$$\boldsymbol{r}_0 = \{\boldsymbol{x}_O, \boldsymbol{y}_O\}, \qquad \boldsymbol{r}_l = f_r(\boldsymbol{r}_{l-1})$$

NNs with parameters $\psi_{r_l} \to f_{r_l}$.

- Posterior approximation for each layer:

$$q_{\psi_l}(\boldsymbol{\epsilon}_l | \boldsymbol{x}_O, \boldsymbol{y}_O) = \mathcal{N}(g_{\mu_l}(\boldsymbol{r}_l), g_{\sigma_l}(\boldsymbol{r}_l))$$

as NNs with parameters $\psi_{\mu_l} \to g_{\mu_l}$, $\psi_{\sigma_l} \to g_{\sigma_l}$

- Encoder parameters: $\psi = \{\psi_{r_l}, \psi_{\mu_l}, \psi_{\sigma_l}\}_{l=1}^{L}$

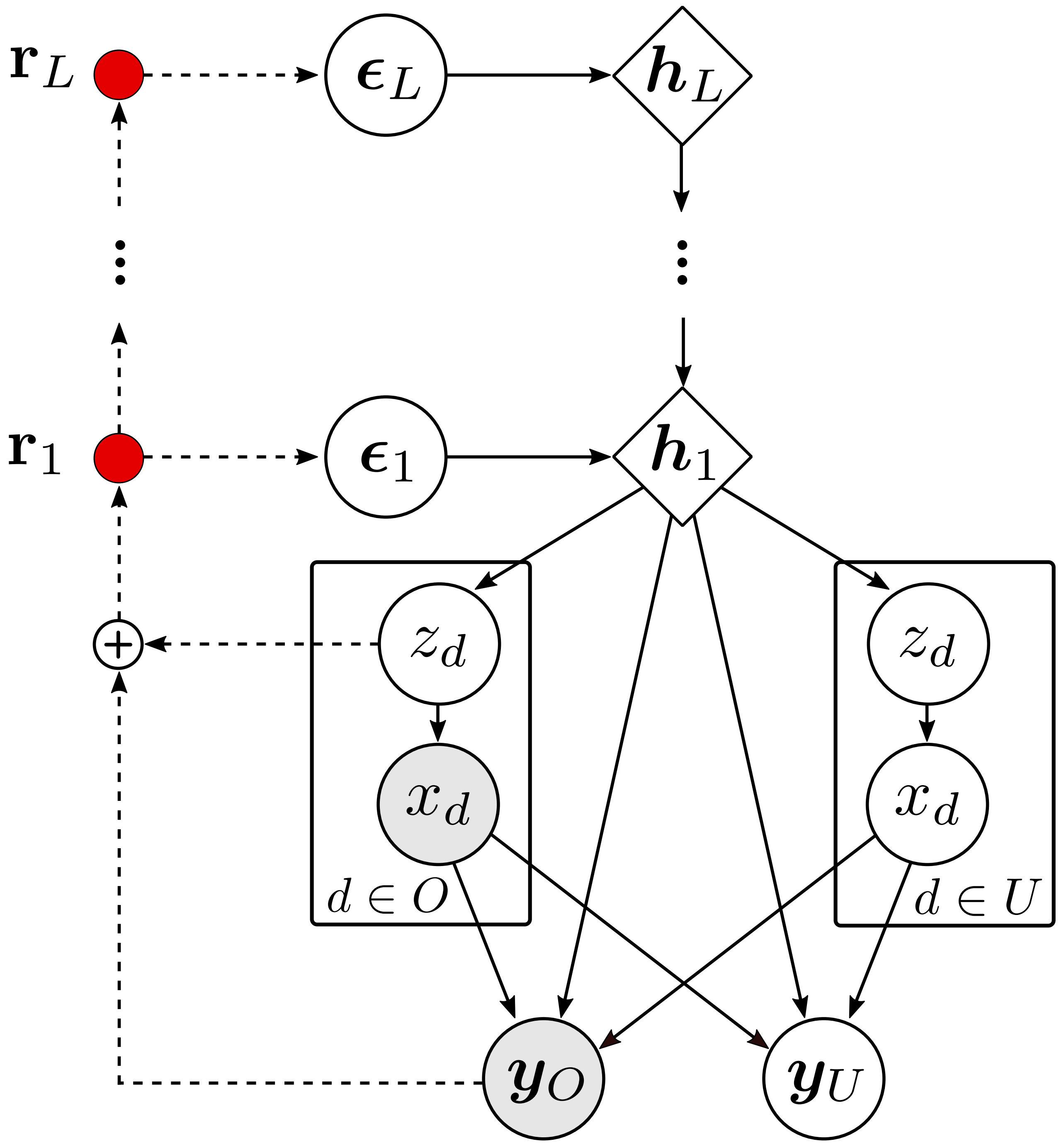
[1] Vahdat et at., 2020 [1] Maaløe et at., 2019

HH-VAEM

ELBO

$$\mathcal{L}_{VI}(\boldsymbol{x}_O, \boldsymbol{y}_O; \{\theta, \psi\}) = \mathbb{E}_{q_\psi}\left[\log \frac{p_\theta(\boldsymbol{z}_O, \boldsymbol{\epsilon})}{q_\psi(\boldsymbol{\epsilon}|\boldsymbol{z}_O, \boldsymbol{x}_O, \boldsymbol{y}_O)}\right] =$$

$$\mathbb{E}_{q_\psi}\left[\log p_\theta(\boldsymbol{z}_O|\boldsymbol{h}_1) + \log p_\theta(\boldsymbol{y}_O|\hat{\boldsymbol{x}}, \boldsymbol{h}_1)\right] - \sum_{l=1}^{L} D_{\mathrm{KL}}\left(q_\psi(\boldsymbol{\epsilon}_l|\boldsymbol{x}_O, \boldsymbol{y}_O)||p(\boldsymbol{\epsilon}_l)\right)$$



- The sum of KLs needs extra manipulation: balance during an warming stage[1]:

$$\gamma_l = \frac{d_l \; \mathbb{E}_{x\sim B}\left[\mathrm{KL}(q(\boldsymbol{\epsilon}_l|\boldsymbol{x})||p(\boldsymbol{\epsilon}))\right]}{\sum_{i=1}^{L} d_i \; \mathbb{E}_{x\sim B}\left[\mathrm{KL}(q(\boldsymbol{\epsilon}_i|\boldsymbol{x})||p(\boldsymbol{\epsilon}))\right]}$$

[1] Vahdat et at., 2020

HH-VAEM

Training HMC on the posterior distribution

- Include HMC with hyper parameters ϕ.
- Define HMC target as the unnormalised posterior for sampling $\boldsymbol{\epsilon} = \{\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_L\}$ from the posterior:

$$p^*(\boxed{\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_L}, \boldsymbol{z}_O, \boldsymbol{y}_O) = p_\theta(\boldsymbol{z}_O|\boldsymbol{h}_1)\, p_\theta(\boldsymbol{y}_O|\hat{\boldsymbol{x}}, \boldsymbol{h}_1) \prod_{i=1}^{L} p(\boldsymbol{\epsilon}_{\boldsymbol{l}})$$

- Define the HMC objective as

$$\mathcal{L}_{HMC}(\boldsymbol{z}_O, \boldsymbol{y}_O; \{\theta, \psi, \phi\}) = \mathbb{E}_{q_\phi^{(T)}(\boldsymbol{\epsilon})}\left[\log p_\theta(\boldsymbol{z}_O|\boldsymbol{h}_1) + \log p_\theta(\boldsymbol{y}_O|\hat{\boldsymbol{x}}, \boldsymbol{h}_1) + \sum_{l=1}^{L} p(\boldsymbol{\epsilon}_l^{(T)})\right]$$

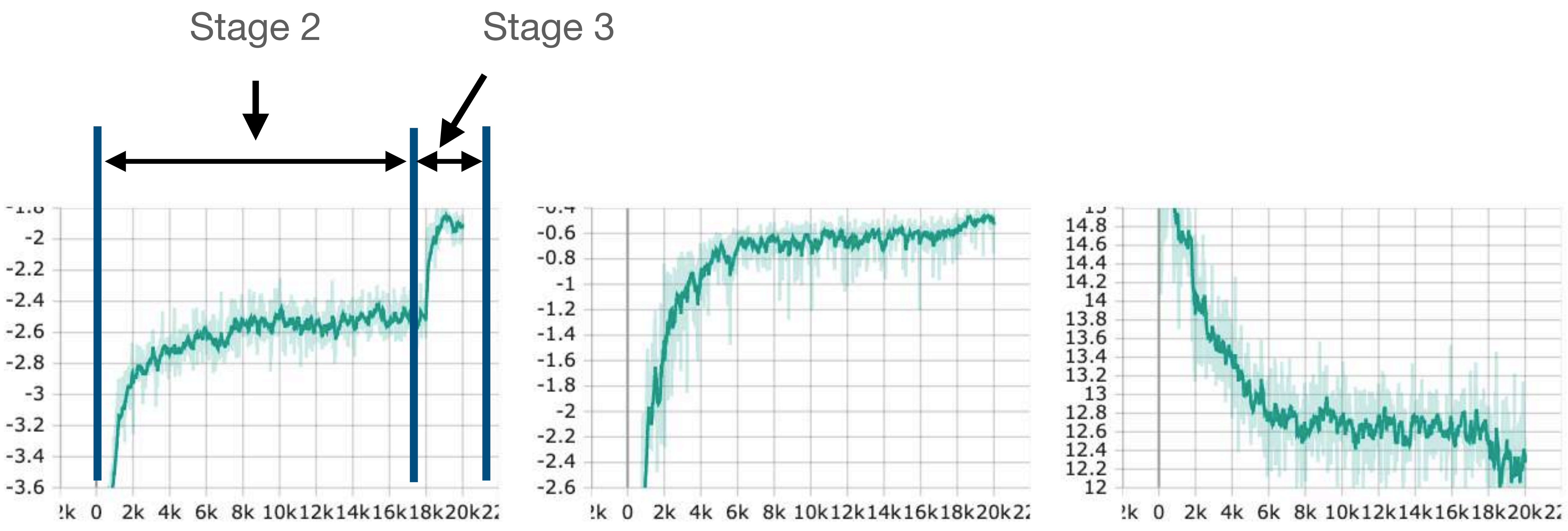
- Tune the scale factor s using the SKSD discrepancy:

$$\mathcal{L}_{SKSD}(\boldsymbol{x}_O, \boldsymbol{y}_O; \boldsymbol{s}) = \text{SKSD}\left(q_\phi^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{z}_O, \boldsymbol{x}_O, \boldsymbol{y}_O; \boldsymbol{s}),\; p(\boldsymbol{\epsilon}|\boldsymbol{z}_O, \boldsymbol{x}_O, \boldsymbol{y}_O)\right)$$

HH-VAEM

Training algorithm

- **Stage 1**: train marginal VAEs on each dimension.
- **Stage 2**: pretrain using ELBO.
- **Stage 3**: jointly train VAE + HMC.



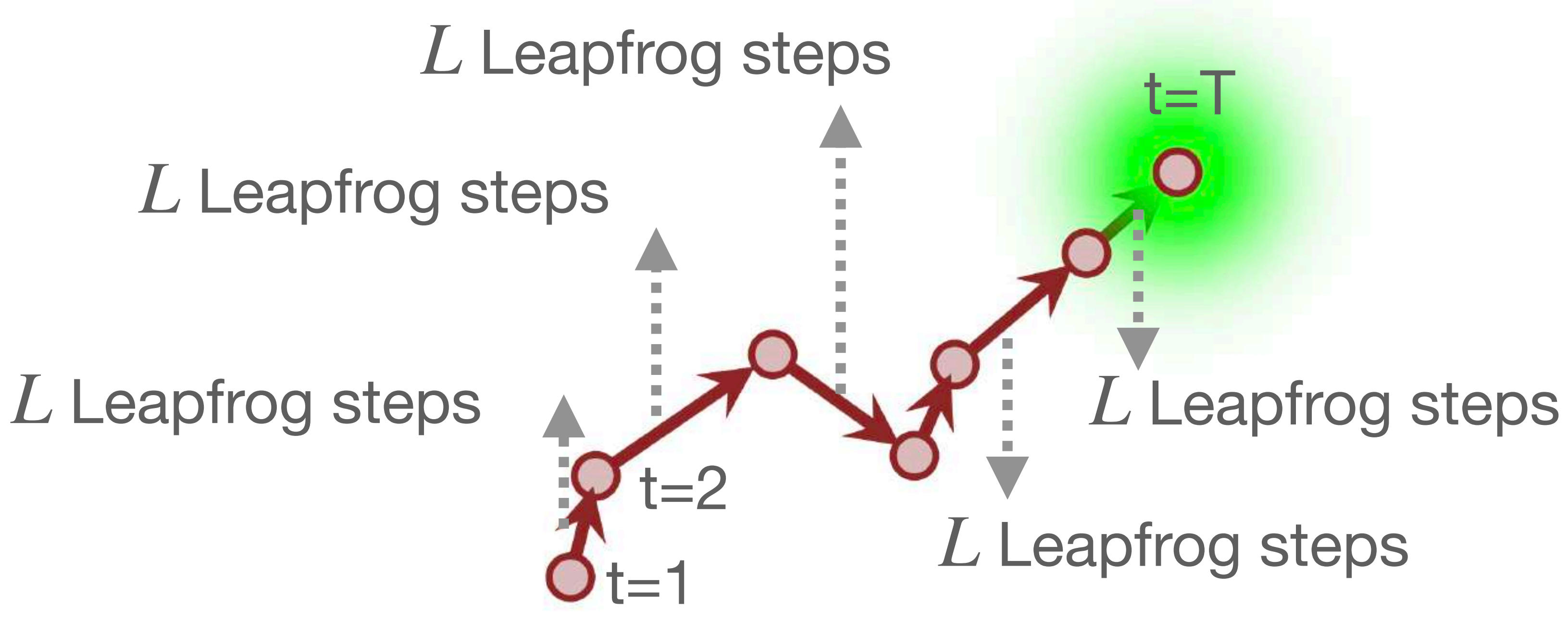
(a) $\log p(\boldsymbol{x}_U|\boldsymbol{x}_O)$ (b) $\log p(\boldsymbol{y}|\boldsymbol{x}_O)$ (c) RMSE

Algorithm 1 Training algorithm for HH-VAEM

Input: data $\left(\boldsymbol{x}_O^{(1:N)}, \boldsymbol{y}_O^{(1:N)}\right)$, steps: T_d, T_{VI}, T_{HMC}
Parameters: $\gamma, \theta, \psi, \phi, s$
STAGE 1: MARGINAL VAES
for $d = 1$ **to** D **do**
  Initialize marginal VAE $\{\theta_d, \gamma_d\}$
  for $t = 1$ **to** T_d **do**
    $\gamma_d^{t+1}, \theta_d^{t+1} \leftarrow \text{Adam}_{\gamma_d^t, \theta_d^t}(\mathcal{L}_d)$
  end for
end for
STAGE 2: DEPENDENCY VAE
for $t = 1$ **to** T_{VAE} **do**
  $\theta^{t+1}, \psi^{t+1} \leftarrow \text{Adam}_{\theta^t, \psi^t}(\mathcal{L}_{VI})$
end for
STAGE 3: JOINTLY OPTIMIZING VAE + HMC
for $t = 1$ **to** T_{HMC} **do**
  $\psi^{t+1} \leftarrow \text{Adam}_{\psi^t}(\mathcal{L}_{VI})$
  $\theta^{t+1}, \phi^{t+1} \leftarrow \text{Adam}_{\theta^t, \phi^t}(\mathcal{L}_{HMC})$
  $s^{t+1} \leftarrow \text{Adam}_{s^t}(\mathcal{L}_{SKSD})$
end for

HH-VAEM

Computational cost



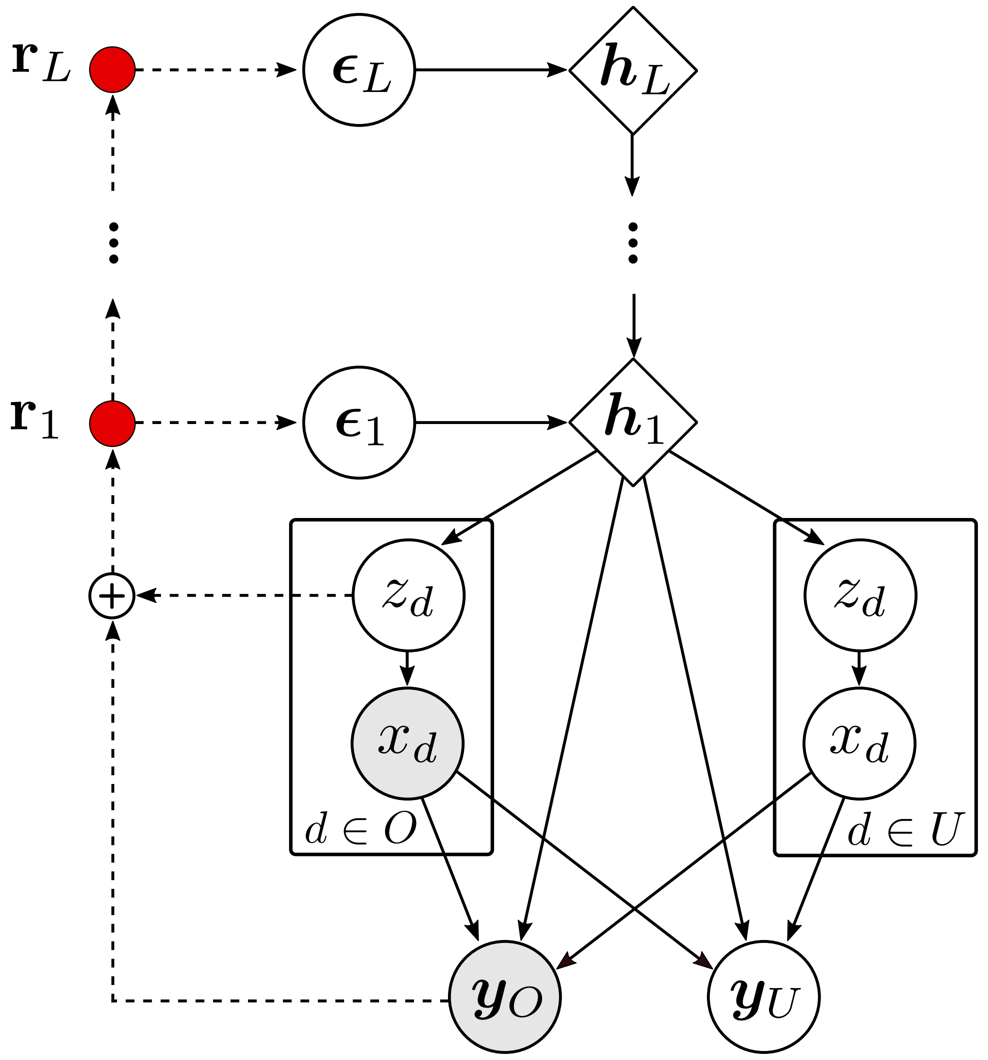
- The L Leapfrog steps are executed each cycle in $t = 1 : T$

(batch_size, parallel_chains, latent_dimension)

$$\boldsymbol{r}_{i+\frac{1}{2}} = \boldsymbol{r}_i + \frac{1}{2}\boldsymbol{\phi} \odot \boxed{\nabla_{\epsilon_i}} \log p^* \boxed{(\boldsymbol{\epsilon}_i)},$$

$$\boldsymbol{\epsilon}_{i+1} = \boldsymbol{\epsilon}_i + \boldsymbol{r}_{i+\frac{1}{2}} \odot \boldsymbol{\phi} \odot \frac{1}{\boldsymbol{M}},$$

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_{i+\frac{1}{2}} + \frac{1}{2}\boldsymbol{\phi} \odot \nabla_{\epsilon_{i+1}} \log p^*(\boldsymbol{\epsilon}_{i+1})$$



- Where $\log p^*(\boldsymbol{\epsilon}) = \log p(\boldsymbol{\epsilon}, \boldsymbol{z}, \boldsymbol{y}) = \log p(\boldsymbol{z}|\boldsymbol{h}_1) + p(\boldsymbol{y}|\boldsymbol{h}_1, \hat{\boldsymbol{x}}) + p(\boldsymbol{\epsilon})$
- Computing the gradients requires:
    1. Obtaining likelihood parameters (decoder for $p(\mathbf{z} \,|\, \mathbf{h}_1)$, predictor for $p(\mathbf{y} \,|\, \mathbf{h}_1, \hat{\mathbf{x}})$.
    2. Evaluating and perform the automatic differentiation.
- The extra computational cost is approximately a factor of $2TL$.

Bayesian active feature acquisition

Proposed method: sampling-based

- Bayesian reward[1] can also be expressed in terms of the Mutual Information:

$$R(i, \boldsymbol{x}_O) = D_{\mathrm{KL}}\left[p(\boldsymbol{y}, x_i|\boldsymbol{x}_O)||p(\mathbf{y}|\boldsymbol{x}_O)p(x_i|\boldsymbol{x}_O)\right] = \mathcal{I}(\boldsymbol{y}; x_i\ |\boldsymbol{x}_O) =$$
$$= \iint_{x_i, \boldsymbol{y}} p_{x_i, \boldsymbol{y}|\boldsymbol{x}_O}(x_i, \boldsymbol{y}|\boldsymbol{x}_O) \log\left(\frac{p_{x_i, \boldsymbol{y}|\boldsymbol{x}_O}(x_i, \boldsymbol{y}|\boldsymbol{x}_O)}{p_{x_i|\boldsymbol{x}_O}(x_i|\boldsymbol{x}_O)p_{\boldsymbol{y}|\boldsymbol{x}_O}(\boldsymbol{y}|\boldsymbol{x}_O)}\right)$$

- Sampling-based estimator of the Mutual Information[2]:

$$\hat{I}(\boldsymbol{y}; x_i\ |\boldsymbol{x}_O) \approx \sum_{ij} p_{x_i, \boldsymbol{y}|\boldsymbol{x}_O}(i, j) \log \frac{p_{x_i, \boldsymbol{y}|\boldsymbol{x}_O}(i, j)}{p_{x_i|\boldsymbol{x}_O}(i)p_{\boldsymbol{y}|\boldsymbol{x}_O}(j)}$$

✓ More flexible than the encoder-based method.

✓ Efficient, easy parallelization.

[1] Bernardo et at., 1979 [2] Kraskov et at., 2004

Experiments

Set up

- **HH-VAEM** with 2 layers of latent variables.
- Baseline models:
  - ***VAEM***: The VAEM[1] strategy.
  - ***MIWAEM***: VAEM combined with the importance weighted estimation proposed in MIWAE[2].
  - ***H-VAEM***: A Hierarchical VAEM with two layers of latent variables and a Gaussian encoder.
  - ***HMC-VAEM***: A VAEM that includes a tuned HMC sampler for the true posterior.
- Datasets:
  - MNIST, Fashion-MNIST -> without marginal VAEs.  
  - 10 UCI Datasets with mixed-type data: Bank, Insurance, Avocado, Naval, Yatch, Diabetes, Concrete, Wine, Energy, Boston.
- Configuration: manually introduced missing features and target with a probability sampled from $U(0.01, 0.99)$ each batch.

[1] Ma et at., 2020   [2] Mattei et at., 2019

Experiments

Mixed-type data

$$\log p(\boldsymbol{x}_U|\boldsymbol{x}_O) = \log \mathbb{E}_{\boldsymbol{\epsilon}\sim q^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{x}_O)}\left[p(\boldsymbol{x}_U|\boldsymbol{\epsilon})\right] \approx \log \frac{1}{k}\sum_i^k p(\boldsymbol{x}_U|\boldsymbol{\epsilon}_i)$$

	Bank	Insurance	Avocado	Naval	Yatch	Diabetes	Concrete	Wine	Energy	Boston
VAEM	2.84 ± 0.07	1.81 ± 0.03	1.89 ± 0.01	0.55 ± 0.05	3.15 ± 0.28	2.78 ± 0.16	2.45 ± 0.26	3.01 ± 0.61	2.09 ± 0.10	2.01 ± 0.23
MIWAEM	2.74 ± 0.05	1.88 ± 0.04	1.92 ± 0.04	0.57 ± 0.03	2.66 ± 0.11	2.55 ± 0.09	2.34 ± 0.51	2.76 ± 0.48	2.06 ± 0.14	1.94 ± 0.23
H-VAEM	2.82 ± 0.06	1.80 ± 0.04	1.89 ± 0.01	0.48 ± 0.06	3.06 ± 0.31	2.74 ± 0.09	2.42 ± 0.21	2.85 ± 0.56	1.72 ± 0.11	1.89 ± 0.24
HMC-VAEM	2.69 ± 0.05	1.77 ± 0.06	1.89 ± 0.02	0.49 ± 0.07	$\mathbf{2.21 \pm 0.24}$	2.72 ± 0.20	2.28 ± 0.29	2.83 ± 0.46	1.73 ± 0.05	1.83 ± 0.16
HH-VAEM	$\mathbf{2.63 \pm 0.04}$	$\mathbf{1.75 \pm 0.03}$	$\mathbf{1.88 \pm 0.05}$	$\mathbf{0.40 \pm 0.05}$	2.47 ± 0.27	$\mathbf{2.54 \pm 0.13}$	$\mathbf{2.28 \pm 0.09}$	$\mathbf{1.90 \pm 0.17}$	$\mathbf{1.71 \pm 0.04}$	$\mathbf{1.83 \pm 0.11}$

Table 1: Test negative log likelihood of the unobserved features for our model and baselines.

$$\log p(\boldsymbol{y}|\boldsymbol{x}_O) = \log \mathbb{E}_{\boldsymbol{\epsilon}\sim q^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{x}_O)}\left[p(\boldsymbol{y}|\boldsymbol{\epsilon})\right] \approx \log \frac{1}{k}\sum_i^k p(\boldsymbol{y}|\boldsymbol{\epsilon}_i),$$

	Bank	Insurance	Avocado	Naval	Yatch	Diabetes	Concrete	Wine	Energy	Boston
VAEM	0.56 ± 0.06	1.20 ± 0.03	1.18 ± 0.02	2.69 ± 0.01	0.61 ± 0.02	1.59 ± 0.19	1.07 ± 0.09	0.28 ± 0.09	0.61 ± 0.14	0.85 ± 0.21
MIWAEM	0.51 ± 0.03	1.15 ± 0.03	1.15 ± 0.03	2.70 ± 0.01	0.60 ± 0.03	$\mathbf{1.36 \pm 0.10}$	0.95 ± 0.22	0.28 ± 0.13	0.54 ± 0.12	0.80 ± 0.21
H-VAEM	0.50 ± 0.03	1.06 ± 0.02	1.18 ± 0.02	2.68 ± 0.01	0.60 ± 0.02	1.71 ± 0.14	1.02 ± 0.09	0.26 ± 0.11	0.46 ± 0.14	0.90 ± 0.22
HMC-VAEM	0.52 ± 0.02	1.00 ± 0.03	1.12 ± 0.03	2.71 ± 0.01	$\mathbf{0.52 \pm 0.15}$	1.55 ± 0.29	0.95 ± 0.26	0.28 ± 0.09	0.41 ± 0.07	0.71 ± 0.13
HH-VAEM	$\mathbf{0.49 \pm 0.03}$	$\mathbf{0.93 \pm 0.06}$	$\mathbf{1.10 \pm 0.01}$	$\mathbf{2.62 \pm 0.01}$	0.56 ± 0.02	1.38 ± 0.18	$\mathbf{0.95 \pm 0.08}$	$\mathbf{0.20 \pm 0.04}$	$\mathbf{0.32 \pm 0.05}$	$\mathbf{0.55 \pm 0.04}$

Table 2: Test negative log likelihood of the predicted target for our model and baselines.

Experiments

MNIST datasets

	VAE	MIWAE	H-VAE	HMC-VAE	**HH-VAE**
MNIST	0.124 ± 0.001	0.121 ± 0.001	0.119 ± 0.001	0.101 ± 0.004	$\mathbf{0.094 \pm 0.003}$
F-MNIST	0.162 ± 0.002	0.160 ± 0.002	0.156 ± 0.002	0.150 ± 0.002	$\mathbf{0.144 \pm 0.002}$

Table 3: Test negative log likelihood of the unobserved features for the MNIST datasets.

	VAE	MIWAE	H-VAE	HMC-VAE	**HH-VAE**
MNIST	0.153 ± 0.009	0.151 ± 0.007	0.146 ± 0.006	0.067 ± 0.007	$\mathbf{0.056 \pm 0.019}$
F-MNIST	0.501 ± 0.012	0.496 ± 0.008	0.494 ± 0.007	0.357 ± 0.060	$\mathbf{0.337 \pm 0.069}$

Table 4: Test negative log likelihood of the predicted target for the MNIST datasets.

	VAE	MIWAE	H-VAE	HMC-VAE	**HH-VAE**
MNIST	0.953 ± 0.004	0.953 ± 0.003	0.953 ± 0.003	0.978 ± 0.003	$\mathbf{0.981 \pm 0.005}$
F-MNIST	0.824 ± 0.005	0.824 ± 0.004	0.824 ± 0.004	0.869 ± 0.015	$\mathbf{0.876 \pm 0.017}$

Table 5: Test accuracy of the predicted digits for the MNIST datasets.

Experiments

Sequential Active Information Acquisition (SAIA)

- Sequentially acquiring high-value information by selecting features that maximize $\hat{I}(\boldsymbol{y}; x_i \mid \boldsymbol{x}_O)$

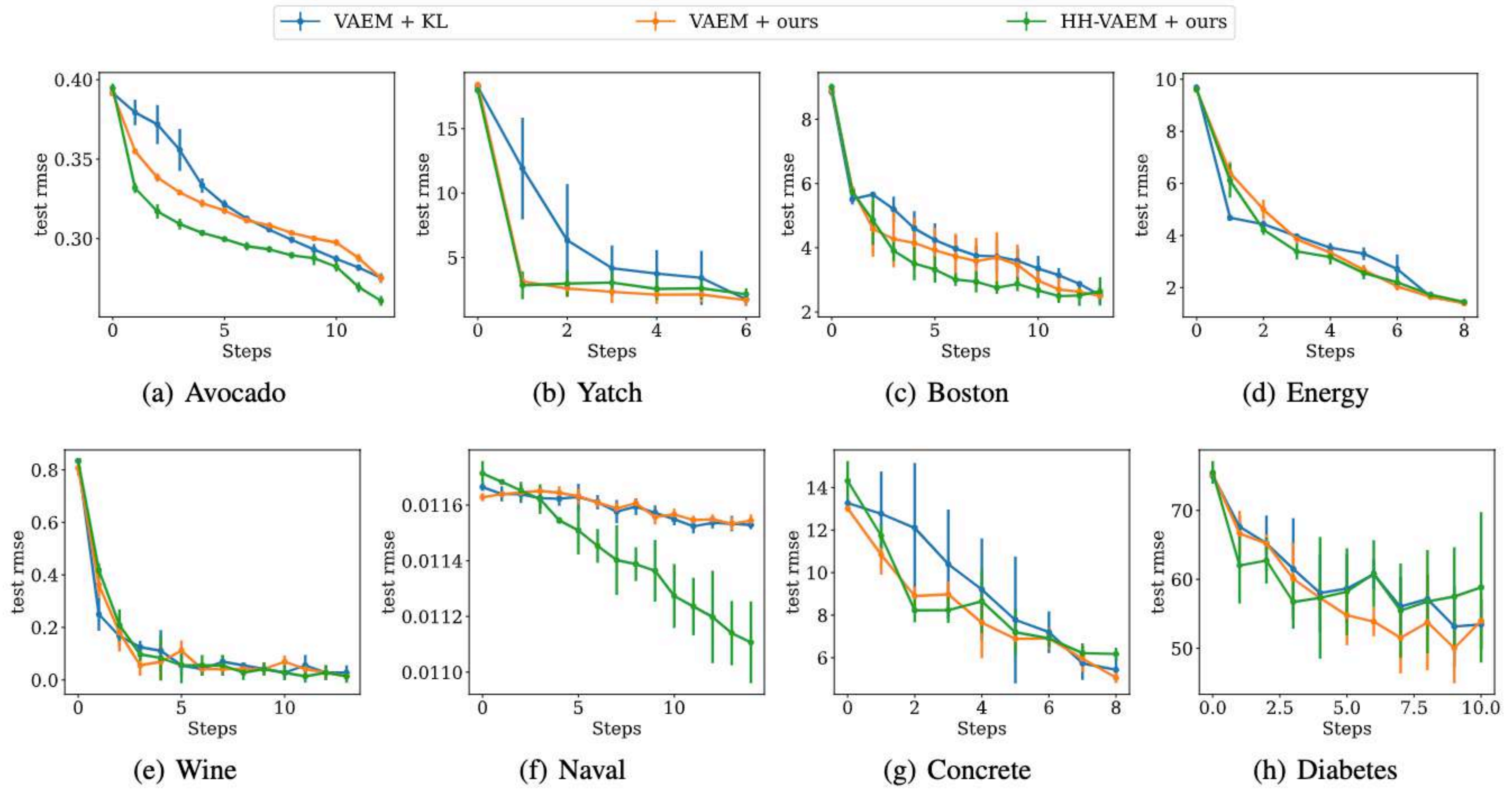
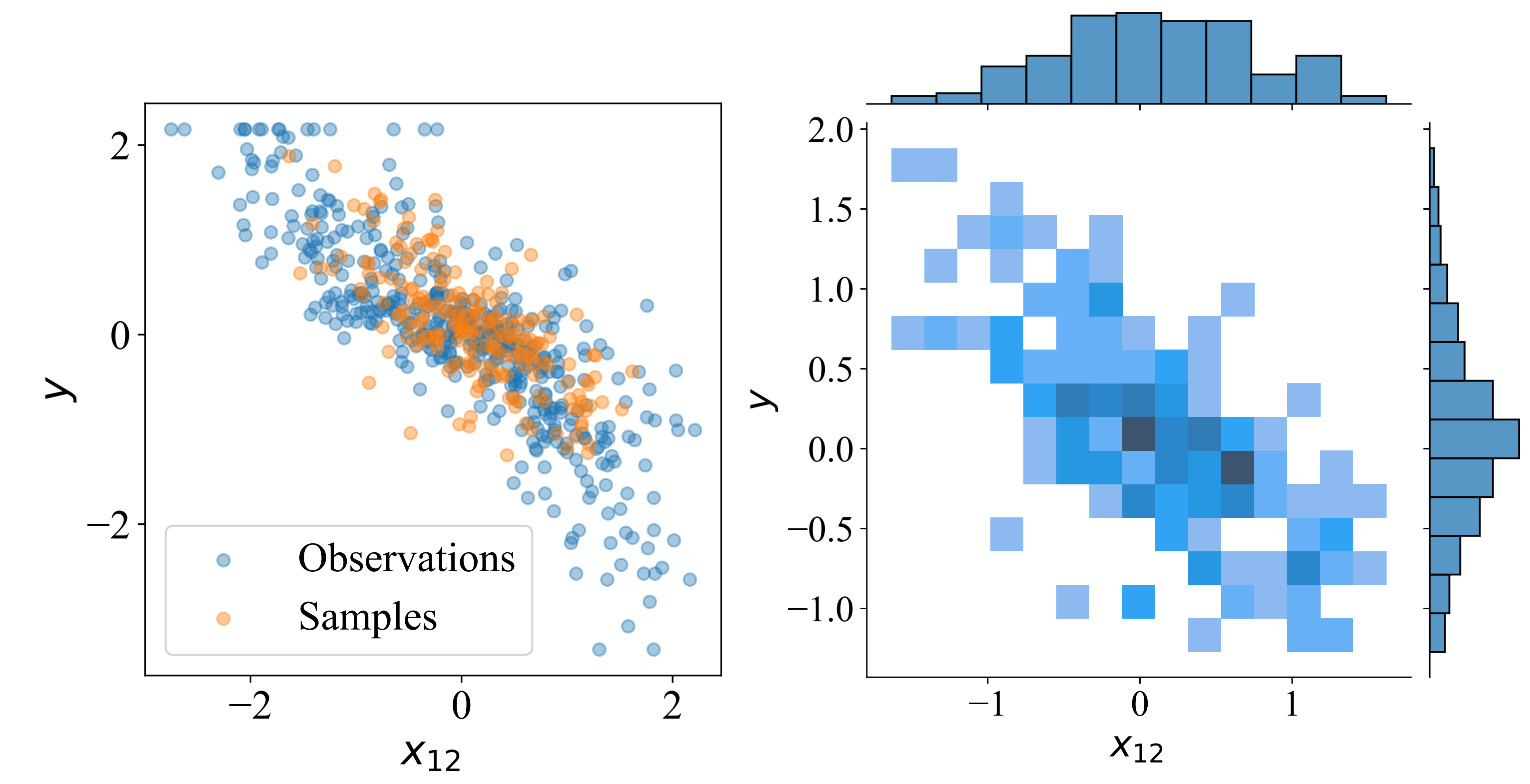
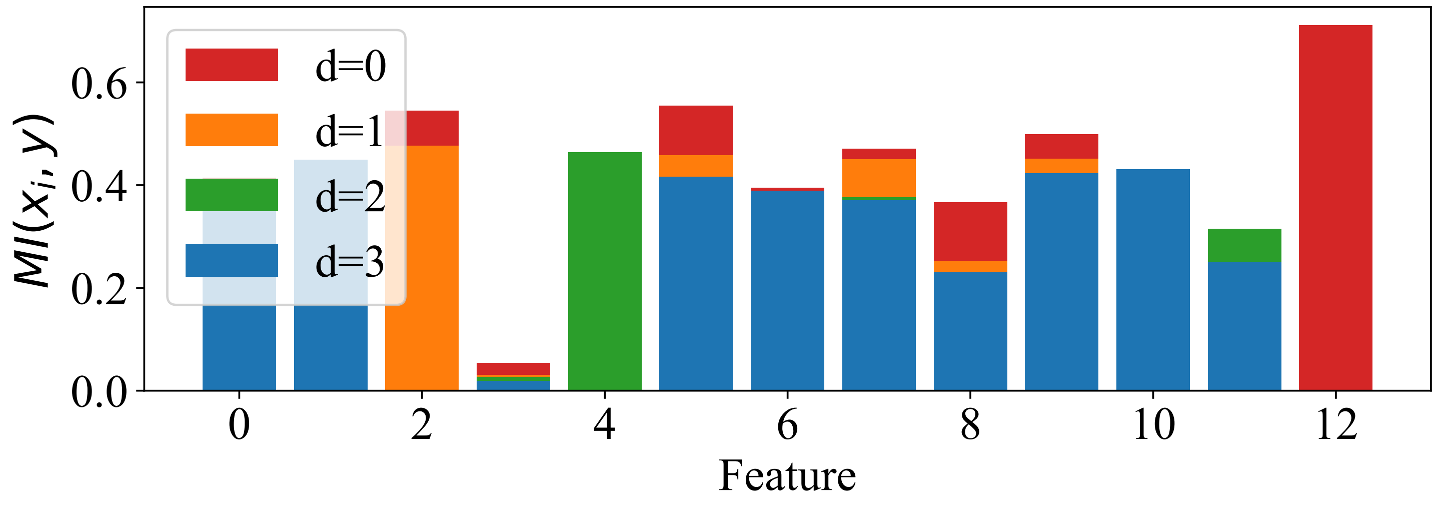


Figure 5: SAIA metric curves. Horizontal axis shows acquisition steps (number of discovered features). Vertical axis is the RMSE.





Conclusion

- We presented:

  1. **HH-VAEM**: novel Hierarchical VAE improved with HMC with automatic hyperparameter optimization.

  2. Novel sampling-based technique based on the Mutual Information estimation for efficient information acquisition.

- Based on the provided experiments, we demonstrate that our methods:

  ✓ Improve approximate inference in hierarchical VAEs wrt to the Gaussian approximation.

  ✓ Improve missing data imputation task.

  ✓ Improve prediction task.

  ✓ Improve active learning task.

References

- Peis, I., Ma, C., & Hernández-Lobato, J. M. (2022). Missing Data Imputation and Acquisition with Deep Hierarchical Models and Hamiltonian Monte Carlo. arXiv preprint arXiv:2202.04599.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Nazabal, A., Olmos, P. M., Ghahramani, Z., & Valera, I. (2020). Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, *107*, 107501.
- Ma, C., Tschiatschek, S., Turner, R., Hernández-Lobato, J. M., & Zhang, C. (2020). VAEM: a deep generative model for heterogeneous mixed type data. *Advances in Neural Information Processing Systems*, *33*, 11237-11247.
- Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., & Zhang, C. (2018). Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*.
- Vahdat, A., & Kautz, J. (2020). NVAE: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, *33*, 19667-19679.
- Maaløe, L., Fraccaro, M., Liévin, V., & Winther, O. (2019). Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, *32*.

References

- Child, R. (2020). Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*.
- Tomczak, J., & Welling, M. (2018, March). VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics* (pp. 1214-1223). PMLR.
- Ruiz, F. J., Titsias, M. K., Cemgil, T., & Doucet, A. (2021, December). Unbiased gradient estimation for variational auto-encoders using coupled Markov chains. In *Uncertainty in Artificial Intelligence* (pp. 707-717). PMLR.
- Cremer, C., Li, X., & Duvenaud, D. (2018, July). Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning* (pp. 1078-1086). PMLR.
- Mattei, P. A., & Frellsen, J. (2019, May). MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning* (pp. 4413-4423). PMLR.
- Bernardo, J. M. (1979). Expected information as expected utility. *the Annals of Statistics*, 686-690.
- Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, *2*(11), 2.

References

- Betancourt, M., & Girolami, M. (2015). Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, *79*(30), 2-4.
- Campbell, A., Chen, W., Stimper, V., Hernandez-Lobato, J. M., & Zhang, Y. (2021, July). A gradient based strategy for hamiltonian monte carlo hyperparameter optimization. In *International Conference on Machine Learning* (pp. 1238-1248). PMLR.
- Gong, W., Li, Y., & Hernández-Lobato, J. M. (2020). Sliced kernelized Stein discrepancy. *arXiv preprint arXiv:2006.16531*.
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical review E*, *69*(6), 066138.