Information Acquisition and Distributions of Functions with Deep Generative Models

Ignacio Peis
Universidad Carlos III de Madrid
ipeis@tsc.uc3m.es

2 June 2023

Pioneer Centre for Artificial Intelligence, Copenhagen, Denmark

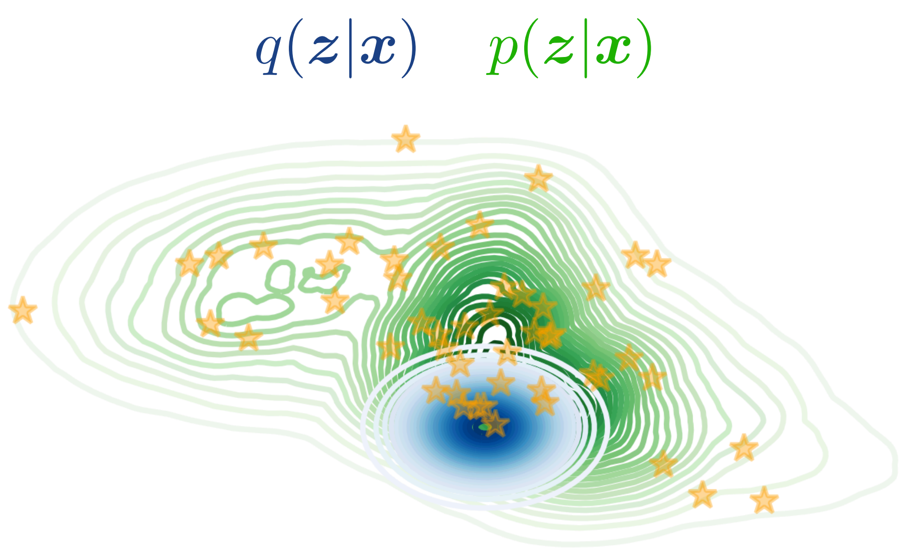
Introduction

Research Questions: Part I

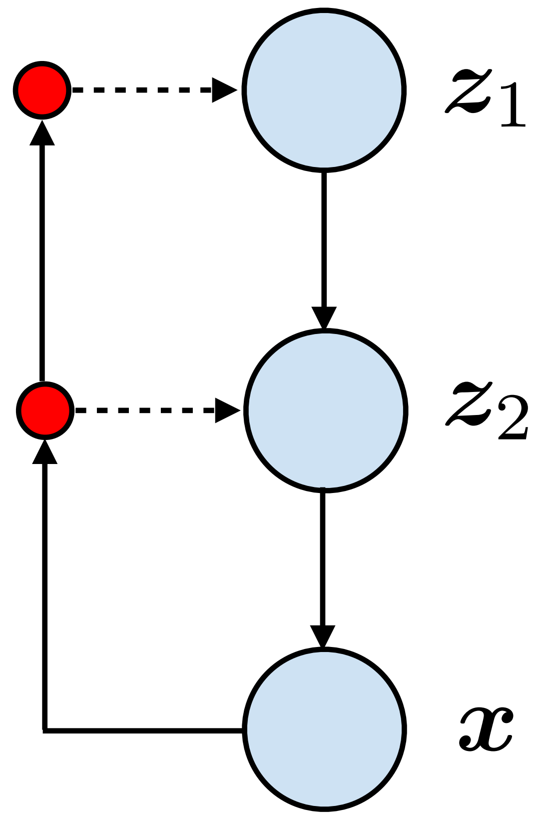
- One-layered VAEs approximate inference can be improved via Markov Chain Monte Carlo [1-4].

1. Could we leverage MCMC methods for Hierarchical VAEs?

2. If so, could we improve incomplete data handling with MCMC?



HMC samples (orange)



[1] Campbell et at., 2019 [2] Caterini et at., 2018 [3] Salimans et at., 2018 [4] Ruiz et at., 2021

Introduction

Research Questions: Part II

- VAEs are successful in generating structured data.

  1. Could we generate non-structured data via **functions** [5,6] using a VAE framework?

  2. Can we encode weights of a Neural Network?

[5] Dupont et at., 2022   [6] Dupont et at., 2022

Variational Autoencoders

Definition[7]

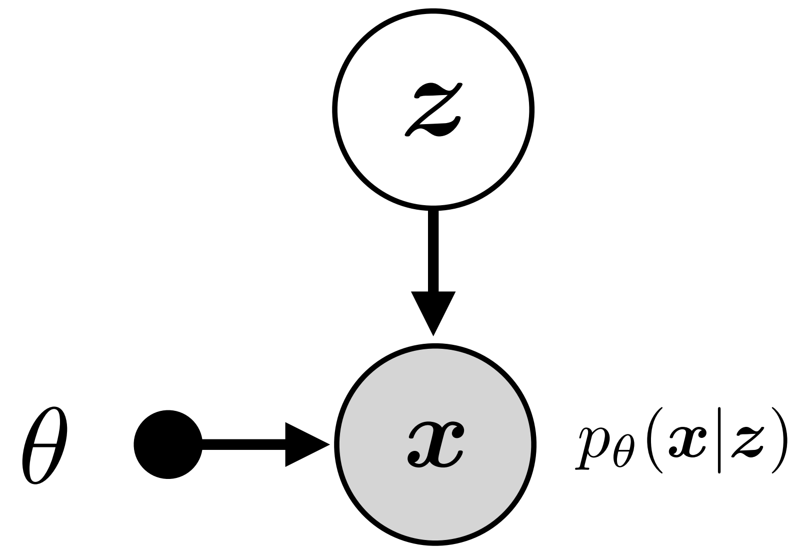
- Generative, explicit density models with intractable marginal log-likelihood.

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

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

- Posterior distribution:

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



[7] Kigma et at., 2013

Variational Autoencoders

Amortised Variational Inference (I)

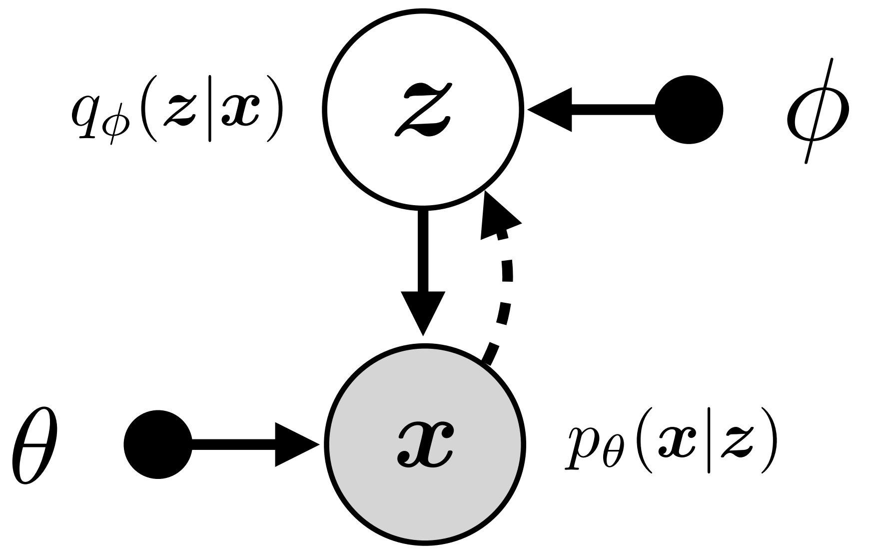
- Learn a **Gaussian** approximation of the posterior using observed data by minimizing

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

- Which is equivalent to maximizing

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

named **Evidence Lower Bound (ELBO).**

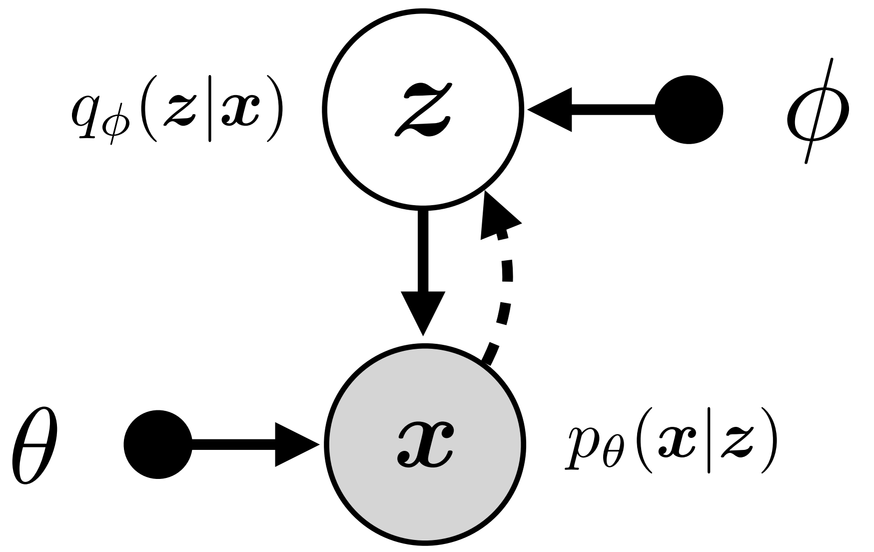


Variational Autoencoders

Amortised Variational Inference (I)

The **ELBO** is typically expressed as

$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log \underbrace{p_\theta(\boldsymbol{x}|\boldsymbol{z})}_{\textbf{Decoder}} - D_{KL}\left(\underbrace{q_\phi(\boldsymbol{z}|\boldsymbol{x})}_{\textbf{Encoder}} || p(\boldsymbol{z})\right) \longleftarrow \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$



Variational Autoencoders

Amortised Variational Inference (II)

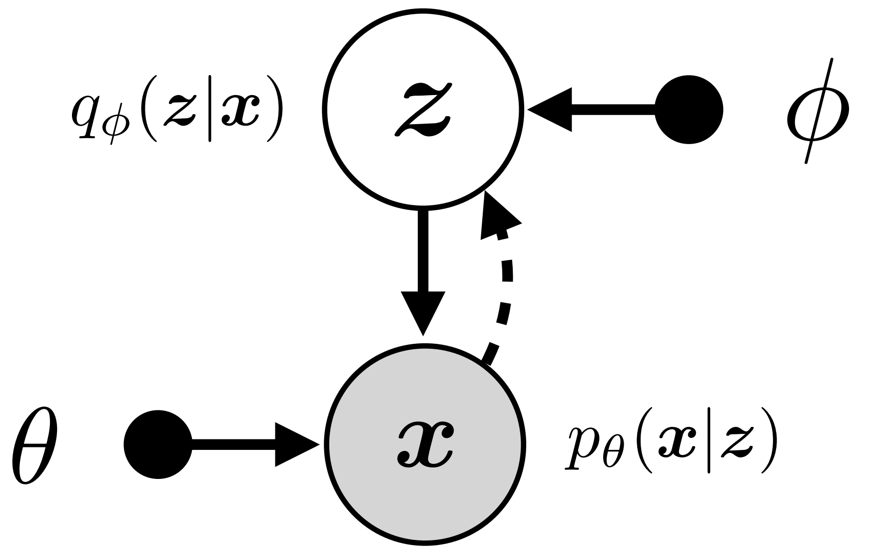
- First **reconstruction** term requires an MC estimator:

$$\hat{\mathcal{L}}(\boldsymbol{x}) = \frac{1}{S}\sum_{i=1}^{S}\left(\log p_\theta(\boldsymbol{x}|\boldsymbol{z}^{(s)})\right) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right)$$

Reparameterization trick[7]

$$\boxed{\begin{gathered}\boldsymbol{z}^{(s)} = f_{\boldsymbol{\mu}}(\boldsymbol{x}) + f_{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot \epsilon^{(s)} \\ \epsilon^{(s)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\end{gathered}}$$

Works reasonably good even for S=1



- Second **regularization** term can be computed in close form.

[7] Kigma et at., 2013

Variational Autoencoders

Training

For each batch of B samples:

1. Encode to the parameters of the approximate posteriors $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)$.

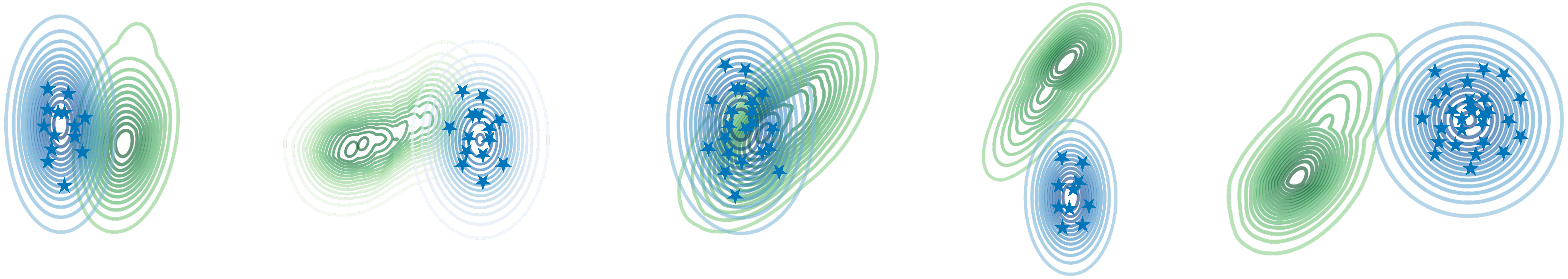
2. Draw a sample from each $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)$.

3. Optimization step on $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

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

Variational Autoencoders

Approximate Inference



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

- Can we get better samples that follow the green contour? ✓

[8]

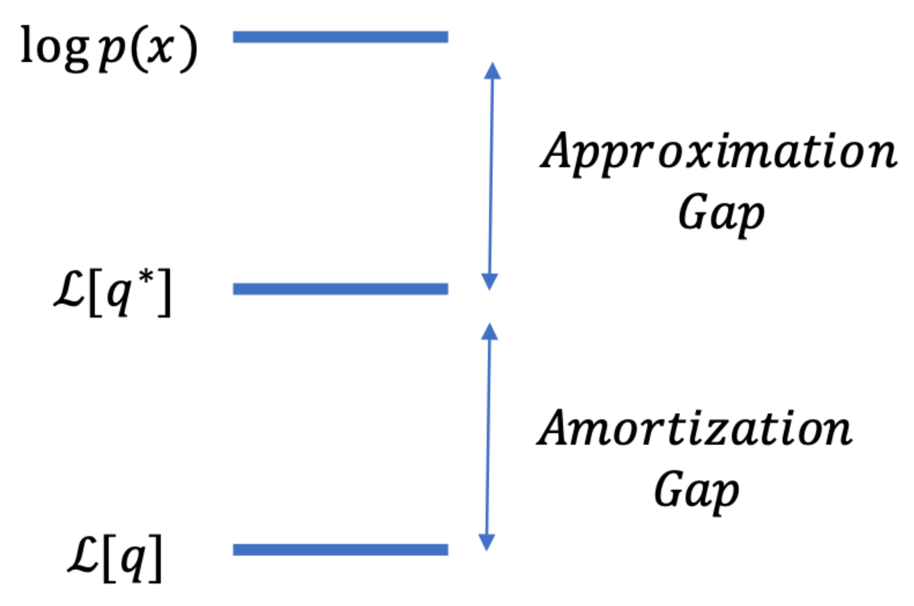


Figure 1. Gaps in Inference

[8] Cremer et at., 2018

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

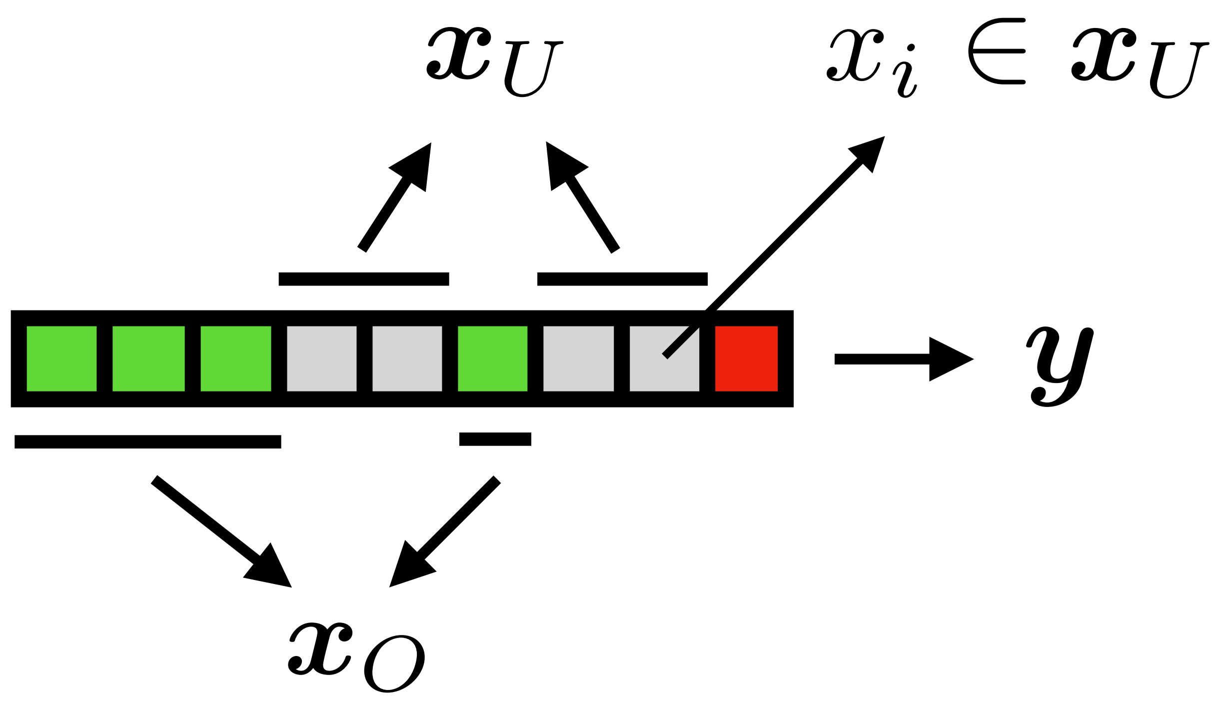
Challenges

Enhance information acquisition with VAEs

- Discovery of high-value information.

- Bayesian reward function [9] as an expected gain of information:

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



[9] Bernardo et at., 1979

Challenges
Enhance information acquisition with VAEs

- Approximated in [10,11] by transforming the reward into the latent space:

$$\hat{R}(i, \boldsymbol{x}_o) = \mathbb{E}_{\hat{p}(\boldsymbol{x}_i|\boldsymbol{x}_o)} D_{KL}\left[q\left(\boldsymbol{z}|\boldsymbol{x}_i, \boldsymbol{x}_o\right) || \, q\left(\boldsymbol{z}|\boldsymbol{x}_o\right)\right] - \mathbb{E}_{\hat{p}(\boldsymbol{y}, \boldsymbol{x}_i|\boldsymbol{x}_o)} D_{KL}\left[q\left(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{x}_i, \boldsymbol{x}_o\right) || \, q\left(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{x}_o\right)\right]$$

- These methods are based on **Gaussian** approximations of the true posterior.

[10] Ma et at., 2018 [11] Ma et at., 2020

Challenges

Improve missing data imputation with VAEs

- Imputation under a VAE framework [10-13]:

$$p(\boldsymbol{x}_U|\boldsymbol{x}_O) = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x}_O)}[p(\boldsymbol{x}_U|\boldsymbol{z})] \approx \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_O)}[p(\boldsymbol{x}_U|\boldsymbol{z})]$$

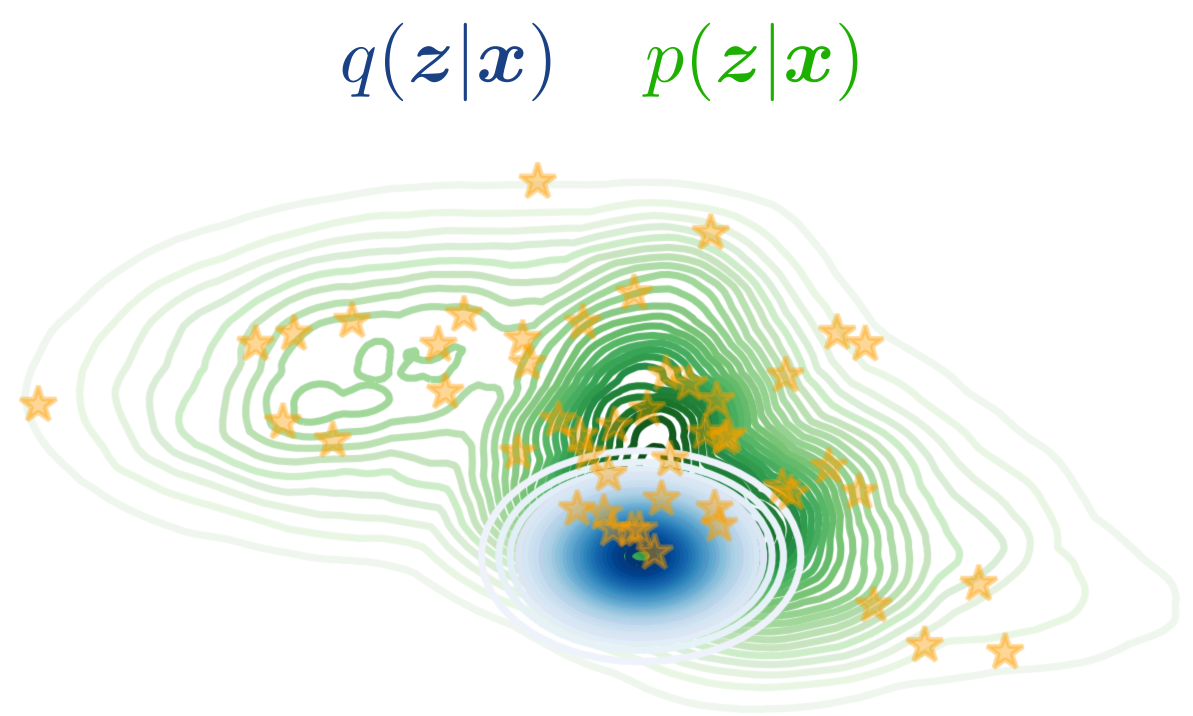
- Also based on **Gaussian** approximations of the true posterior.

[10] Ma et at., 2018 [11] Ma et at., 2020 [12] Nazabal et at., 2020 [13] Mattei et at., 2020

Challenges

Sampling-based methods for incomplete data related tasks

Expectations over the intractable posterior should leverage a well-designed MCMC approximation method when compared to a Gaussian-based approximation.



HMC samples (orange)

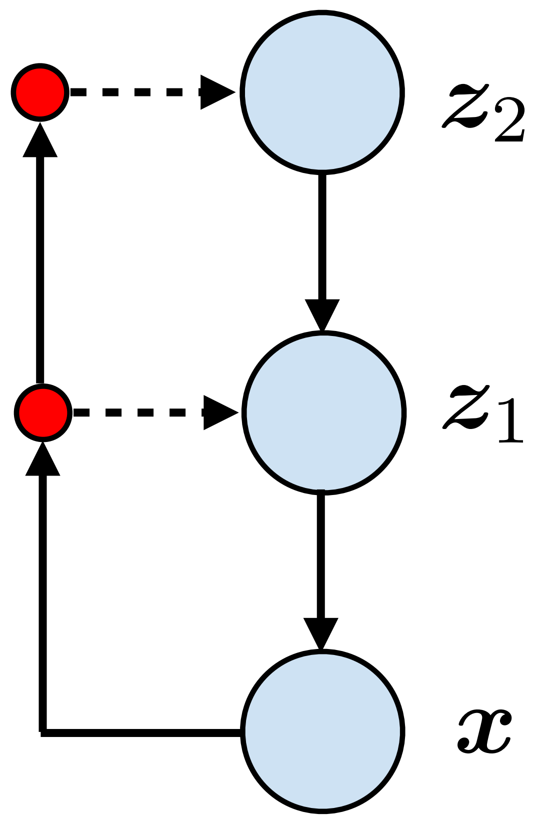
- ☐ **Imputation**: $p(\boldsymbol{x}_U|\boldsymbol{x}_O) \approx \mathbb{E}_{q^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{x}_O)}[p(\boldsymbol{x}_U|\boldsymbol{\epsilon})]$.
- ☐ **Prediction**: $p(\boldsymbol{y}|\boldsymbol{x}_O) \approx \mathbb{E}_{q^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{x}_O)}[p(\boldsymbol{y}|\boldsymbol{\epsilon}, \boldsymbol{x}_O, \hat{\boldsymbol{x}}_U)]$.
- ☐ **Sampling-based active learning**.

Challenges

Hierarchical VAEs

- Hierarchical VAEs are successful at increasing flexibility.

$$p(\boldsymbol{z}_L)\prod_{l=1}^{L-1} p(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})$$



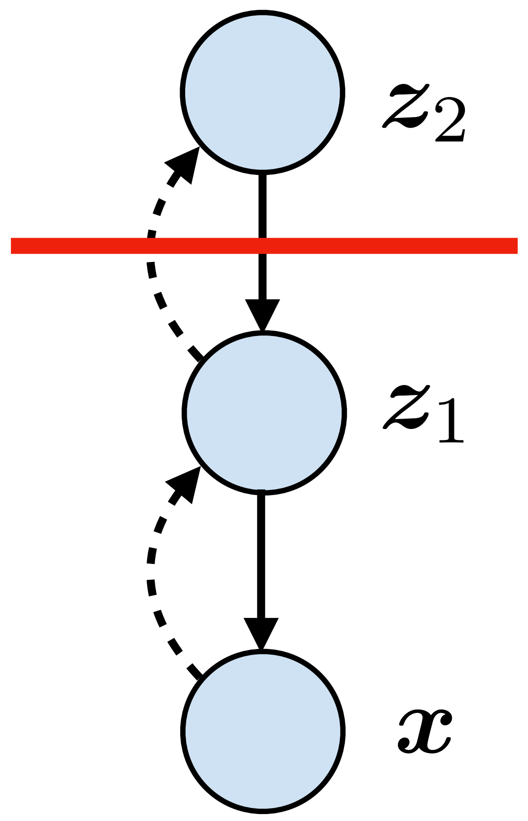
- The hierarchy allows for modelling:
  - Abstract to specific generative factors.
  - Global to local generative factors.

[14]

[14] Child, 2020

Challenges
Hierarchical VAEs

- **Delicate inference** (posterior collapse).



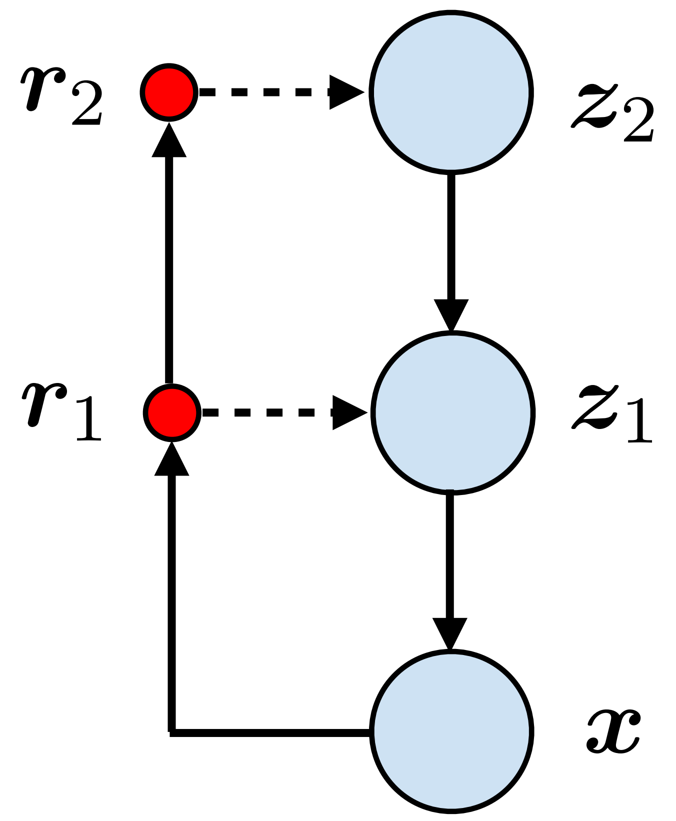
$$ELBO(\mathbf{x}) = \mathbb{E}_{Q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z}_1) - KL[q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1|\mathbf{z}_2)] - KL[q(\mathbf{z}_2|\mathbf{z}_1)||p(\mathbf{z}_2)]\right]$$

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

Challenges
Hierarchical VAEs

- **Solution**: top-down inference.

$$Q(\mathbf{z}_1, \mathbf{z}_2 | \mathbf{x}) = q(\mathbf{z}_1 | \mathbf{z}_2, \mathbf{x}) q(\mathbf{z}_2 | \mathbf{x}).$$

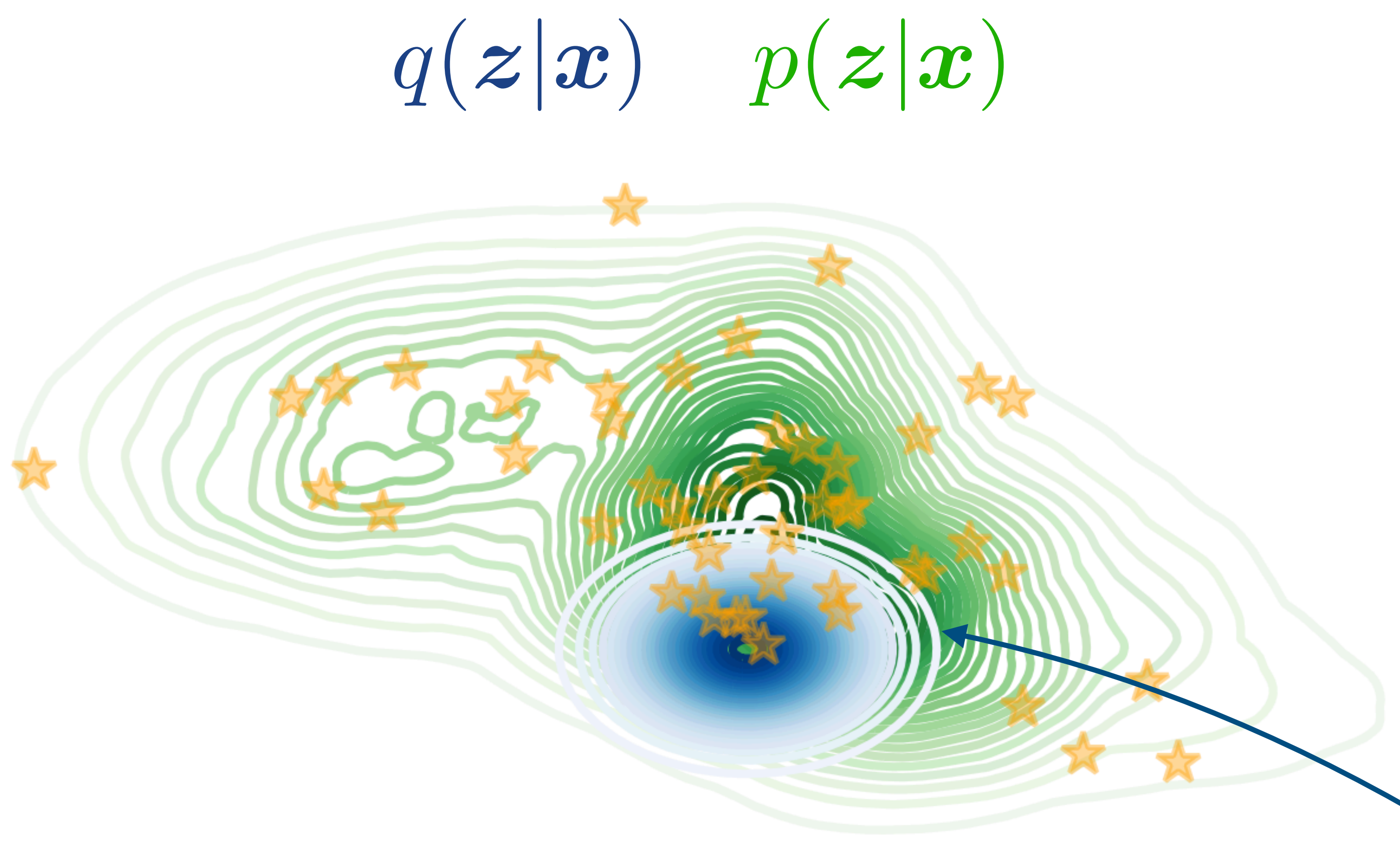


✓ Posterior collapse is relaxed.

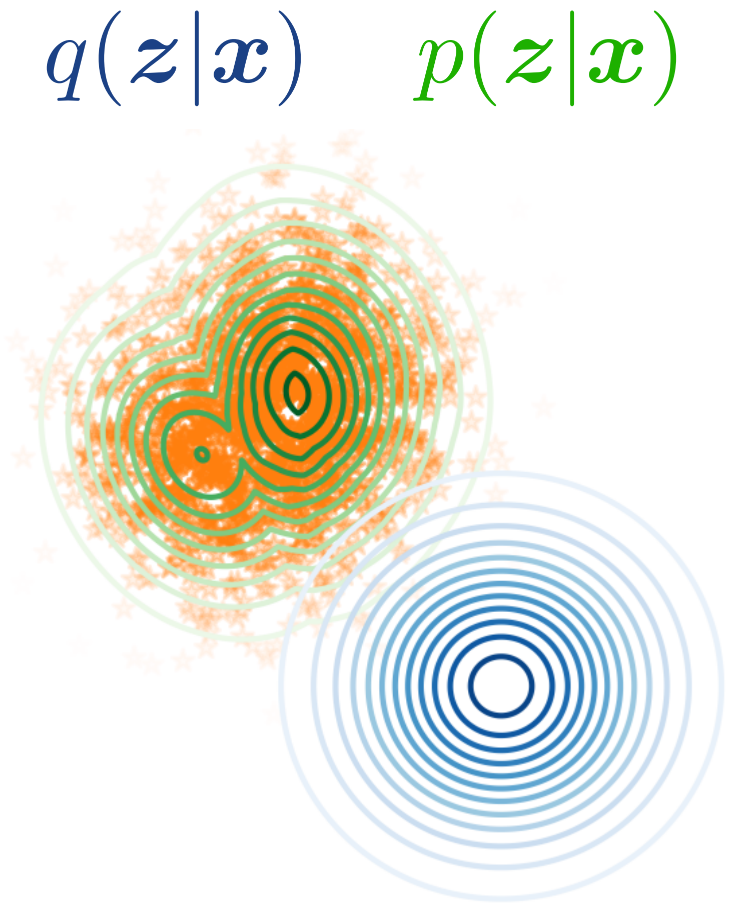
✦ Inference bias worsens with latent dimensionality.

Challenges

MCMC for improving inference in VAEs



[15]HMC samples (orange)



[15]HMC samples (orange)

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

$$\approx \frac{1}{S}\sum_{s=1}^{S} \log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}^{(s)}) - D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})\right)$$

[15] Peis et al., 2021

Challenges

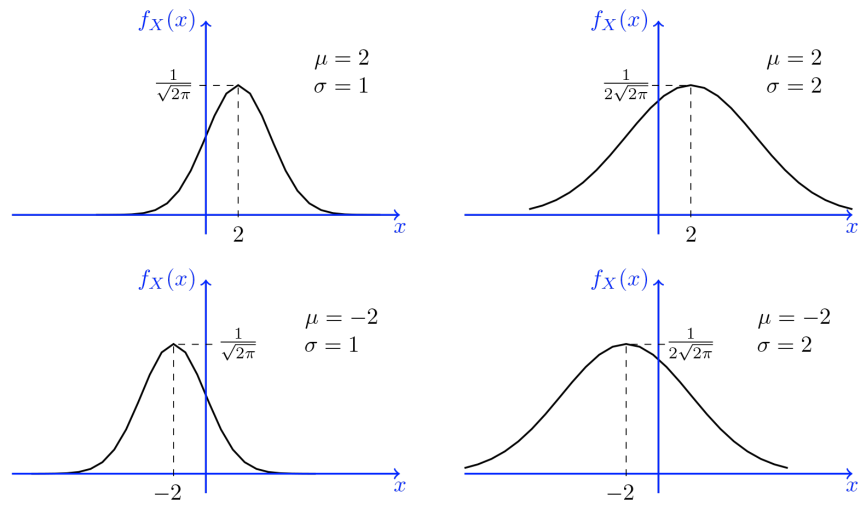
Handling partial heterogeneous data

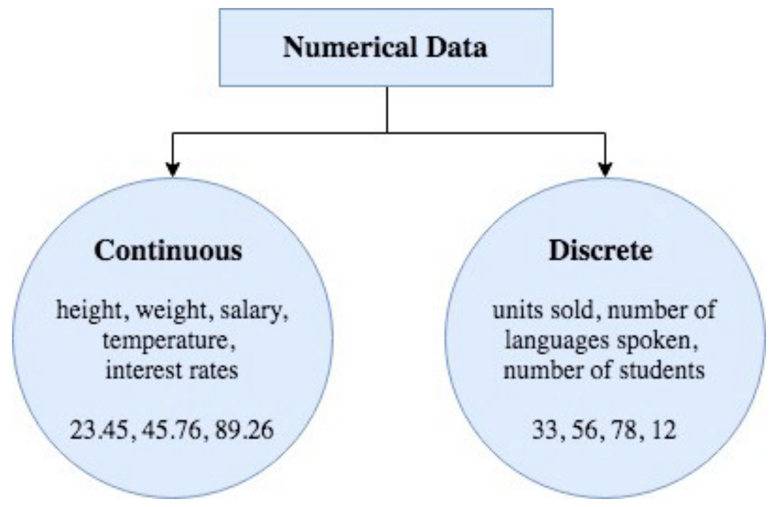
- Factorization over dimensions[10-13]:

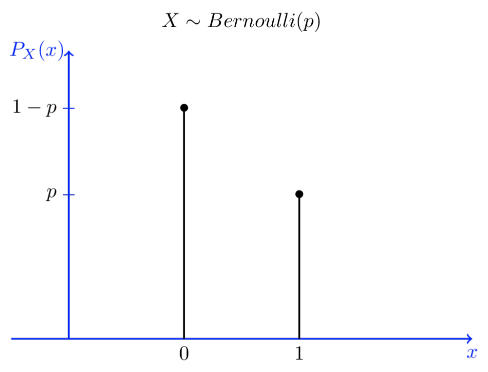
$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{d=1}^{D} \mathbb{I}(x_d \in \boldsymbol{x}_O) \log p_\theta(x_d|\boldsymbol{z})\right] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}_O)||p(\boldsymbol{z}))$$

- Naïve approach for heterogeneous[12,13]: use a different likelihood per dimension.

- **Problem**: inbalanced likelihoods.







[10] Ma et at., 2018 [11] Ma et at., 2020 [12] Nazabal et at., 2020 [13] Mattei et at., 2020

Challenges

Handling partial heterogeneous data

- **Solution**[11]: 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 a joint dependency VAE (θ, ϕ) on the marginally encoded data:

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

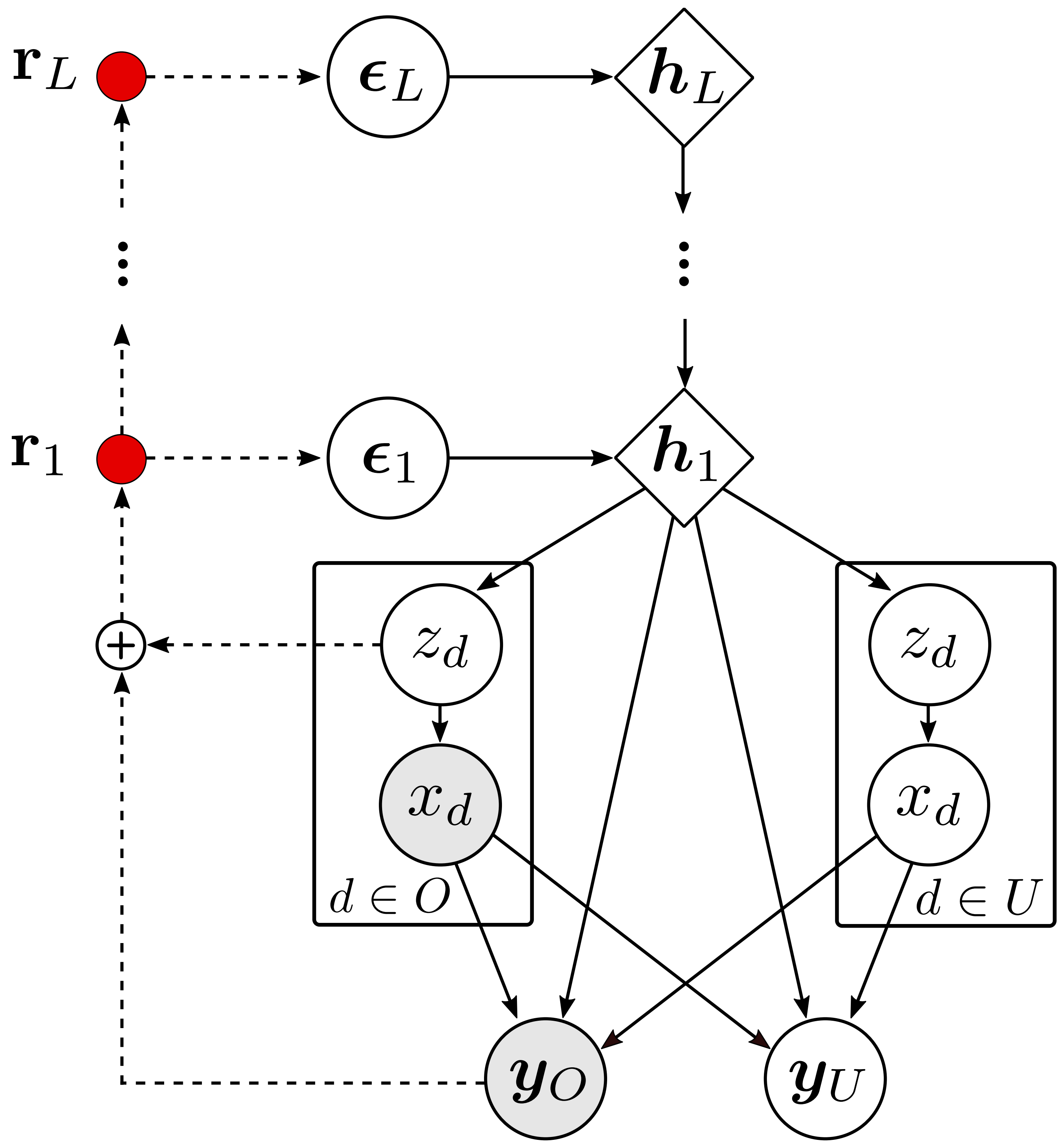
- Interdependencies between heterogenous variables are better captured by the dependency VAE.

[11] Ma et at., 2020

Contributions

Hierarchical Hamiltonian VAE for mixed-type incomplete data (HH-VAEM)

- Increased flexibility by using hierarchical latent space.
- Improved inference by means of automatically tuned HMC.
- Reparameterization for well-posed HMC on relaxed posterior.
- Heterogeneous data handling.
- More accurate imputation and prediction.
- More effective information acquisition.



Contributions (II)

Sampling-based method for active information acquisition

- **Sampling**-based estimator [16] of the Mutual Information:

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

$$= \iint_{x_i,\boldsymbol{y}} p_{x_i,\boldsymbol{y}|\boldsymbol{x}_O}(x_i, \boldsymbol{y}|\boldsymbol{x}_O) \log\left(\frac{p_{x_i,\boldsymbol{y}|\boldsymbol{x}_O}(x_i, \boldsymbol{y}|\boldsymbol{x}_O)}{p_{x_i|\boldsymbol{x}_O}(x_i|\boldsymbol{x}_O)p_{\boldsymbol{y}|\boldsymbol{x}_O}(\boldsymbol{y}|\boldsymbol{x}_O)}\right)$$

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

✓ Avoids the Gaussian approximation

✓ Efficient, easy parallelization.

[16] Kraskov et at., 2004

Method

Hamiltonian Monte Carlo

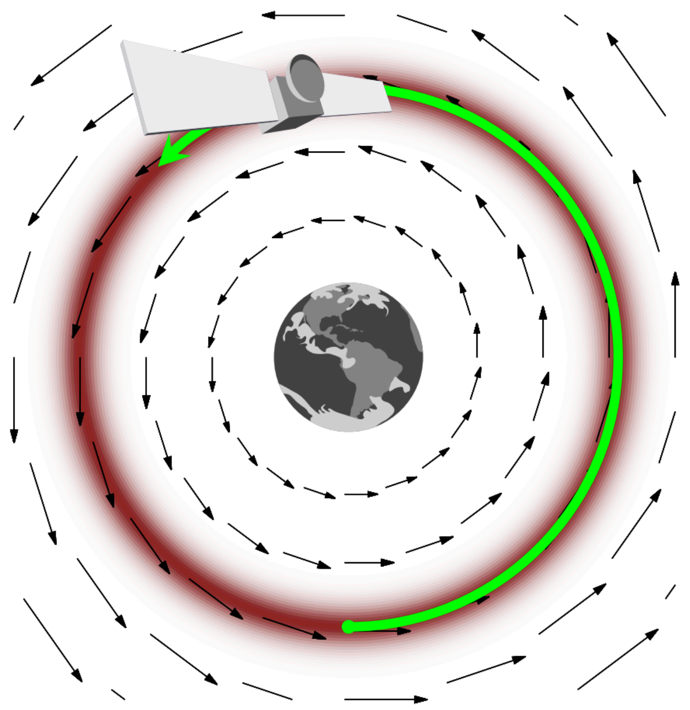
- Sample from complex distributions via unnormalised targets

1. Phase and momentum space

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

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

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

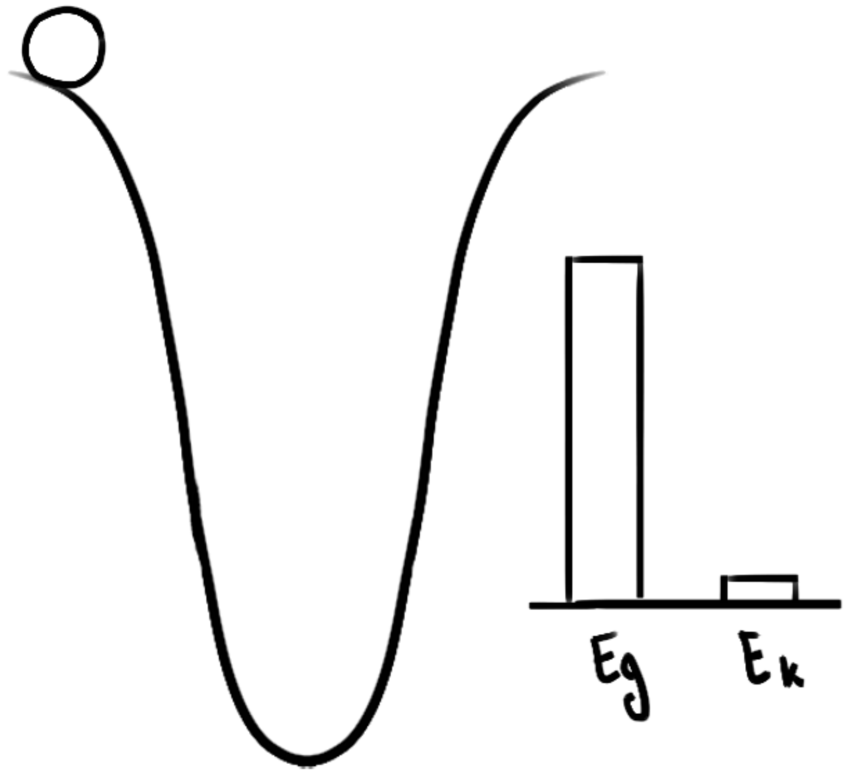


2. Hamiltonian equations

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

$$\Rightarrow$$

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



Method

Hamiltonian Monte Carlo

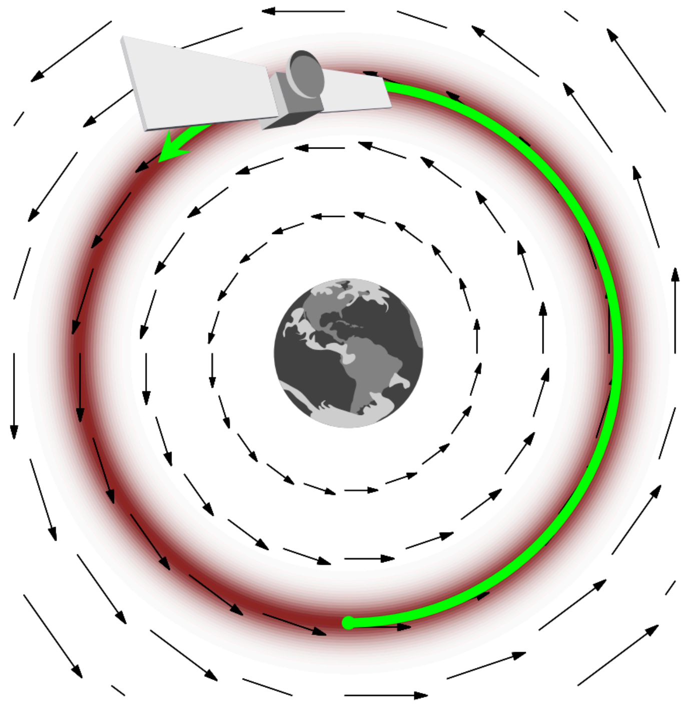
- Sample from complex distributions via unnormalised targets

1. Phase and momentum space

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

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

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

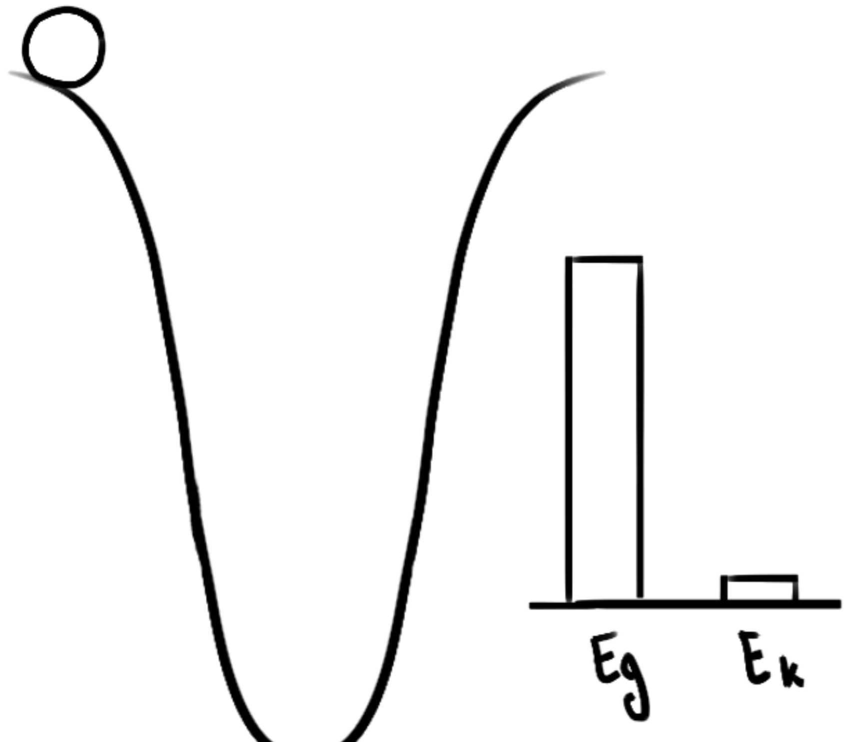


2. Hamiltonian equations

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

$\Rightarrow$

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



Method

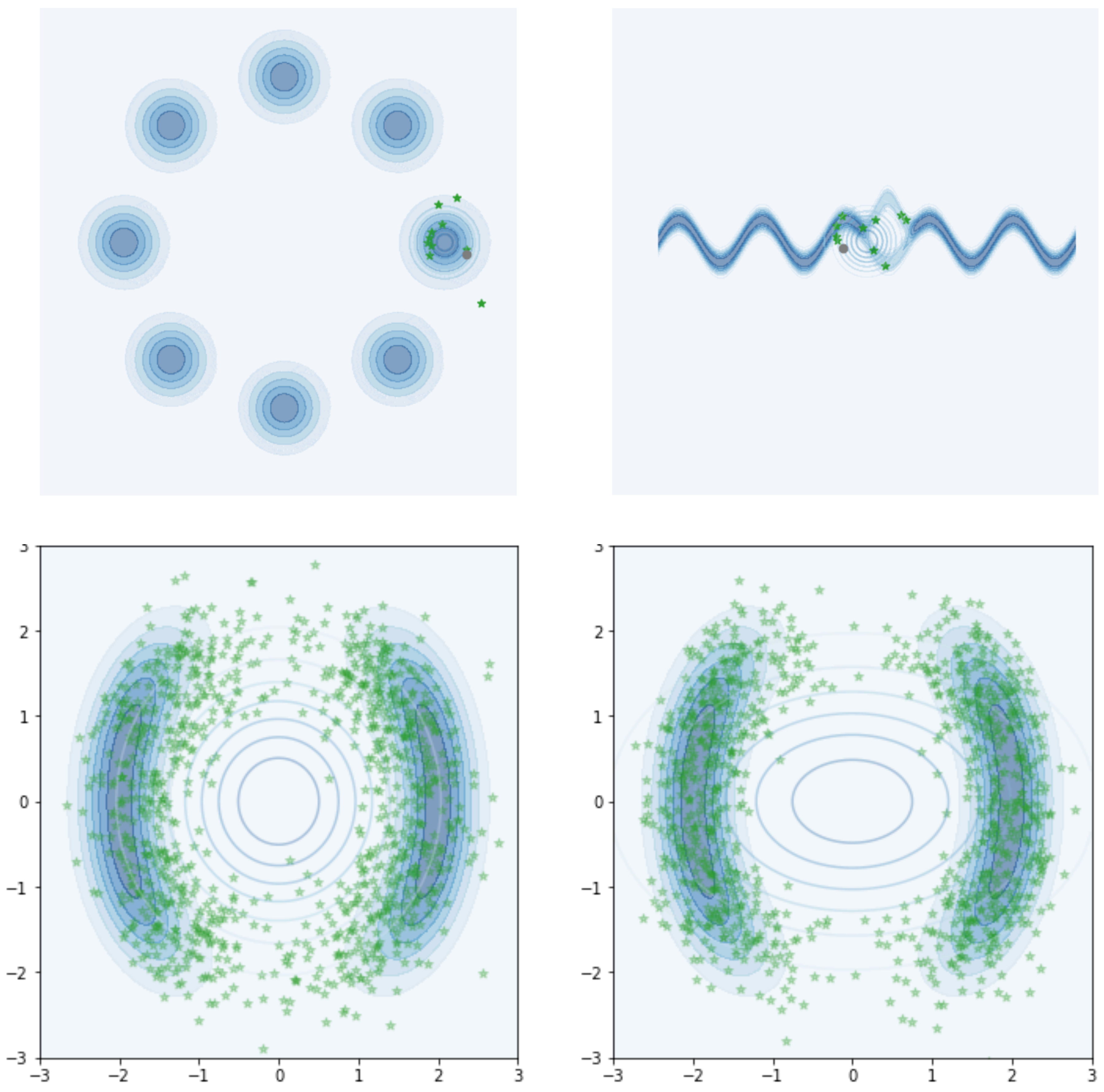
Hamiltonian Monte Carlo

- Discrete trajectories (*chains*) of T updates, ending in:

$$q^{(T)}(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z}|\boldsymbol{x})$$

- Target: true posterior density.

- Needed:
  - 1. Good initial proposal (encoder).
  - 2. Well-defined hyperparameters.



Random hyperparameters

Tuned hyperparameters

Method

HMC Hyperparameter tuning [1]

- Tuning the hyperparameters via Variational Inference:

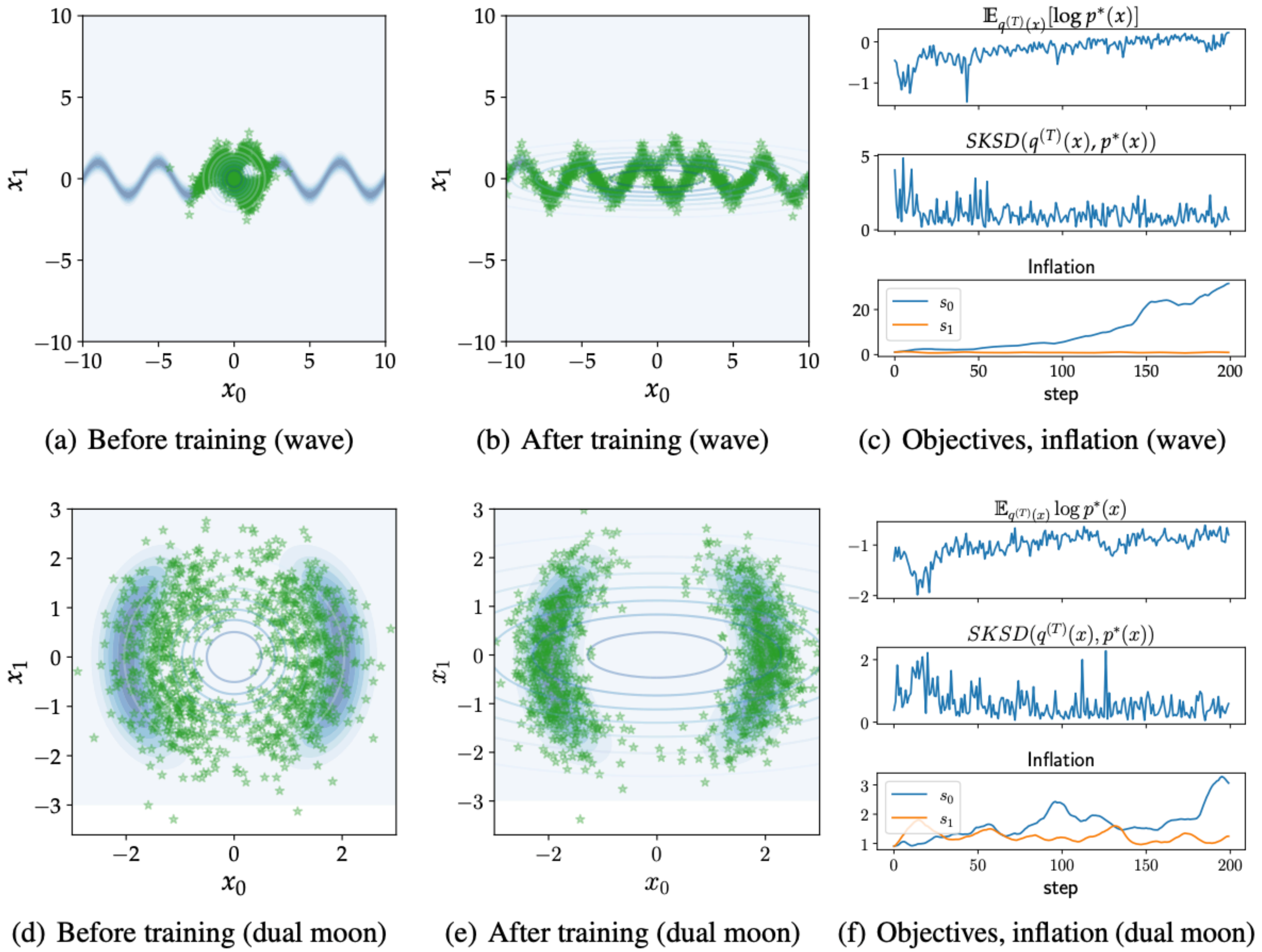
$$\phi^* = \underset{\phi}{\operatorname{argmax}} \ \mathbb{E}_{q_\phi^{(T)}(\boldsymbol{z})}\left[\log p^*(\boldsymbol{z})\right] + H\left[q_\phi^{(T)}(\boldsymbol{z})\right]$$

- Add an **inflation** parameter for scaling the proposal [17]

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

- Code available at:

https://github.com/ipeis/HMCTuning



(a) Before training (wave) (b) After training (wave) (c) Objectives, inflation (wave)

(d) Before training (dual moon) (e) After training (dual moon) (f) Objectives, inflation (dual moon)

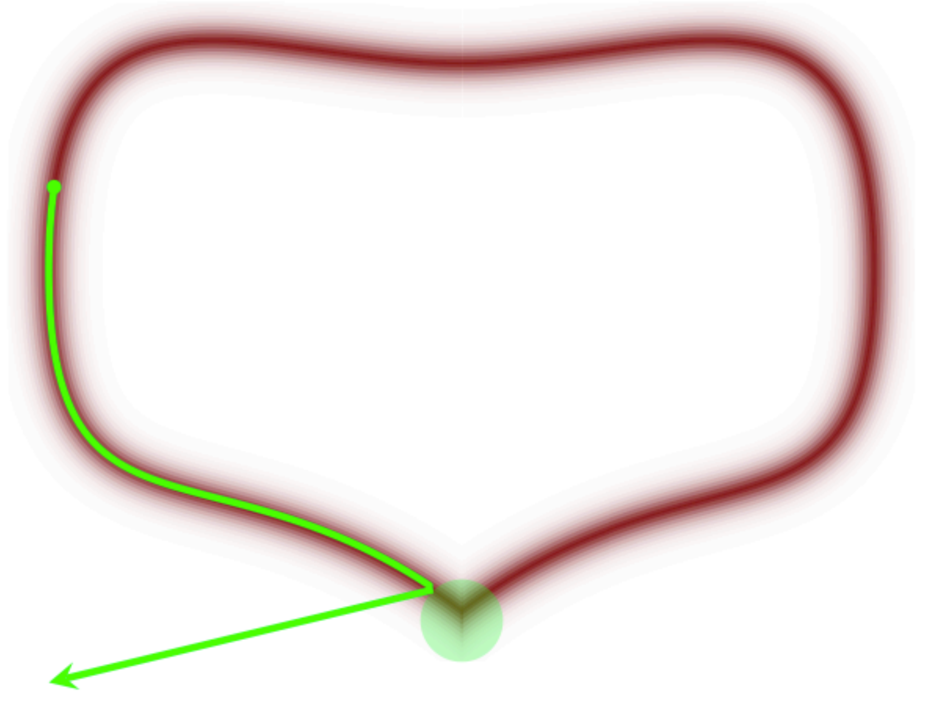
[1] Campbell et at., 2021 [17] Gong et at., 2020

Method

HMC is ill-posed for Hierarchical VAEs

- Hierarchical dependencies lead to **huge gradients [18, 19]**

$$
\begin{aligned}
\boldsymbol{r}_{l+\frac{1}{2}} &= \boldsymbol{r}_l + \frac{1}{2}\boldsymbol{\phi} \odot \nabla_{z_l} \log p^*(\boldsymbol{z}_l)\,, \\
\boldsymbol{z}_{l+1} &= \boldsymbol{z}_k + \boldsymbol{r}_{l+\frac{1}{2}} \odot \boldsymbol{\phi} \odot \frac{1}{\boldsymbol{M}}\,, \\
\boldsymbol{r}_{l+1} &= \boldsymbol{r}_{l+\frac{1}{2}} + \frac{1}{2}\boldsymbol{\phi} \odot \nabla_{z_{l+1}} \log p^*(\boldsymbol{z}_{l+1}),
\end{aligned}
$$



- Samples can diverge due to integrator overflow issues.

[18] Betancourt et at., 2017 [19] Betancourt et at., 2015

Method

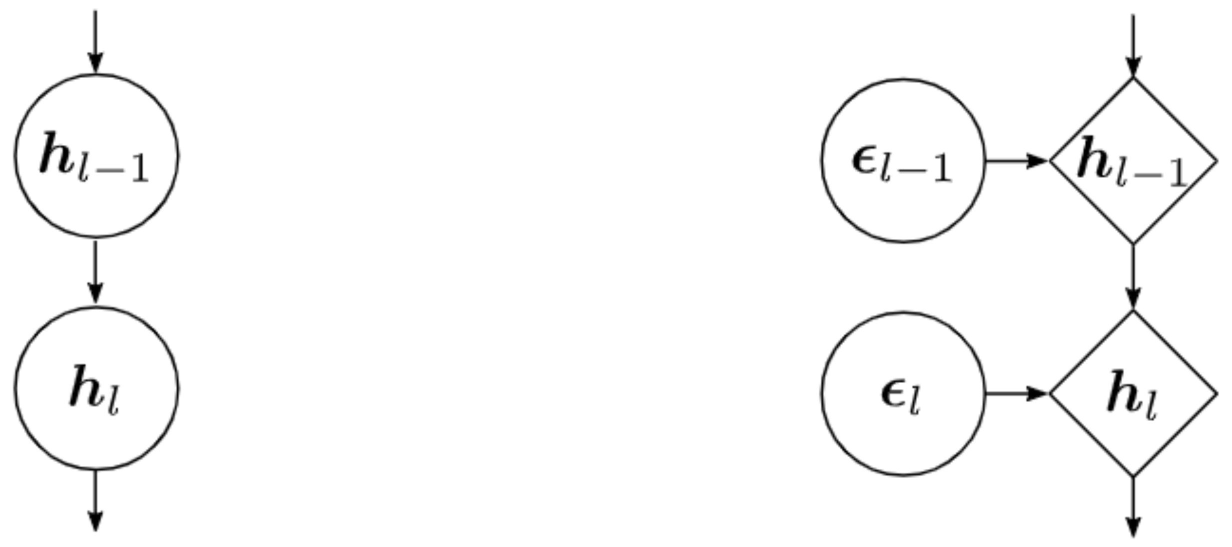
HMC is ill-posed for Hierarchical VAEs

- **Solution**:

  ✓ Reparameterization for 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 $\theta_{\mu_l} \rightarrow f_{\mu_l}, \theta_{\sigma_l} \rightarrow f_{\sigma_l}$



(a) AR hierarchy (b) Reparameterization

Method

Ill-posed for HMC

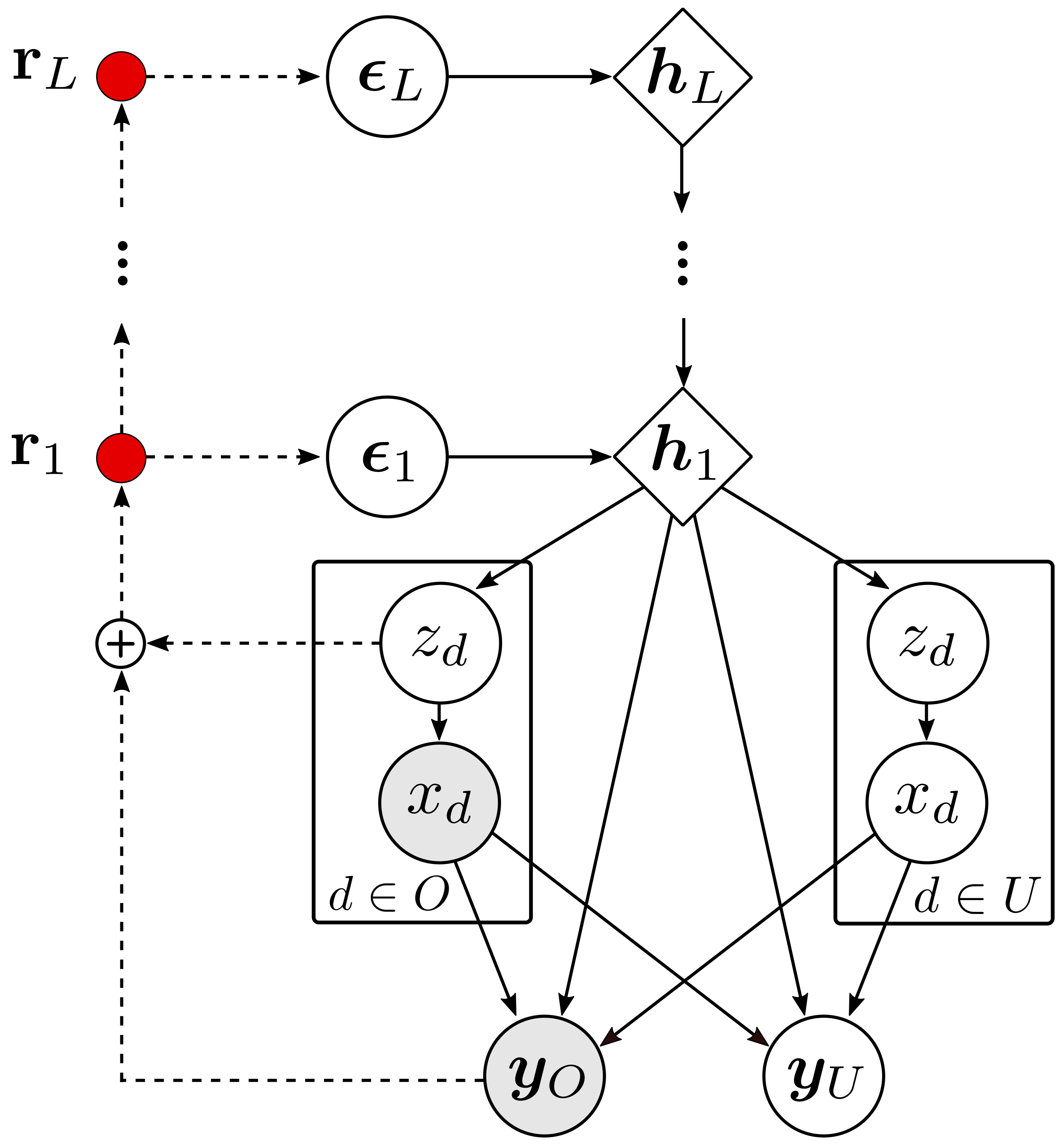
- **Solution**:

  ✓ Reparameterization for 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 $\theta_{\mu_l} \rightarrow f_{\mu_l}, \theta_{\sigma_l} \rightarrow f_{\sigma_l}$

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



Method

Ill-posed for HMC

- **Solution**:

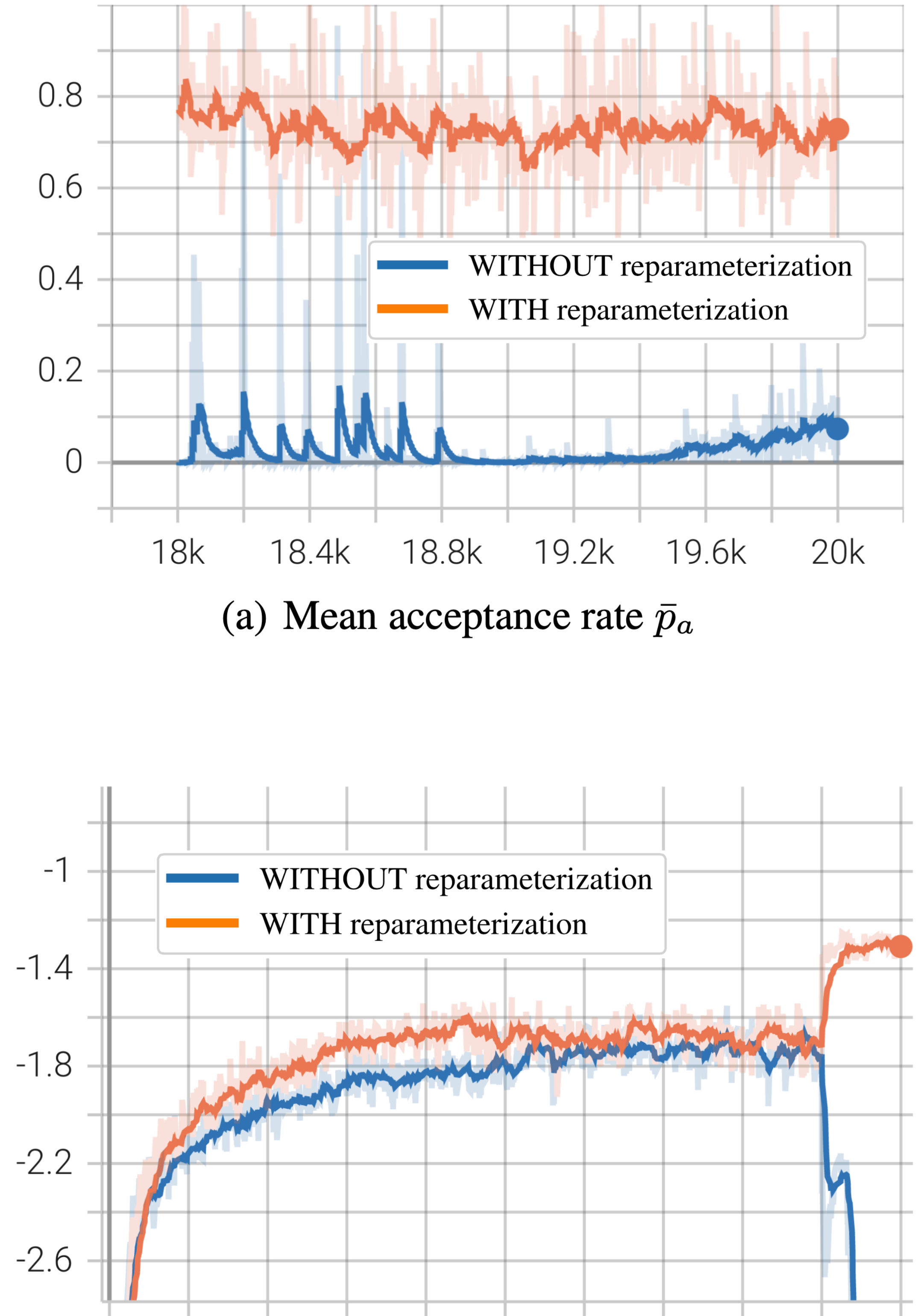
  ✓ Reparameterization for relaxed posterior:

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

  NNs with parameters $\theta_{\mu_l} \rightarrow f_{\mu_l}, \theta_{\sigma_l} \rightarrow f_{\sigma_l}$

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

  ✓ No need to increase complexity of the HMC method.



(a) Mean acceptance rate $\bar{p}_a$

(b) $\log p(\boldsymbol{x}_U|\boldsymbol{x}_O)$

Method

Optimization algorithm

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

- 1. Train marginal VAEs using:

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

Method

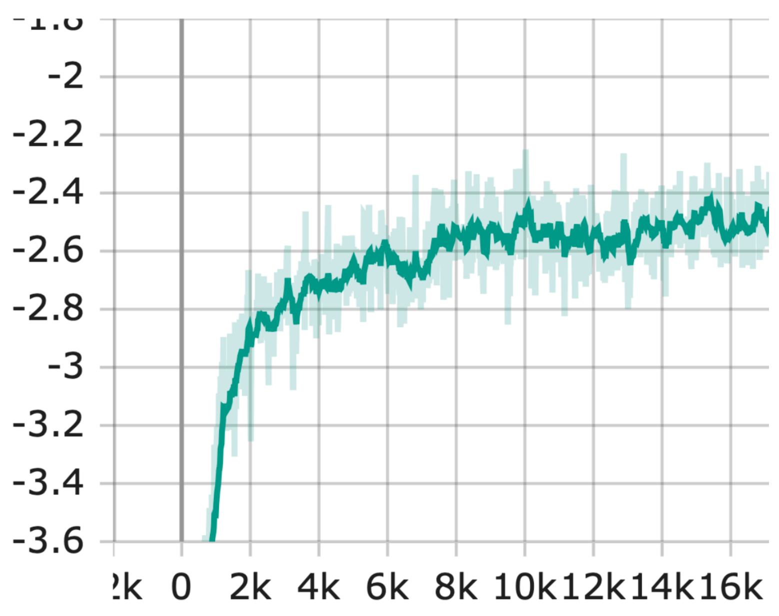
Optimization algorithm

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: γ, θ, ψ, ϕ, 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**
        $\gamma_d^{t+1}, \theta_d^{t+1} \leftarrow \mathrm{Adam}_{\gamma_d^t, \theta_d^t}(\mathcal{L}_d)$
    end for
end for
STAGE 2: DEPENDENCY VAE
for $t = 1$ **to** T_{VAE} **do**
    $\theta^{t+1}, \psi^{t+1} \leftarrow \mathrm{Adam}_{\theta^t, \psi^t}(\mathcal{L}_{VI})$
end for

- 2. Training Hierarchical VAE using the ELBO:

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



(a) $\log p(\boldsymbol{x}_U | \boldsymbol{x}_O)$

Method

Optimization algorithm

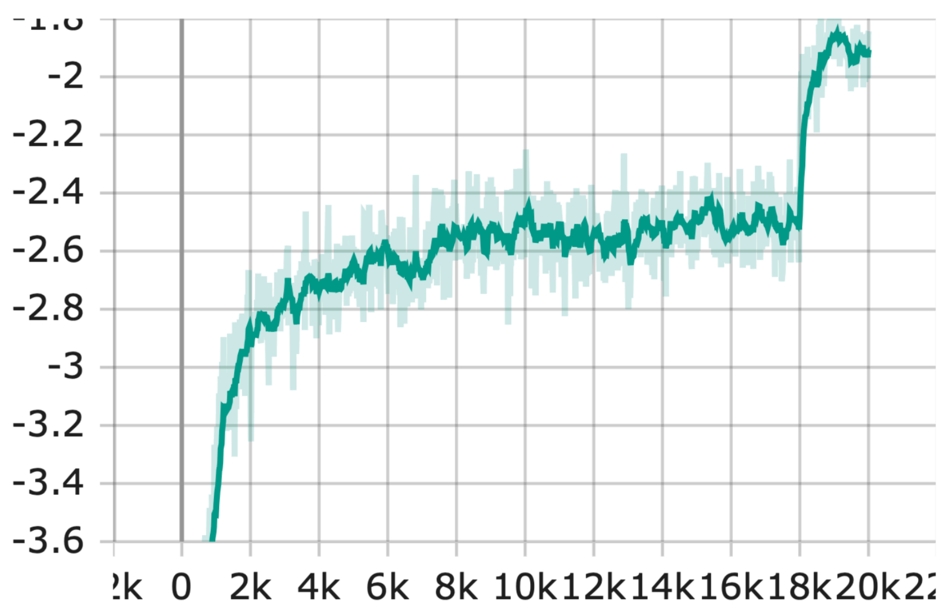
Algorithm 1 Training algorithm for HH-VAEM

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

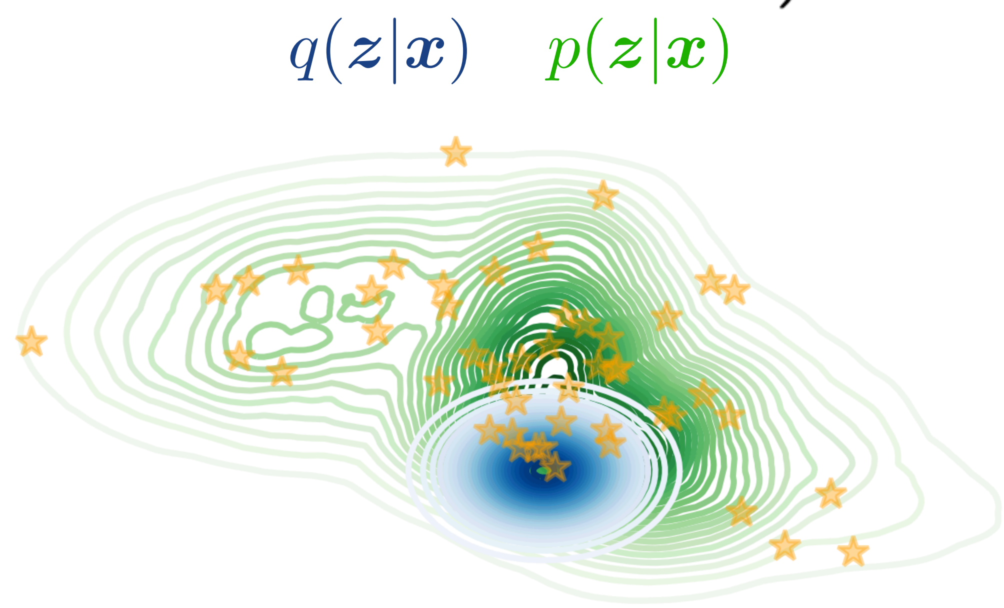
3. Train a) encoder using ELBO, b) HMC hyperparams, decoder and predictor parameters using HMC objective and c) scale using SKSD.

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

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



(a) $\log p(\boldsymbol{x}_U | \boldsymbol{x}_O)$



[2]HMC samples (orange)

Experiments

Set up

- **HH-VAEM** with 2 layers of latent variables.
- Baseline models:
  1. ***VAEM***: Gaussian-based, 1 layer.
  2. ***MIWAEM***: Gaussian-based, 1 layer, importance weighted.
  3. ***HMC-VAEM***: HMC-based, 1 layer.
  4. ***H-VAEM***: Gaussian-based, 2 layers.
- **Training**: missing features and target with a probability sampled from $U(0.01, 0.99)$ each batch.
- **Test**: 50% missing features, fully-missing target.

Experiments

Missing data imputation and target prediction

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

	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.

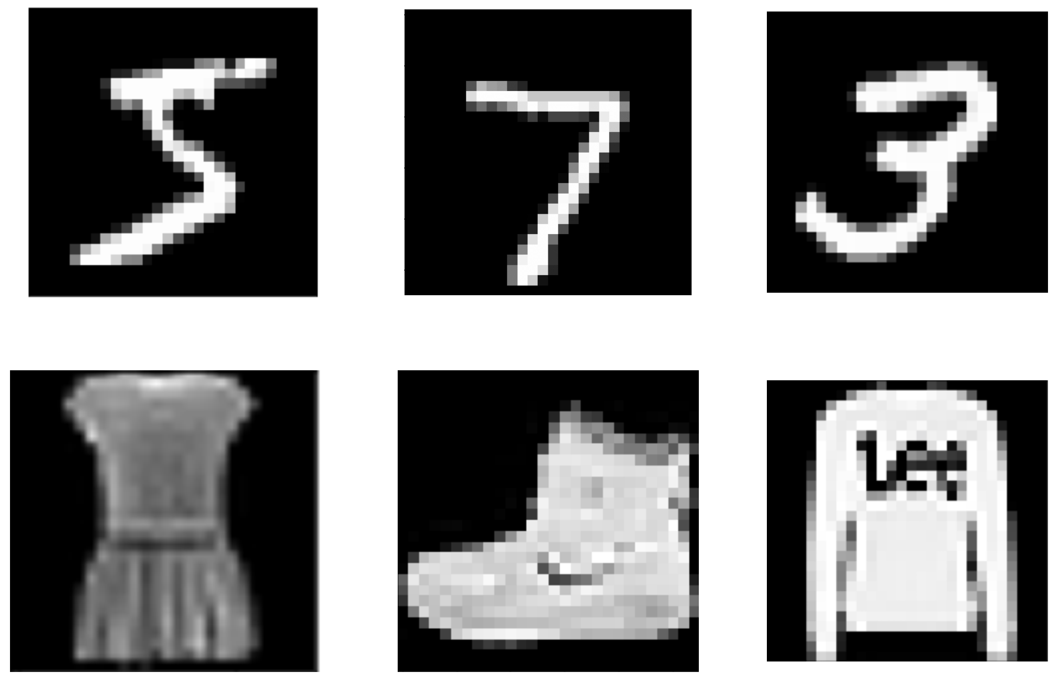
$$\log p(\boldsymbol{y}|\boldsymbol{x}_O) = \log \mathbb{E}_{\boldsymbol{\epsilon}\sim q^{(T)}(\boldsymbol{\epsilon}|\boldsymbol{x}_O)}\left[p(\boldsymbol{y}|\boldsymbol{\epsilon})\right] \approx \log \frac{1}{k}\sum_i^k p(\boldsymbol{y}|\boldsymbol{\epsilon}_i),$$

	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 **our proposed sampling-based reward**:

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

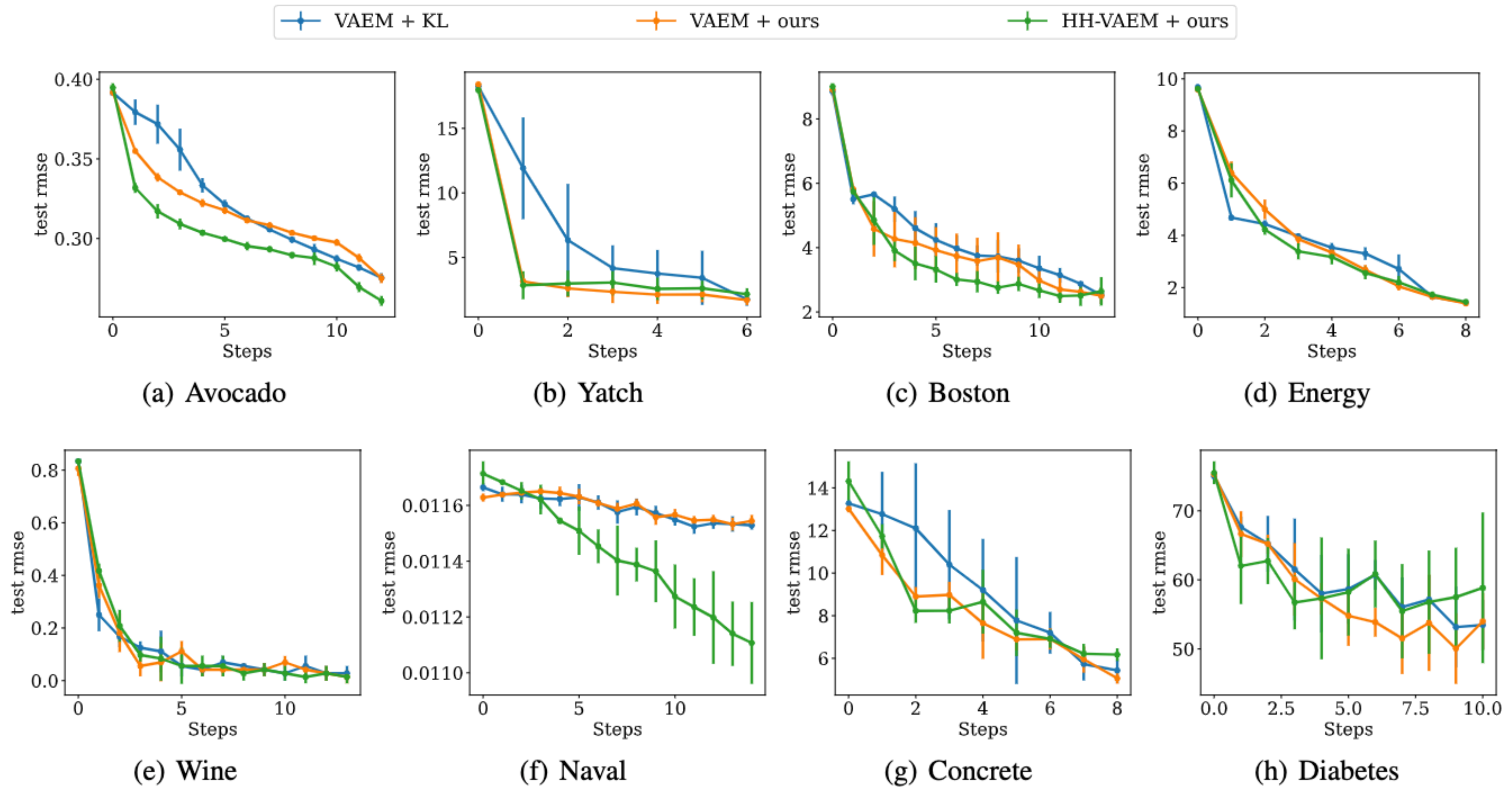


Figure 5: SAIA metric curves. Horizontal axis shows acquisition steps (number of discovered features). Vertical axis is the RMSE.

Experiments

Conditional image inpainting

1. Encode to $q_\phi^{(0)}(\boldsymbol{\epsilon} \mid \boldsymbol{z}_O, \boldsymbol{x}_O, \boldsymbol{y}_O)$

2. Using HMC, sample from $q_\phi^{(T)}(\boldsymbol{\epsilon} \mid \boldsymbol{z}_O, \boldsymbol{x}_O, \boldsymbol{y}_O)$

3. Decode to $p\left(\boldsymbol{x}_U \mid \boldsymbol{\epsilon}^{(T)}\right)$

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 information acquisition task.

Further details

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

Ignacio Peis
Universidad Carlos III de Madrid
Madrid, Spain
ipeis@tsc.uc3m.es

Chao Ma
Microsoft Research
University of Cambridge
Cambridge, UK
cm905@cam.ac.uk

José Miguel Hernández-Lobato
University of Cambridge
Cambridge, UK
jmh233@cam.ac.uk

[Code]

[Automatic HMC code]

[Paper]

[15] Peis et at., 2022

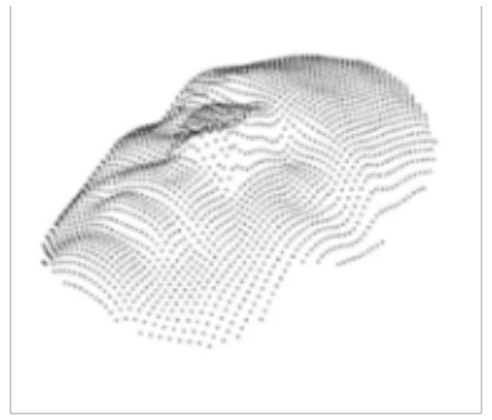
Part II
Variational Mixture of HyperGenerators for Learning Distributions over Functions

Motivation

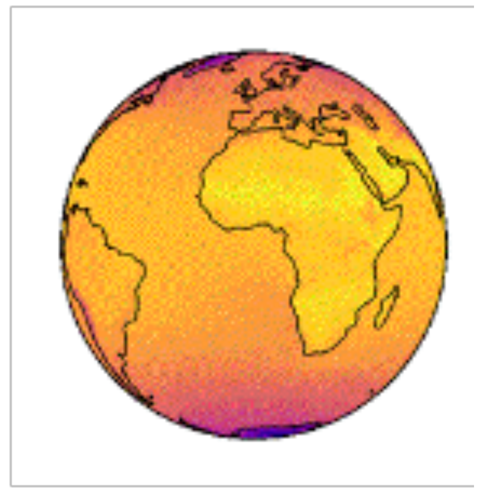
- We typically deal with discretized versions of data that are continuous in nature.

2D Images

3D Images

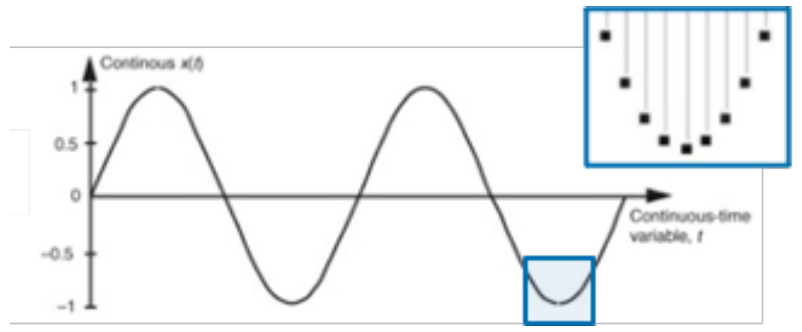


Polar data



Spatial

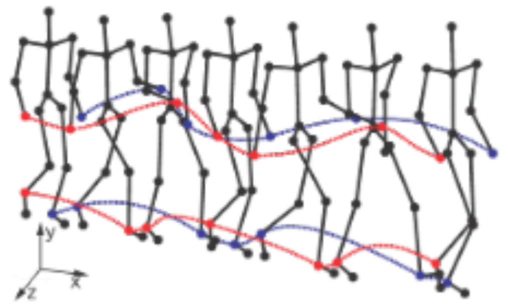
Time series



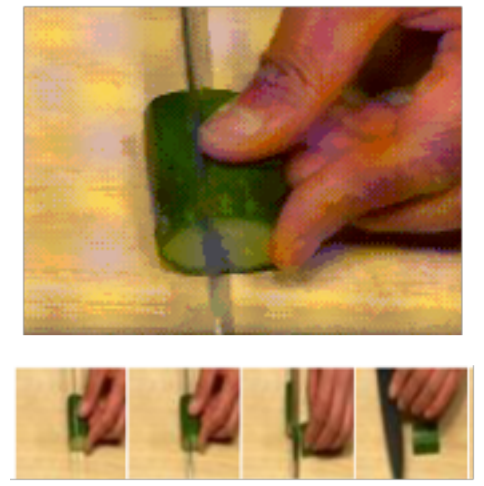
Audio

Temporal

Motion sequences



Video

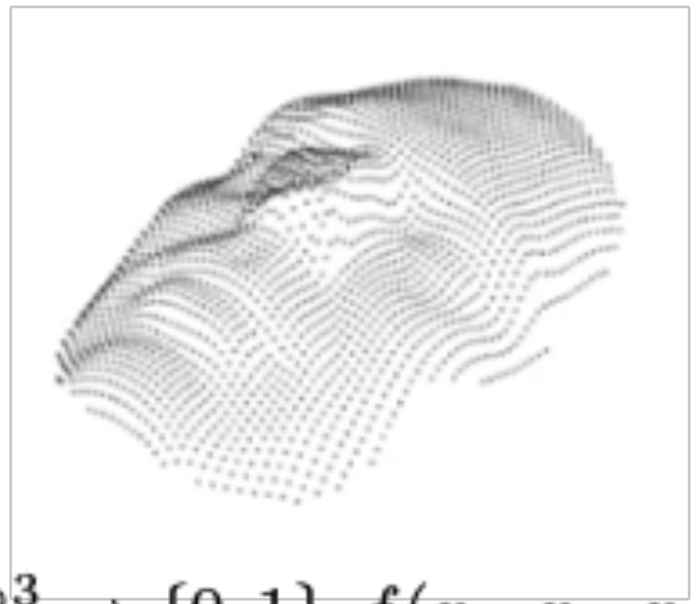


Spatio-temporal

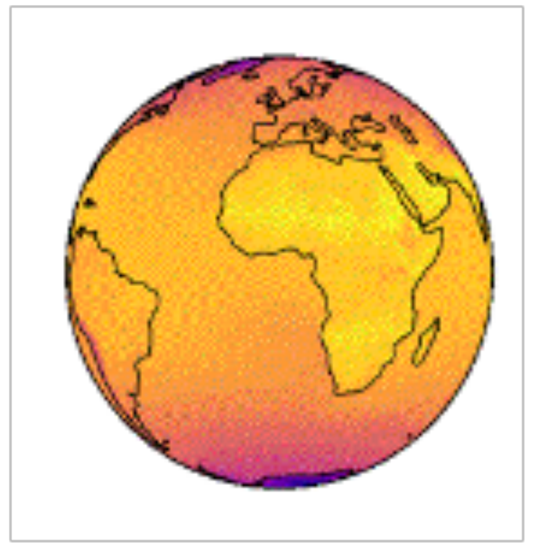
Motivation

- Data can be expressed as a function over continuous coordinate systems.

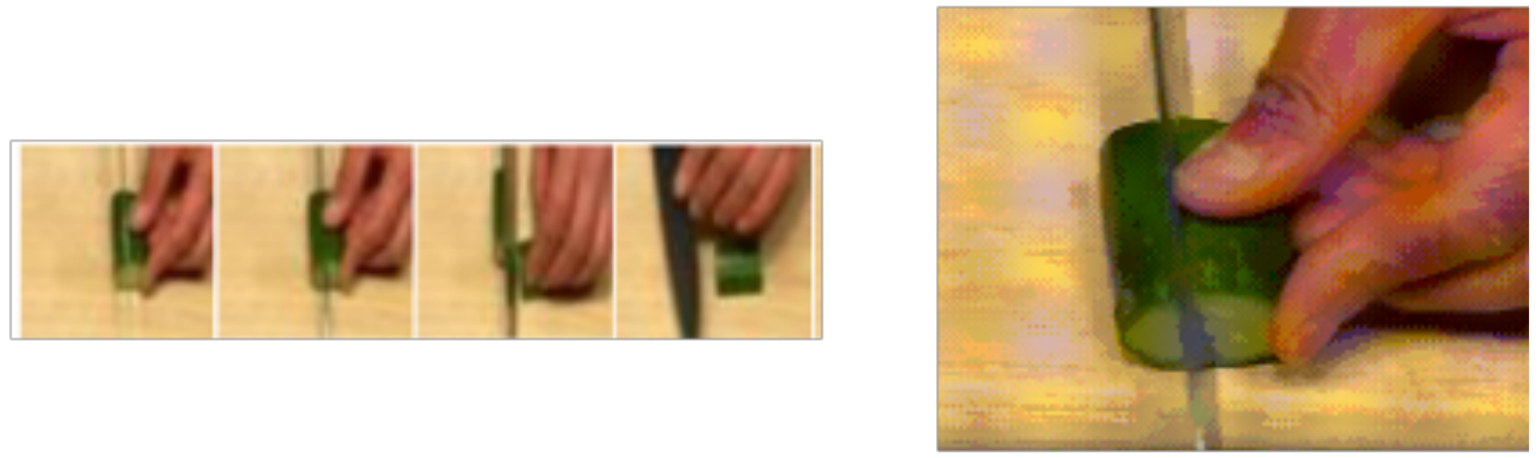
$f:\mathbb{R}^2 \to \mathbb{R}^3, f(x_1,x_2)=(r,g,b)$



$f:\mathbb{R}^3 \to \{0,1\}, f(x_1,x_2,x_3)=p$



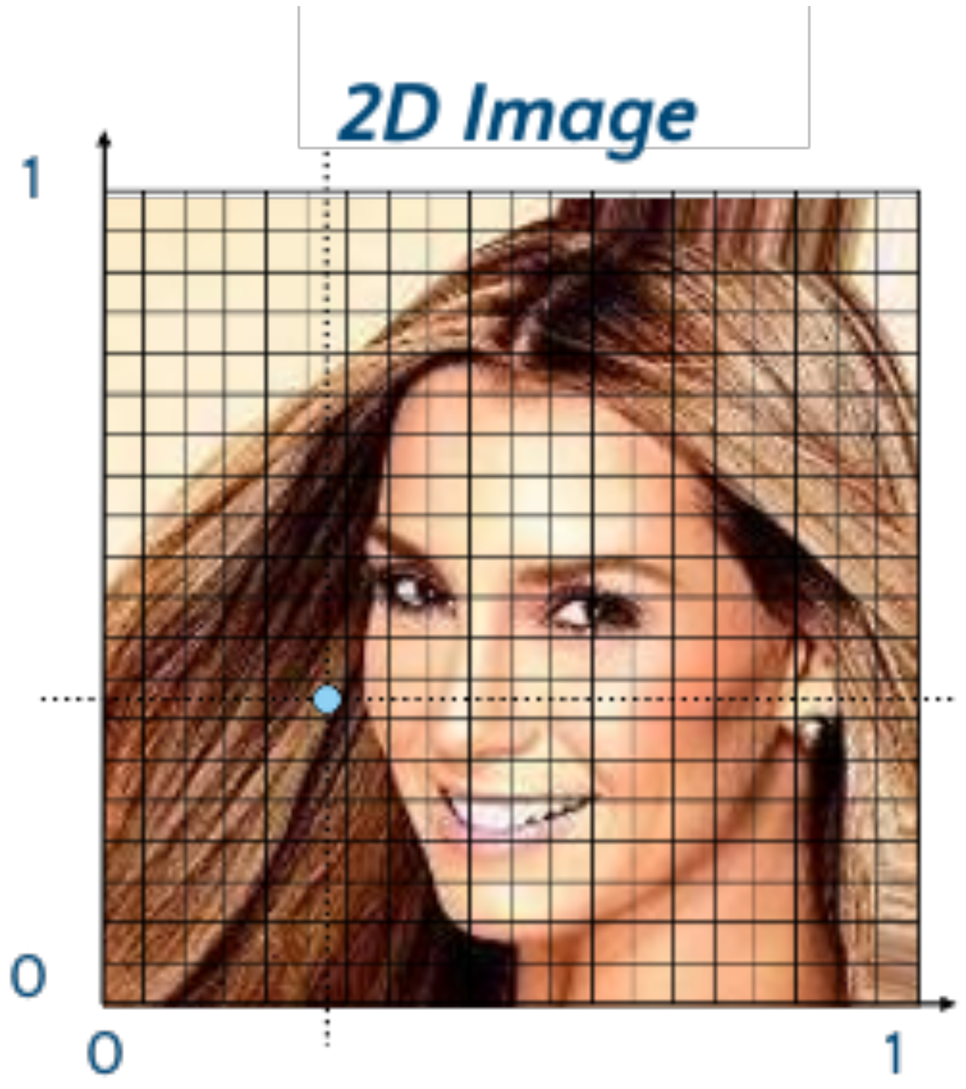
$f:\mathbb{R}^2 \to \mathbb{R}, f(\varphi,\lambda)=T$



$f:\mathbb{R}^3 \to \mathbb{R}^3, f(x_1,x_2,t)=(r,g,b)$

Motivation

- Focusing on images:

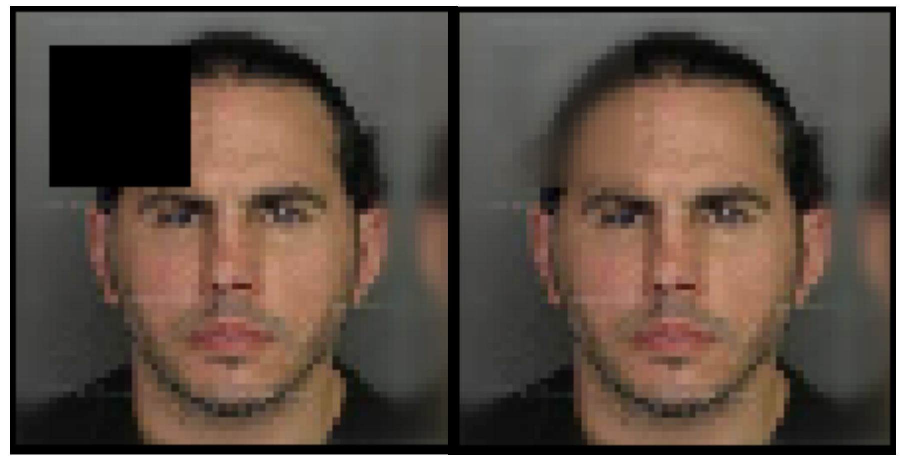


- Each pixel is now a pair $\{\boldsymbol{x}_d, \boldsymbol{y}_d\}$ where $x_d \in \mathbb{R}^2$, $y_d \in \mathbb{R}^3$
- Full image is a pair of sets $\boldsymbol{X} = \{\boldsymbol{x}_d\}_{d=1}^{D}$, $\boldsymbol{Y}_d = \{\boldsymbol{y}_d\}_{d=1}^{D}$
- Generator function $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ creates this speficic image with the mapping $f(\boldsymbol{x}_d) = \boldsymbol{y}_d$, $d \in [1, ..., D]$

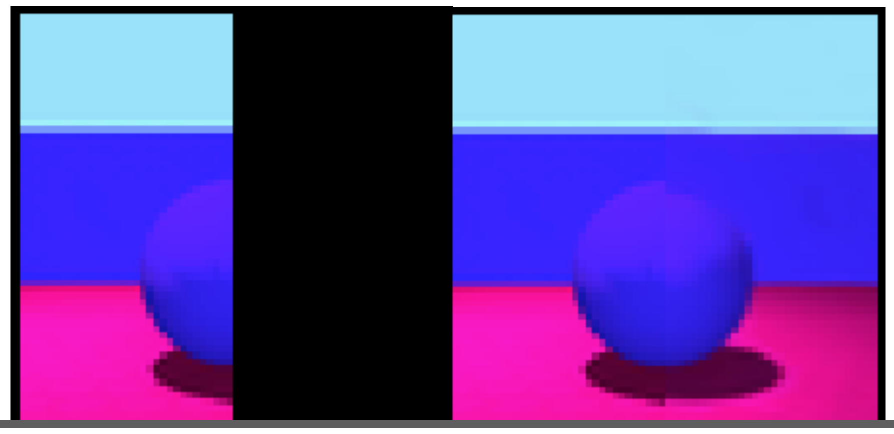
Motivation

- This approach will easily allows for:

Inpainting [6,7]

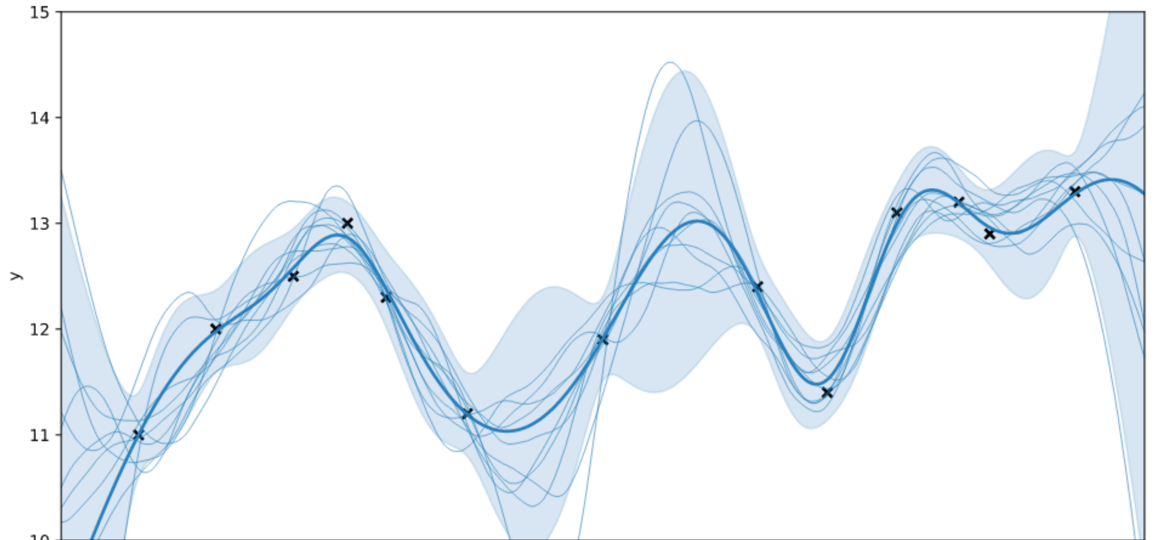


Outpainting [6,7]



Superresolution [6,7]

Conditional Generation

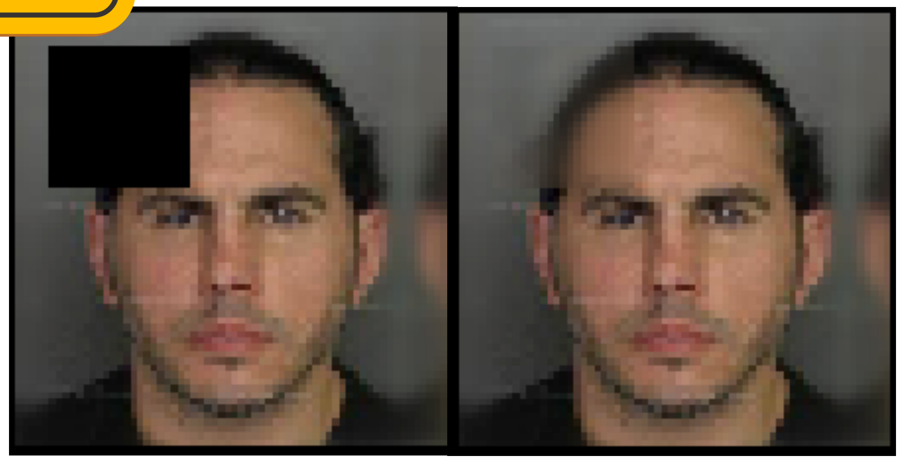


Interpolation

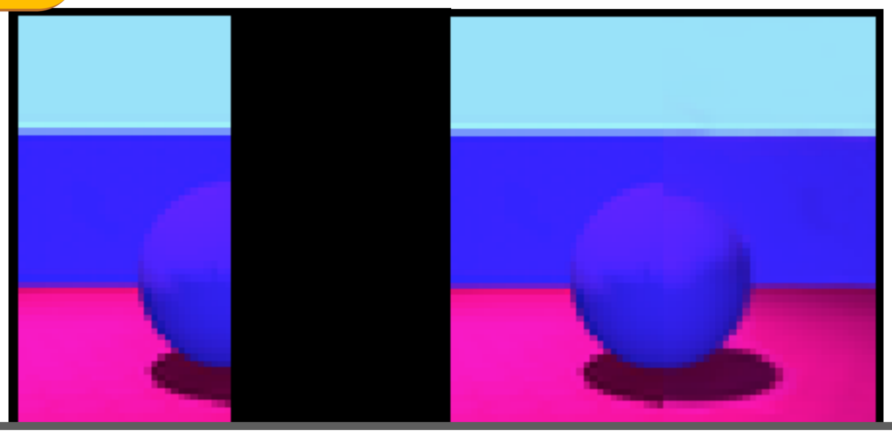
Motivation

- This approach will easily allows for:

Inpainting [6,7]

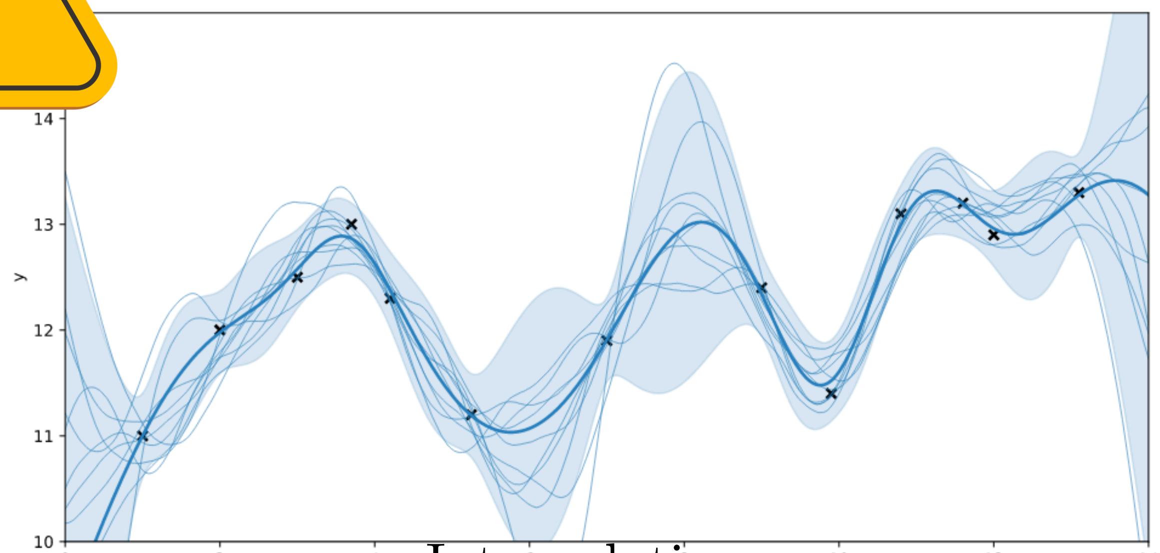


Outpainting [6,7]



Superresolution [6,7]

Conditional Generation

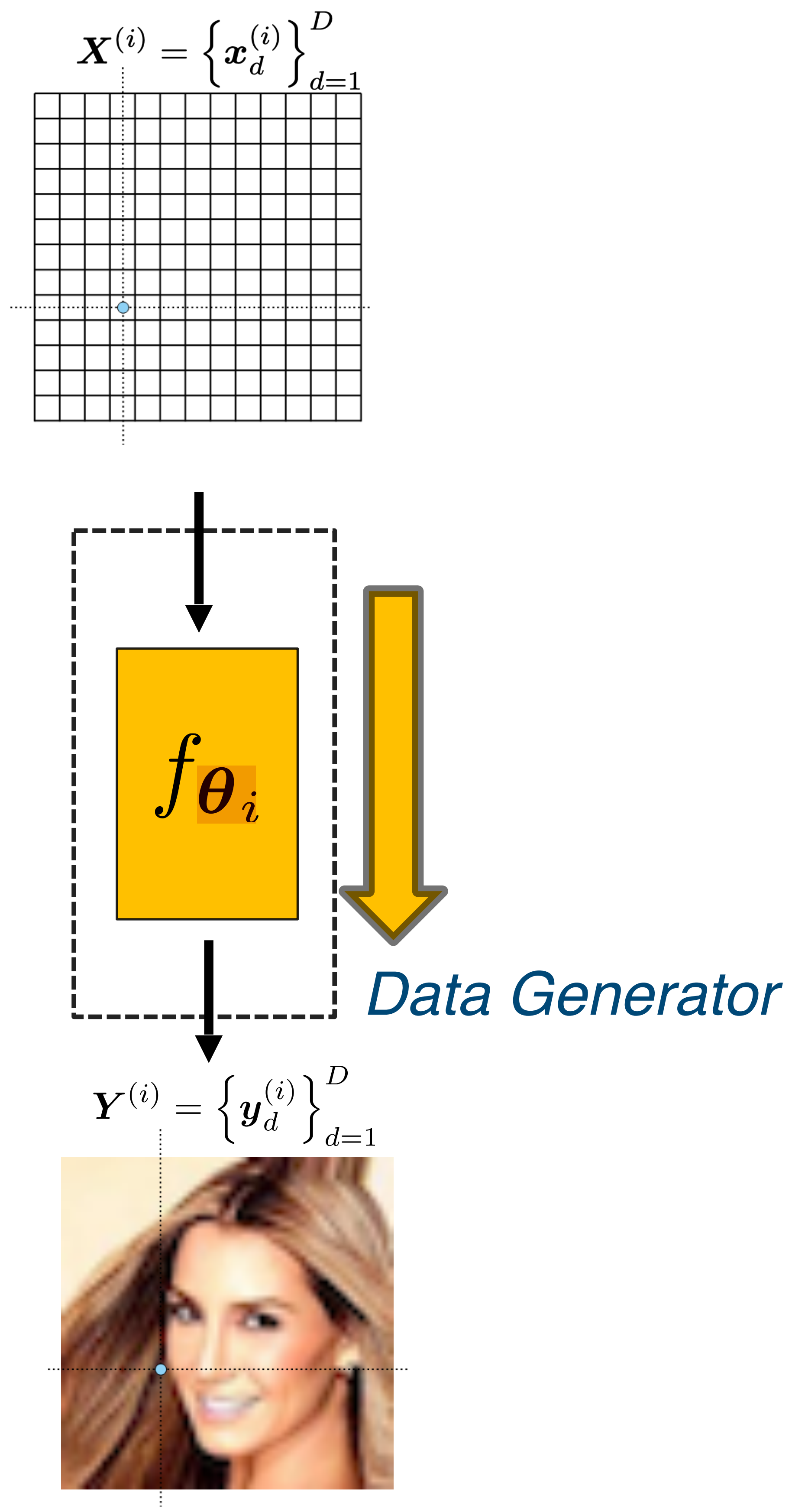


Interpolation

These ones require inference!

Implicit Neural Representations

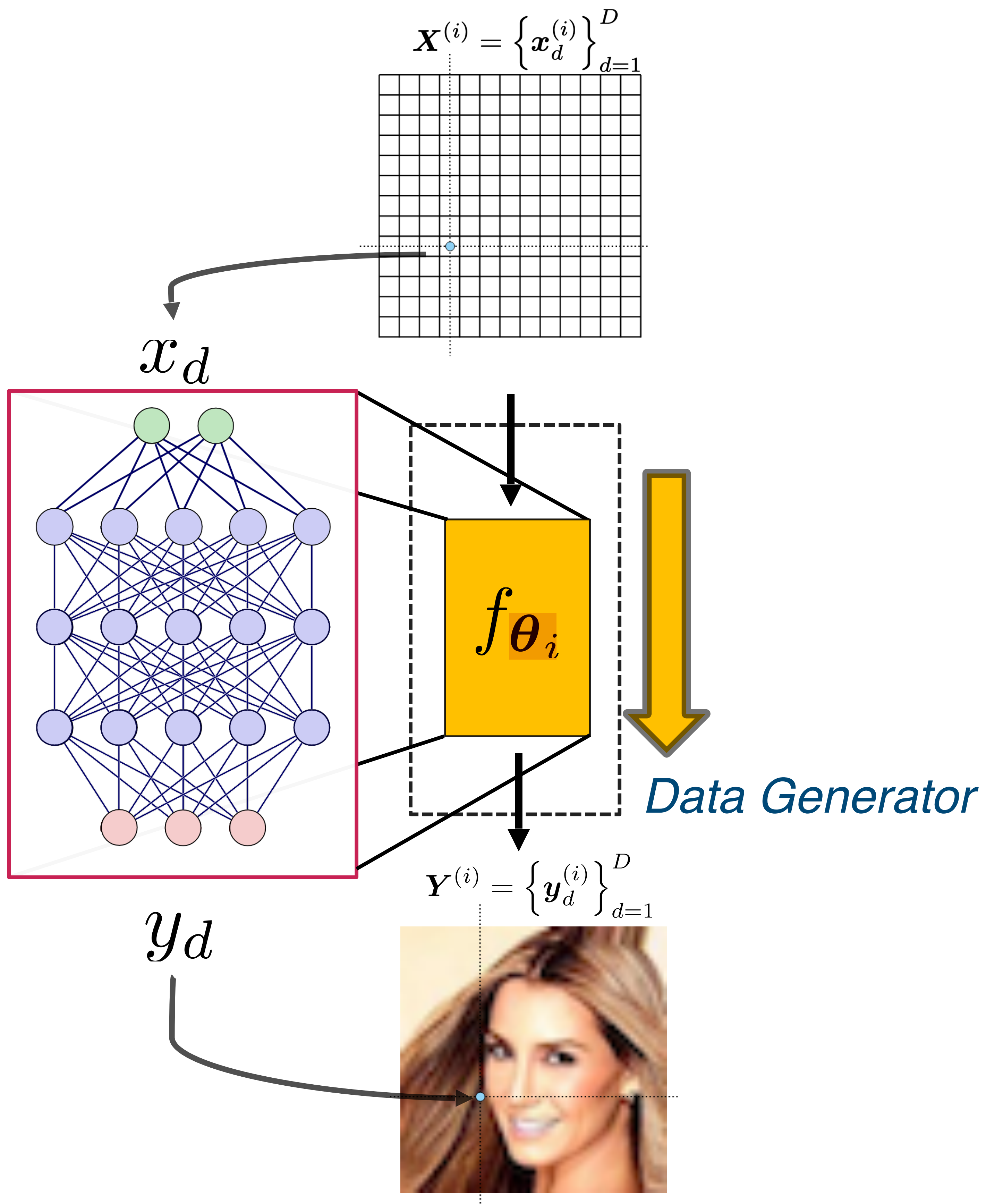
INRs [20-22]



[20] Sitzmann et at., 2020 [21] Mescheder et at., 2019 [20] Sitzmann et at., 2019

Implicit Neural Representations

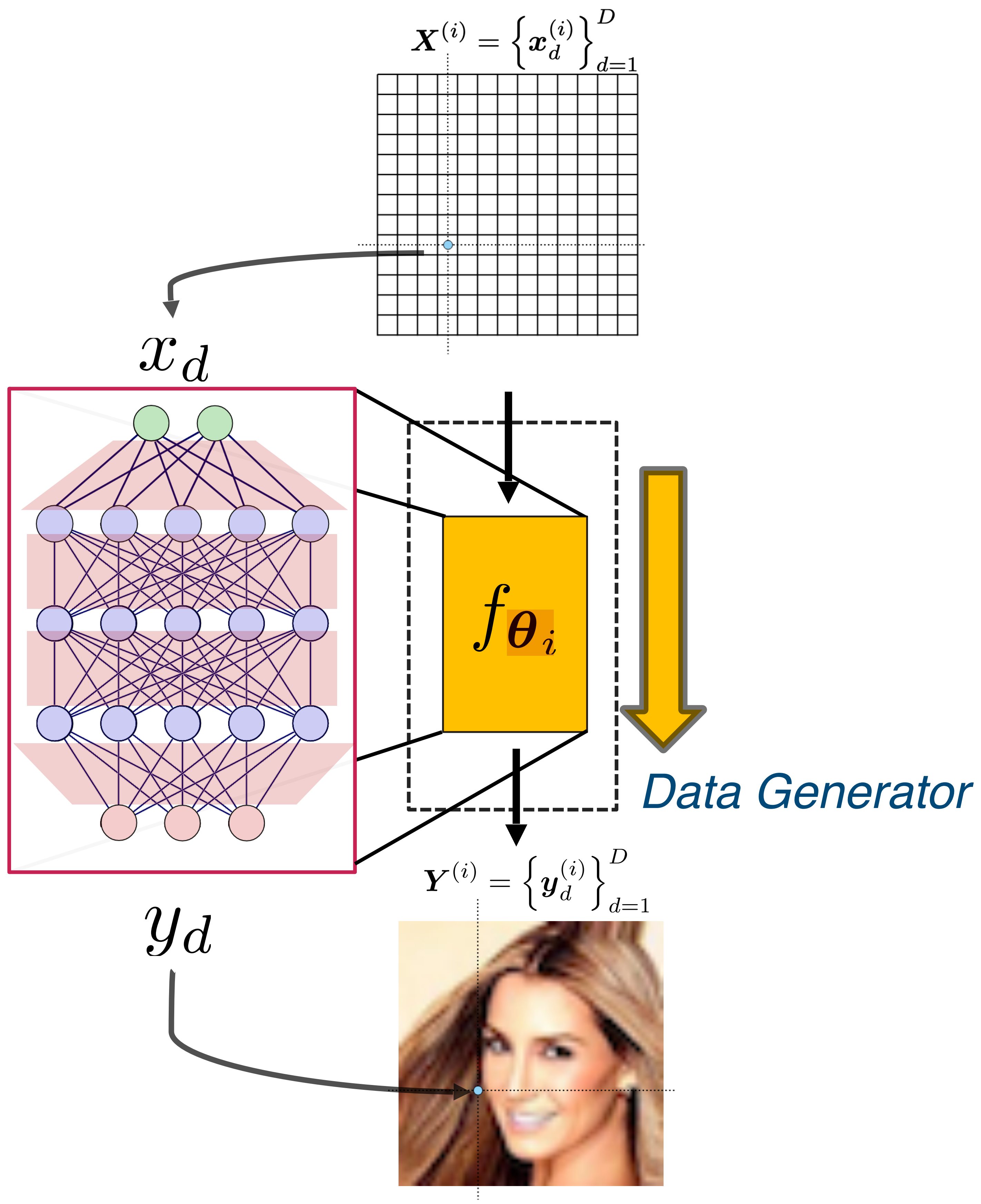
INRs [20-22]



[20] Sitzmann et at., 2020 [21] Mescheder et at., 2019 [20] Sitzmann et at., 2019

Implicit Neural Representations

INRs [20-22]



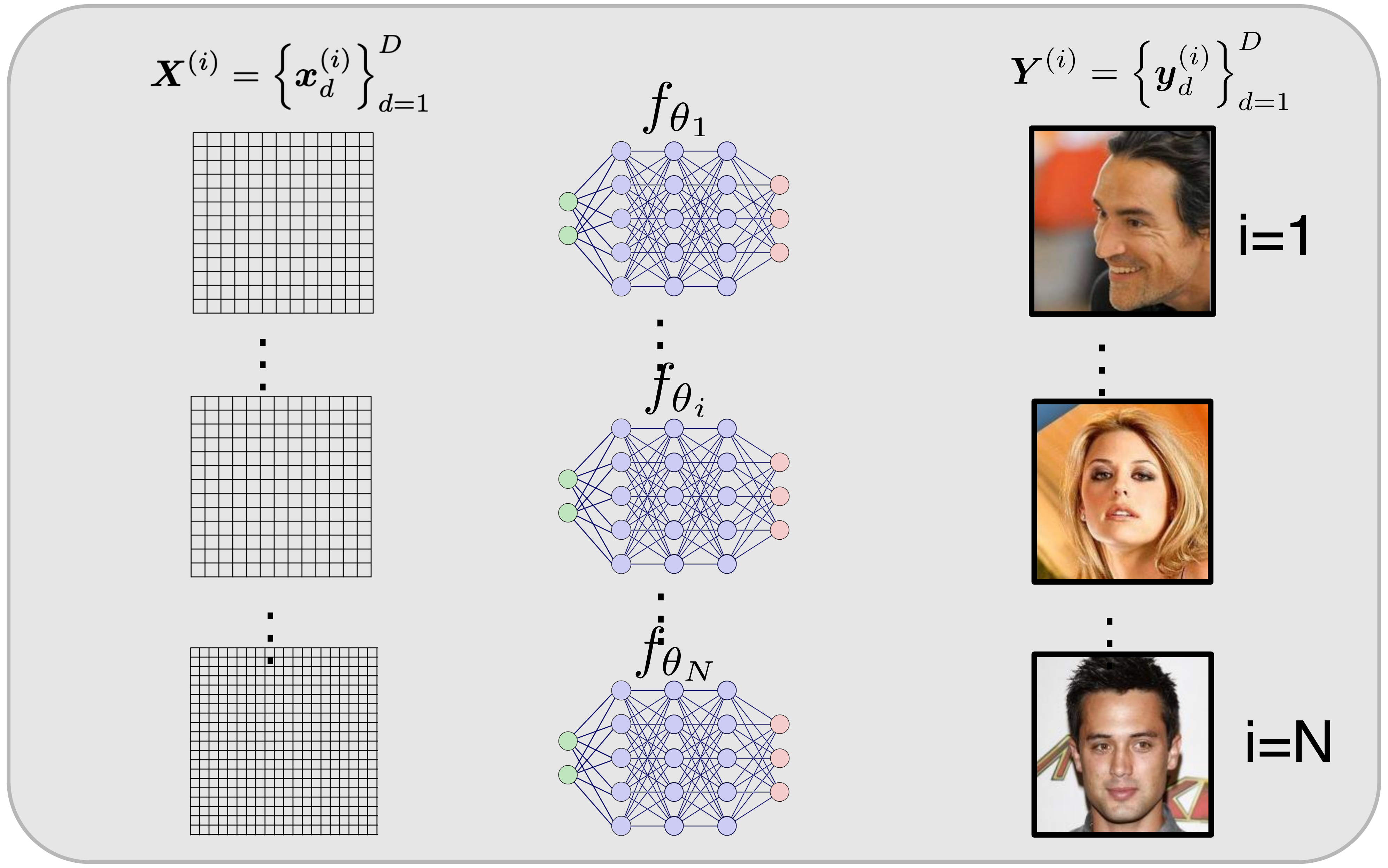
[20] Sitzmann et at., 2020 [21] Mescheder et at., 2019 [20] Sitzmann et at., 2019

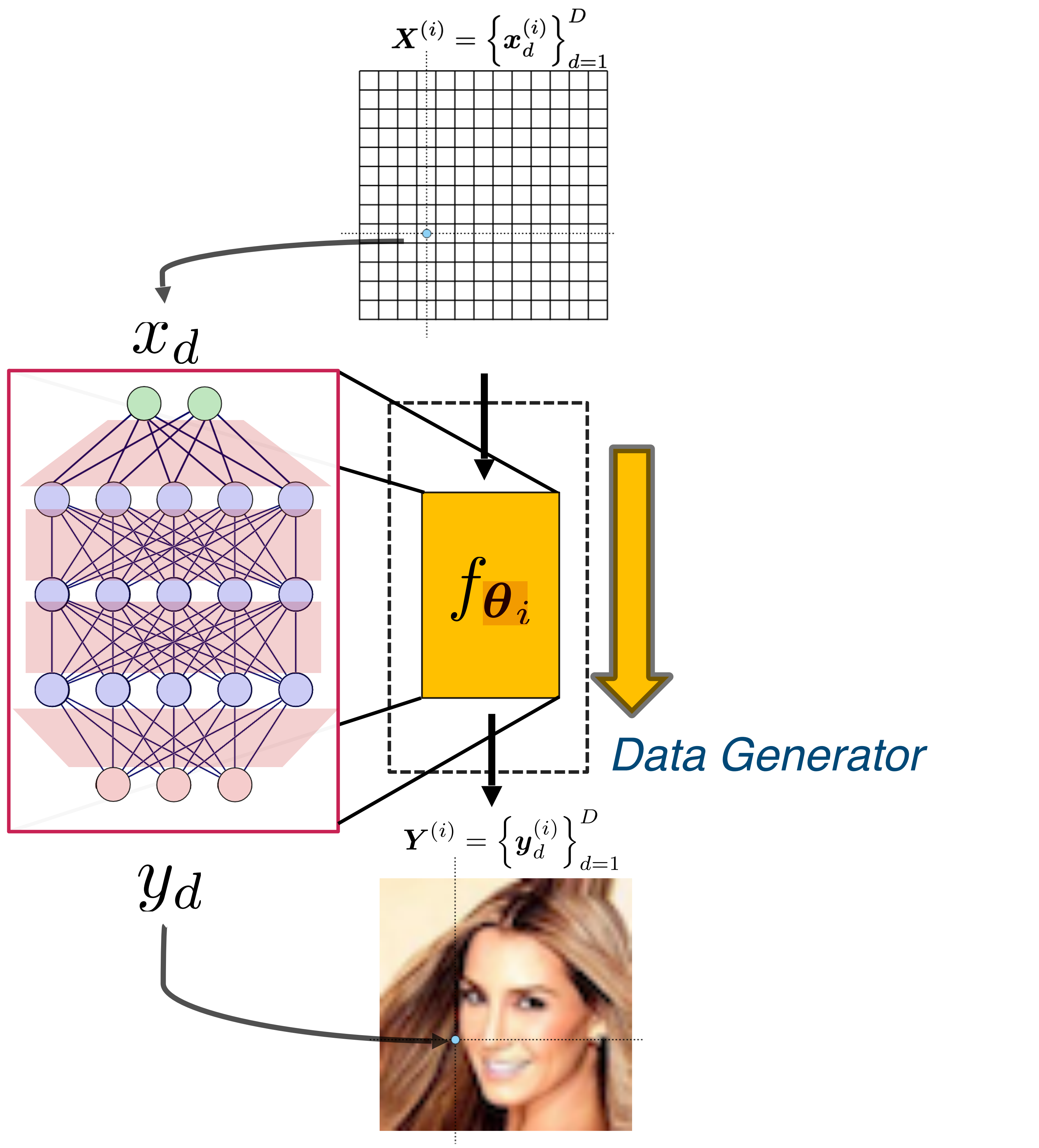
Implicit Neural Representations

INRs [20-22]

- Learning distributions functions within a VAE framework.

Data generator $f_{\boldsymbol{\theta}_i}$ is unique to each image





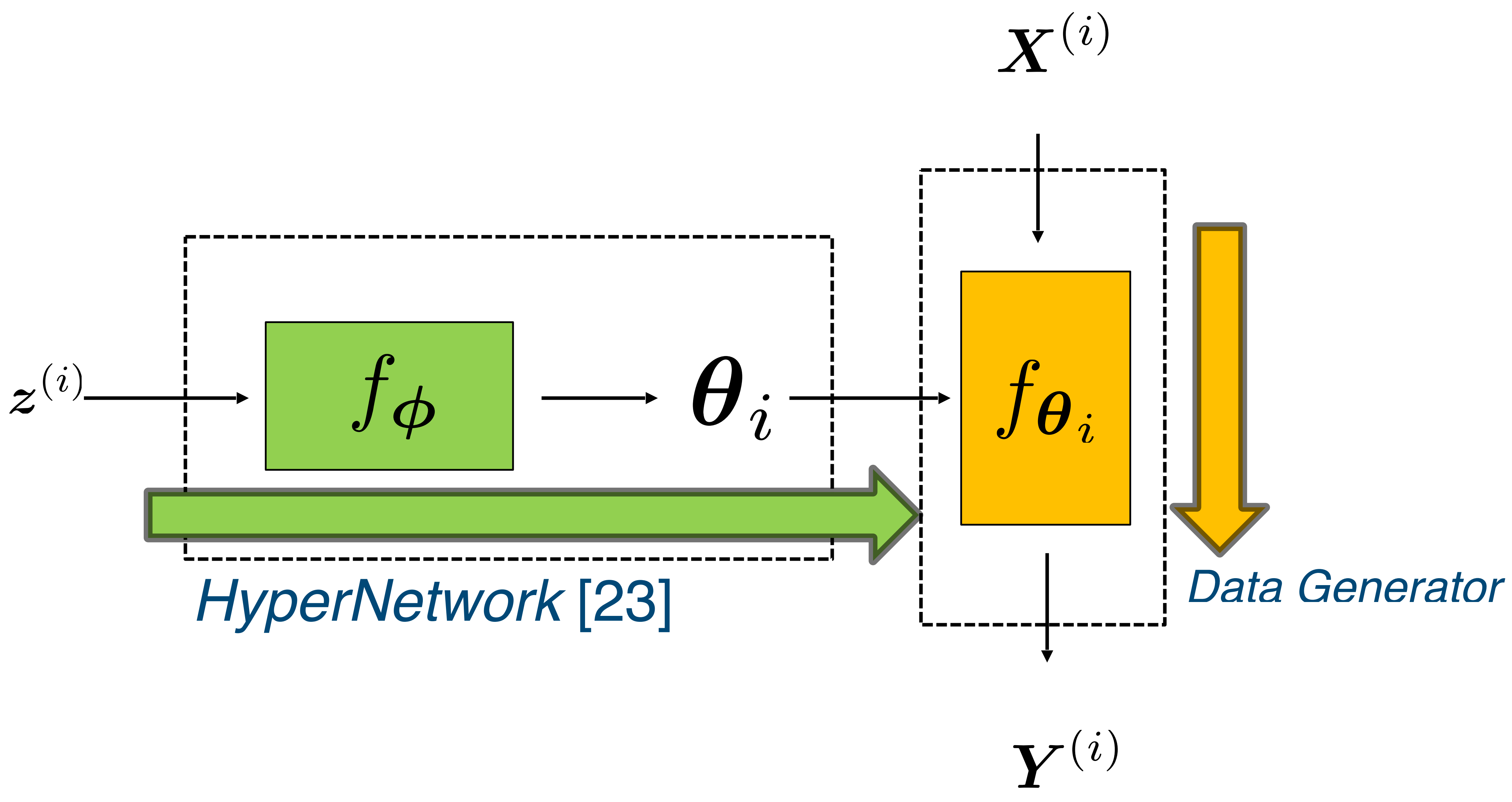
[20] Sitzmann et at., 2020 [21] Mescheder et at., 2019 [20] Sitzmann et at., 2019

Deep Generative Models of INRs

How to scale to large datasets?

How to map a latent representation to an INR?

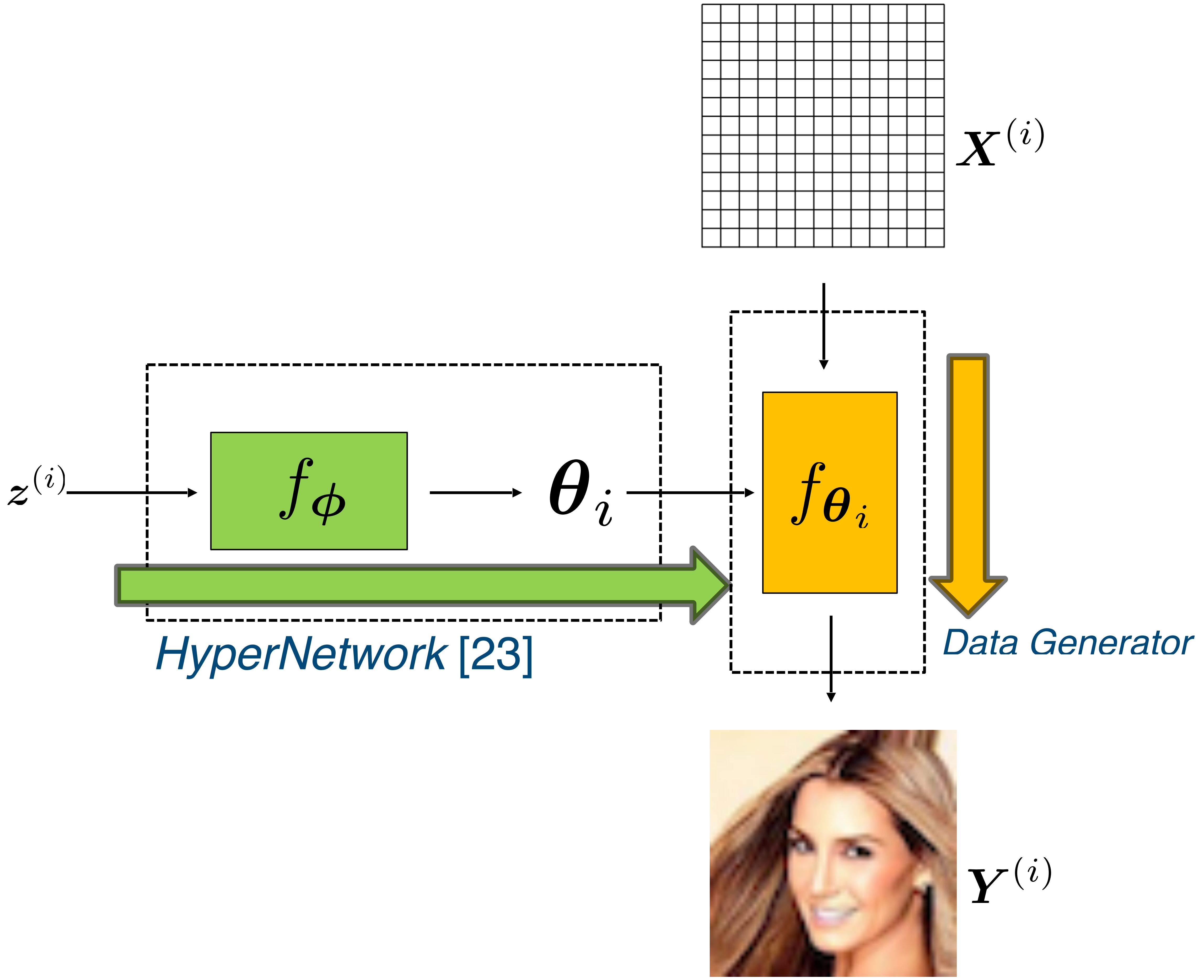
Deep Generative Models of INRs



[23] Ha et at., 2017

Deep Generative Models of INRs

Have $\boldsymbol{z}^{(i)}$, a summary representation of image.



[23] Ha et at., 2017

Previous work

GASP[5]

- Adversarial training:

Figure 5: Training procedure for GASP: 1. Sample a function and evaluate it at a set of coordinate locations to generate fake point cloud. 2. Convert real data sample to point cloud. 3. Discriminate between real and fake point clouds.

❌ Can't tackle inference related tasks.

[5] Dupont et at., 2020

Previous work

GASP[5]

- Adversarial training:

Point-wise Convolution

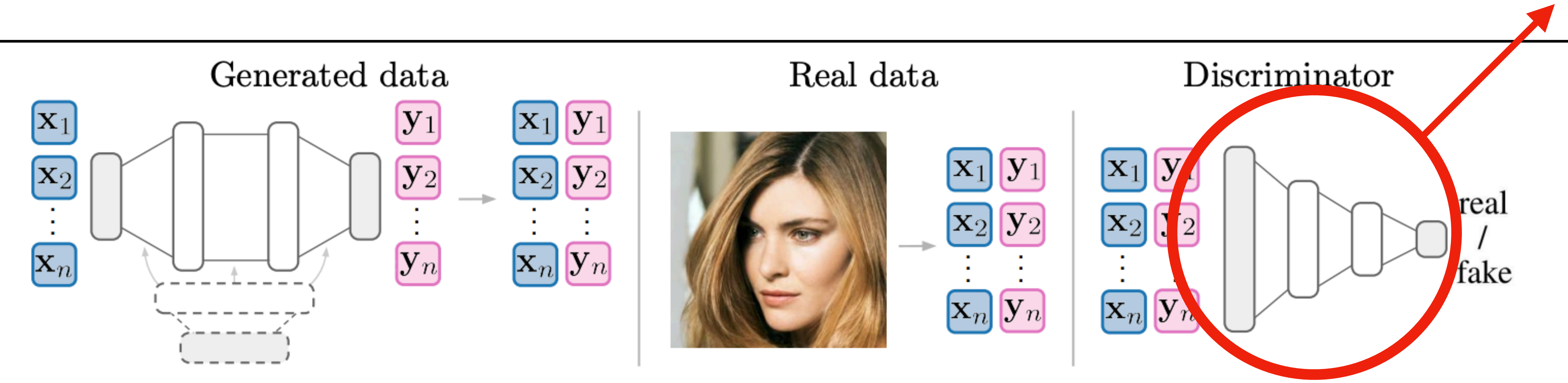


Figure 5: Training procedure for GASP: 1. Sample a function and evaluate it at a set of coordinate locations to generate fake point cloud. 2. Convert real data sample to point cloud. 3. Discriminate between real and fake point clouds.

❌ Can't tackle inference related tasks.

[5] Dupont et at., 2020

Previous work

Functas[6]

- Decoupled training:

    1. Fit an INR per datapoint using SIREN[20] and **modulation vectors**, named **functas**.

    2. Train any generative model on the functa dataset of vectors.

❌ Computationally expensive inference.

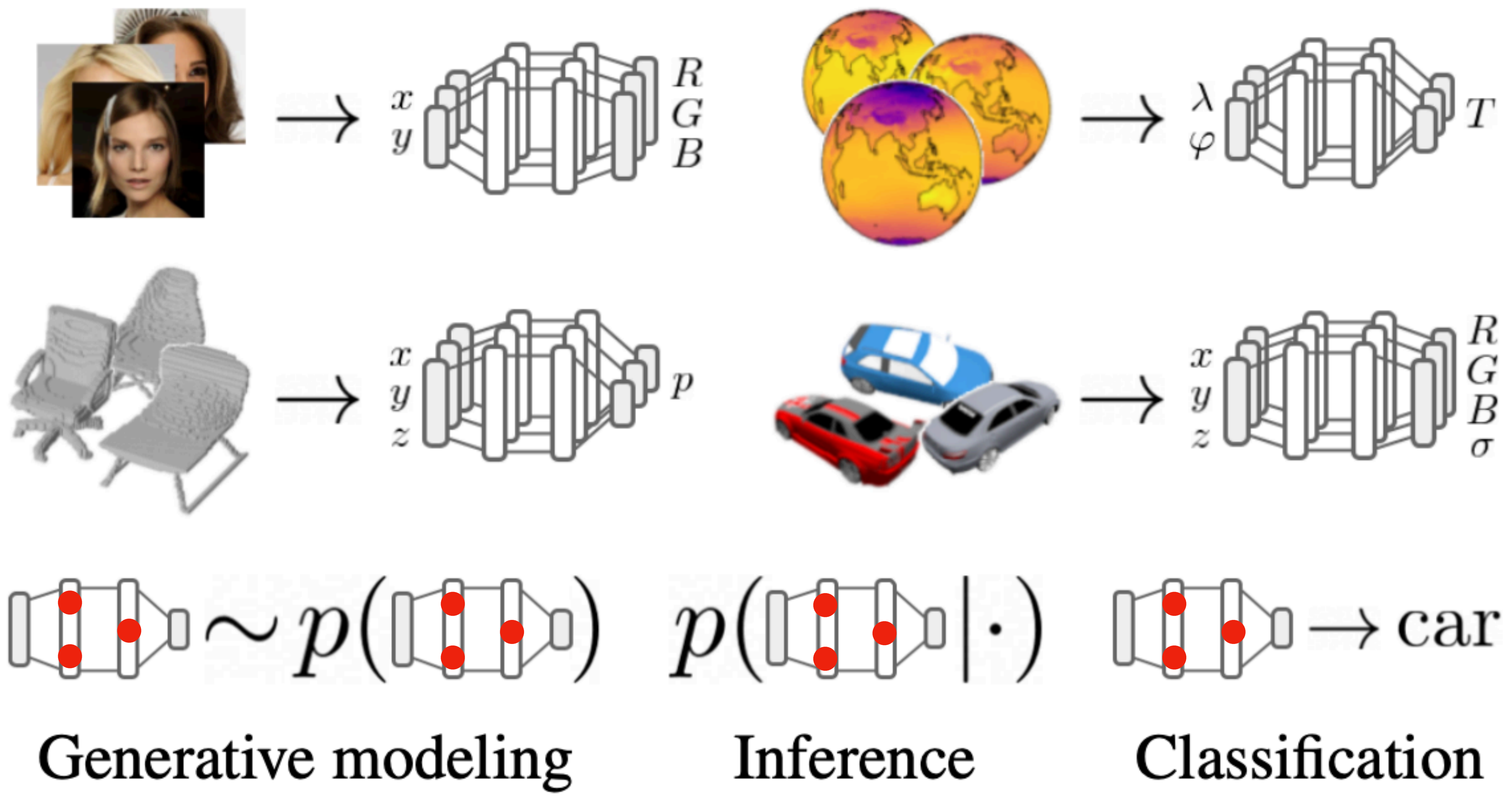


Figure 1. We convert array data into functional data parameterized by neural networks, termed *functa*, and treat these as data points for various downstream machine learning tasks.

[6] Dupont et at., 2022 [20] Sitzmann et at., 2020

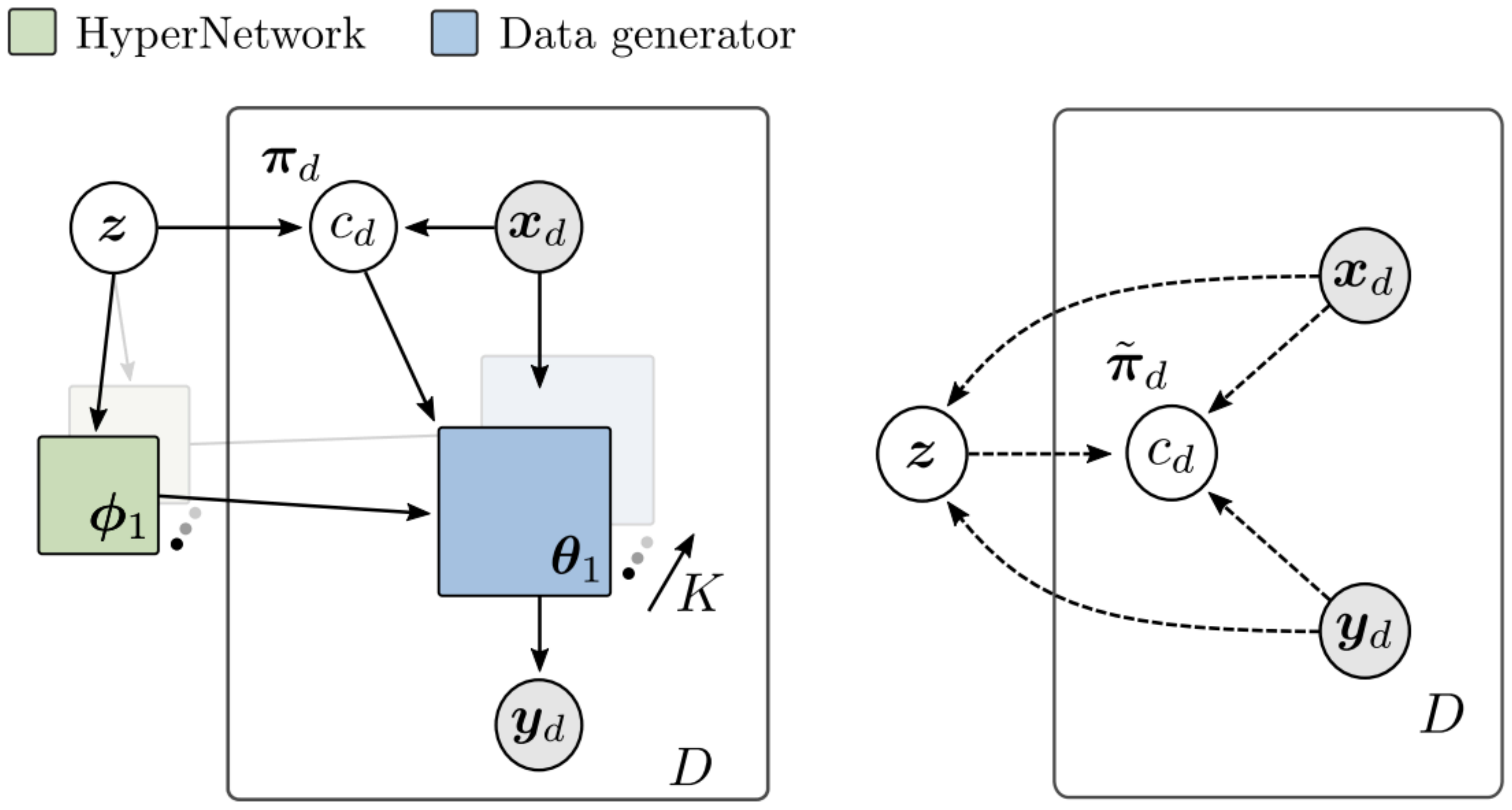
Deep Generative Models of INRs

How to infer the latent representation $\boldsymbol{z}$?

$$q_\gamma(\boldsymbol{z}|\boldsymbol{Y}, \boldsymbol{X}) \qquad\qquad p_\psi(\boldsymbol{z})$$

Proposed method: VAMoH

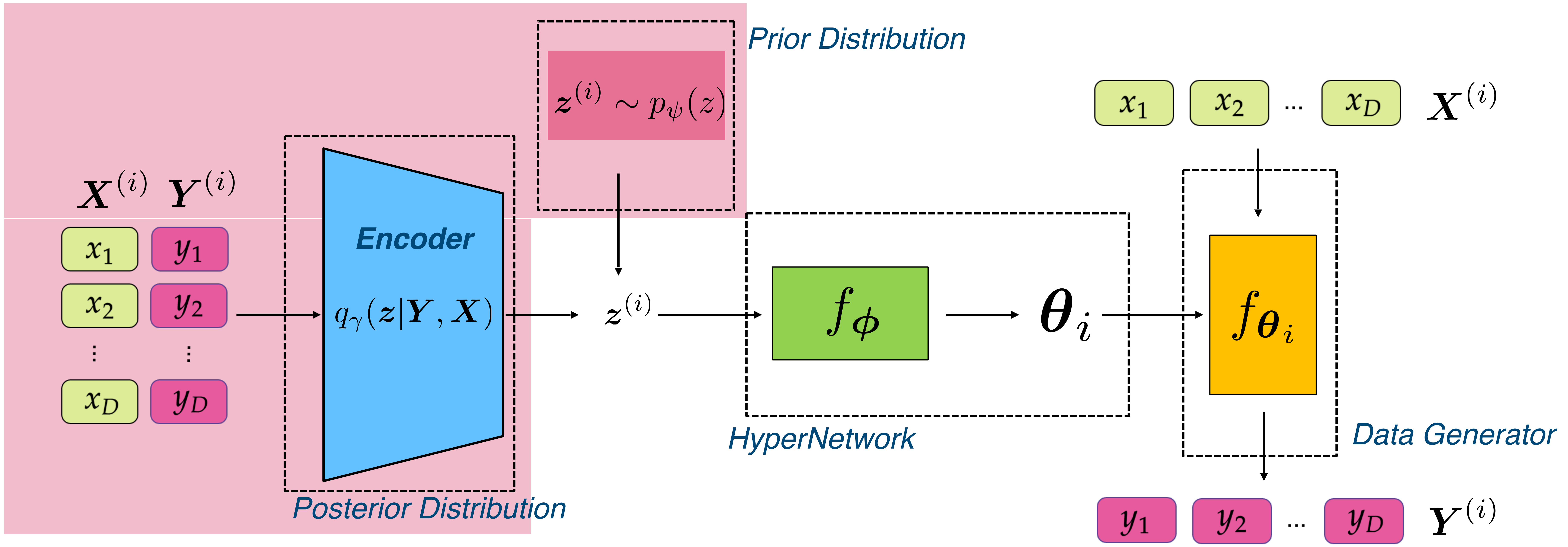
Variational Mixture of HyperGenerators



(a) Generative model (b) Inference model

VAMoH

Encoder

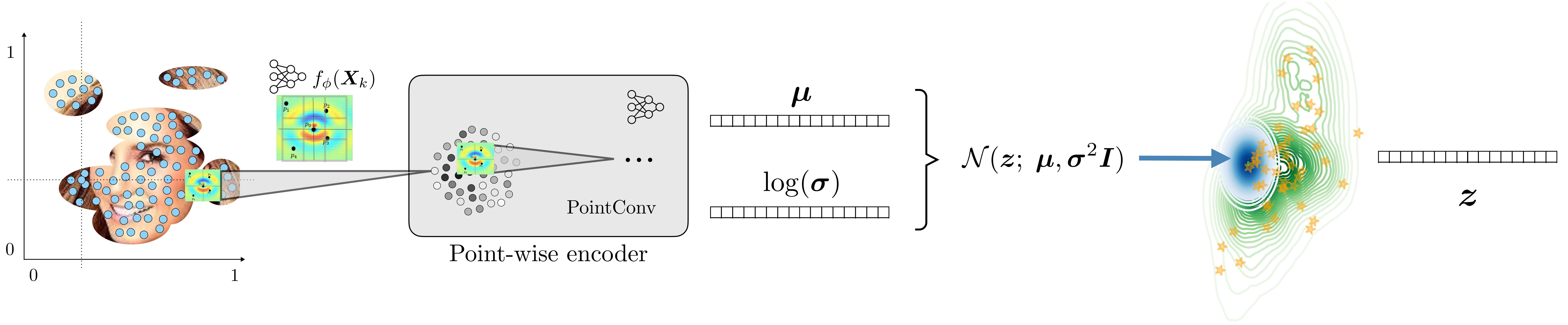


$z^{(i)}$: *Latent Variable*

VAMoH

Encoder

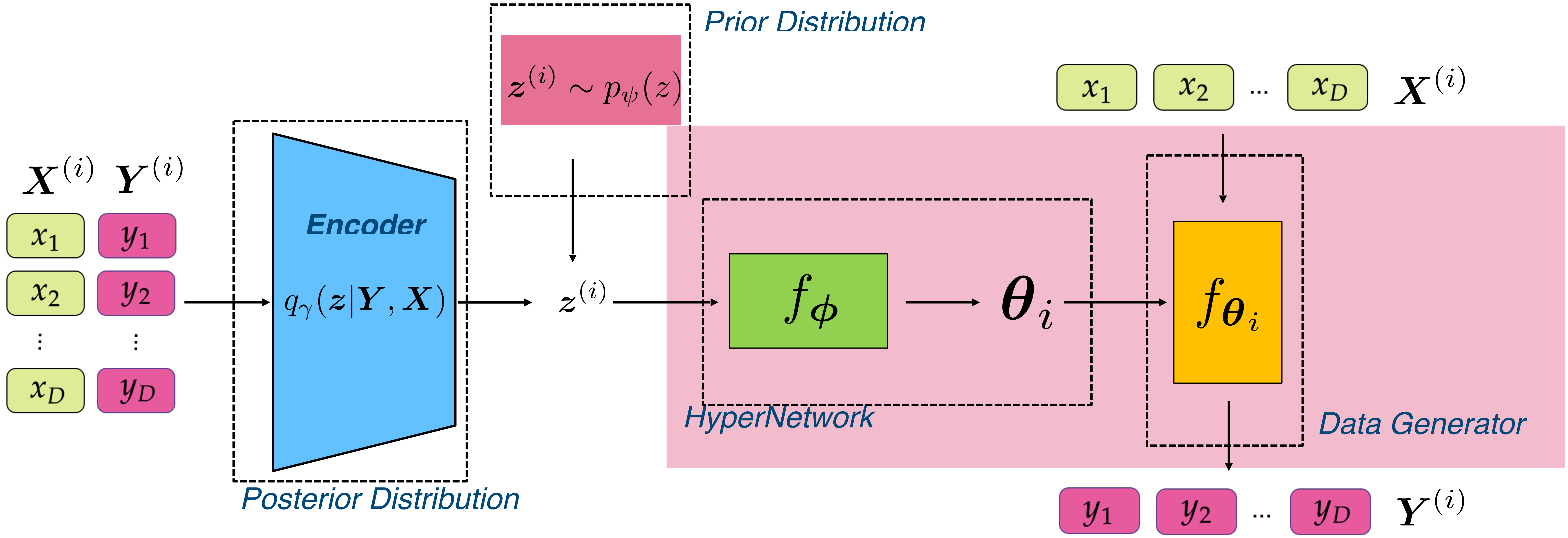
- PointConv[21] encoder for point clouds.



[21] Wu et at., 2019

VAMoH

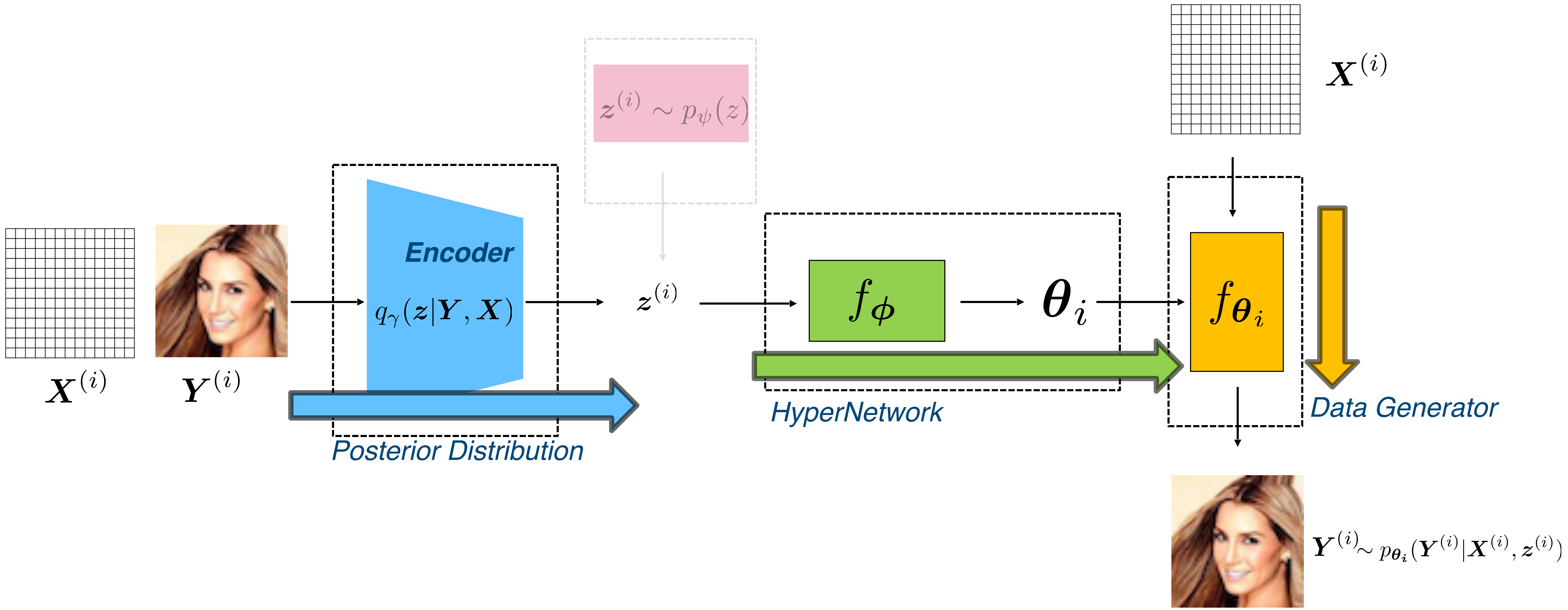
Decoder



$z^{(i)}$: *Latent Variable*

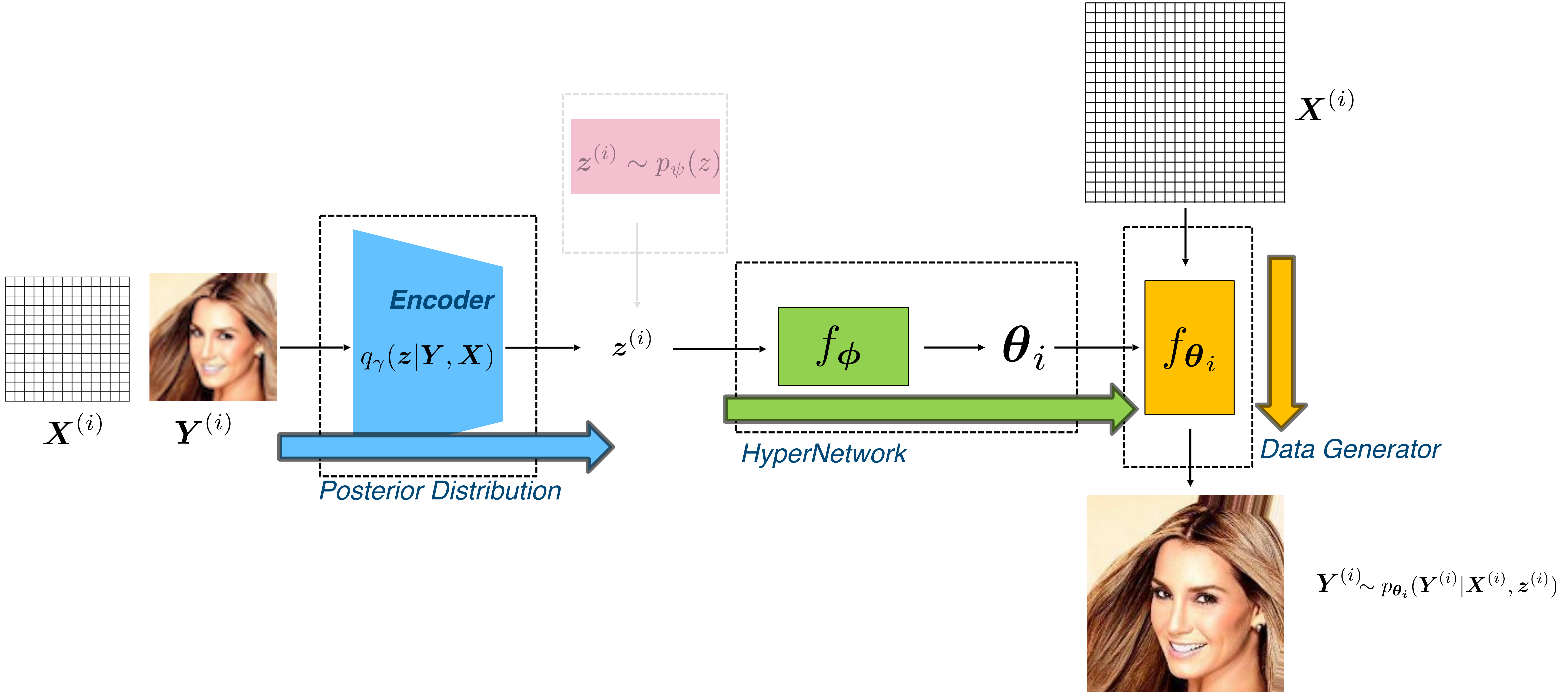
VAMoH

Reconstruction



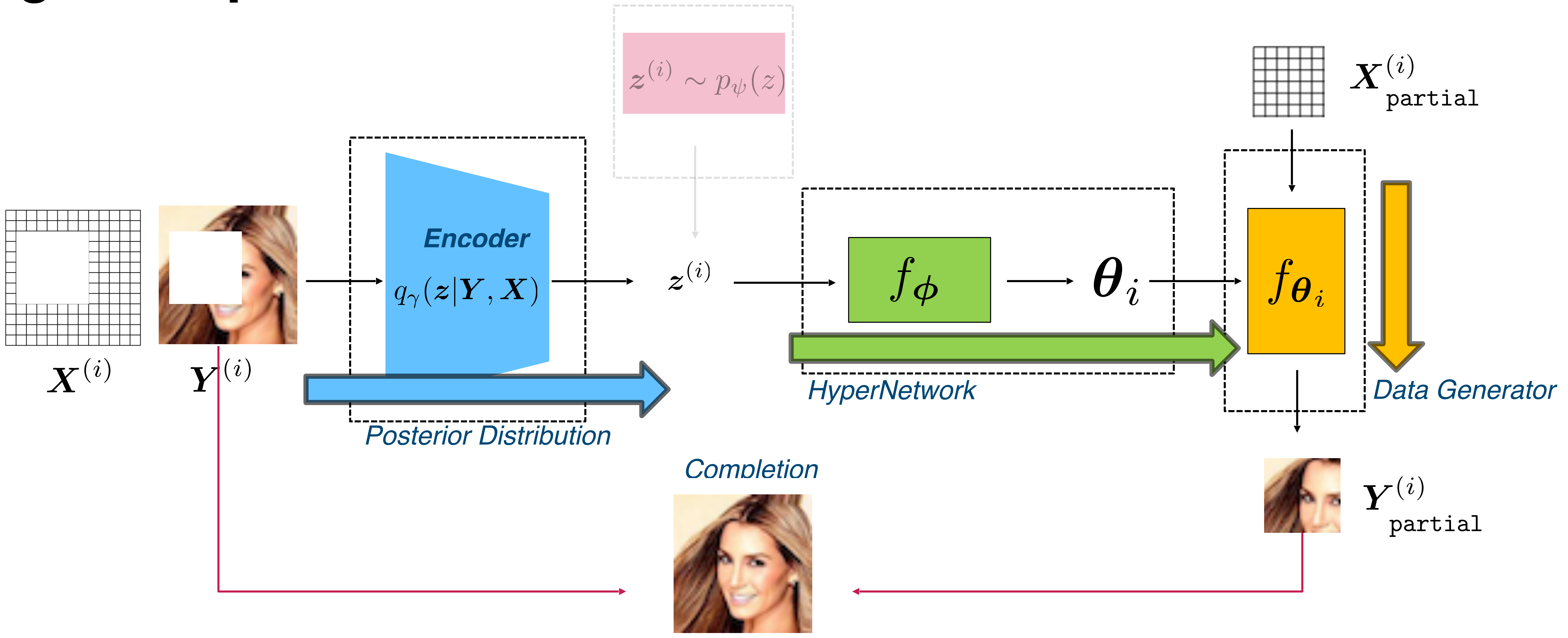
VAMoH

Super Resolution



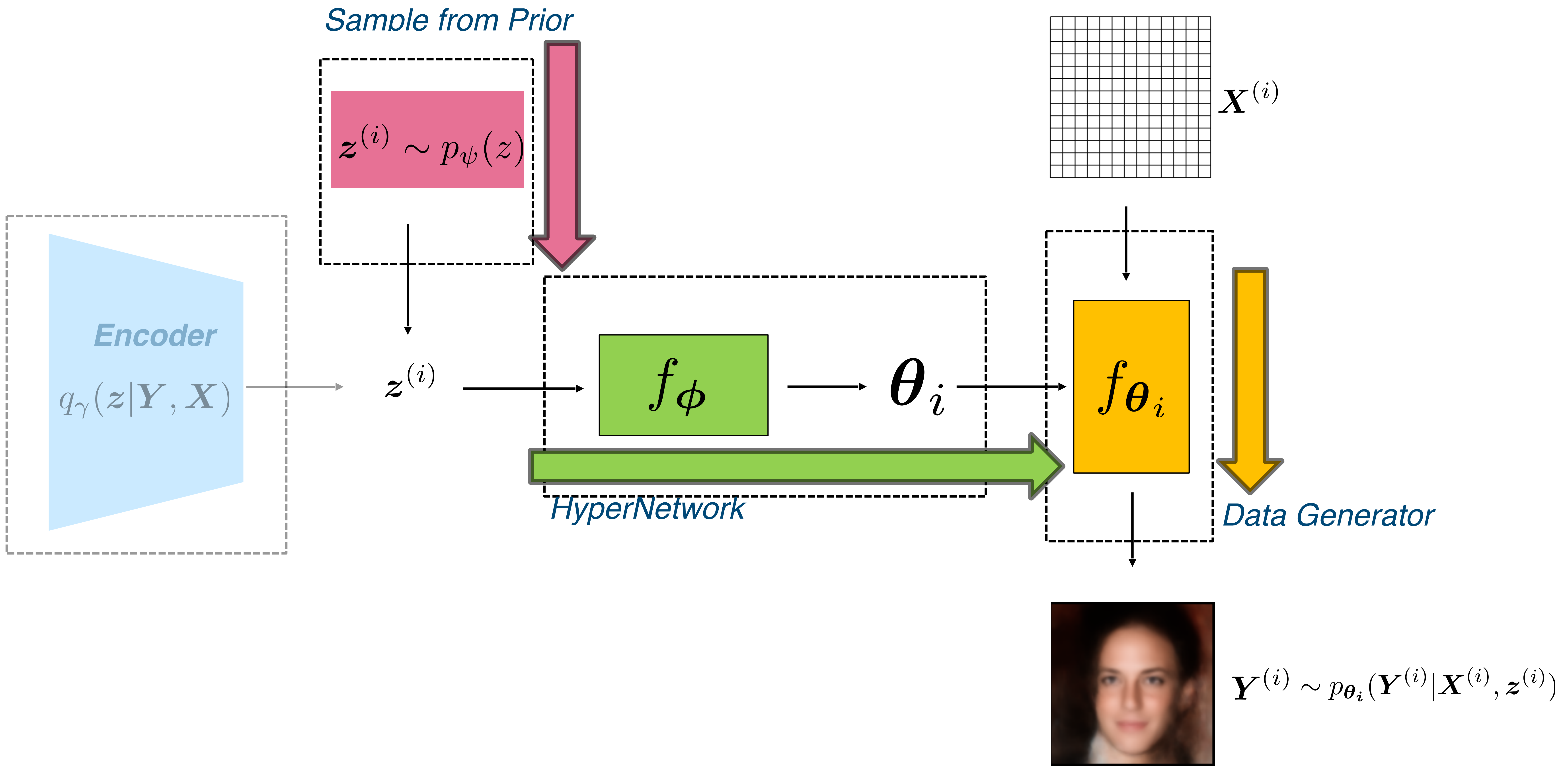
VAMoH

Image Completion



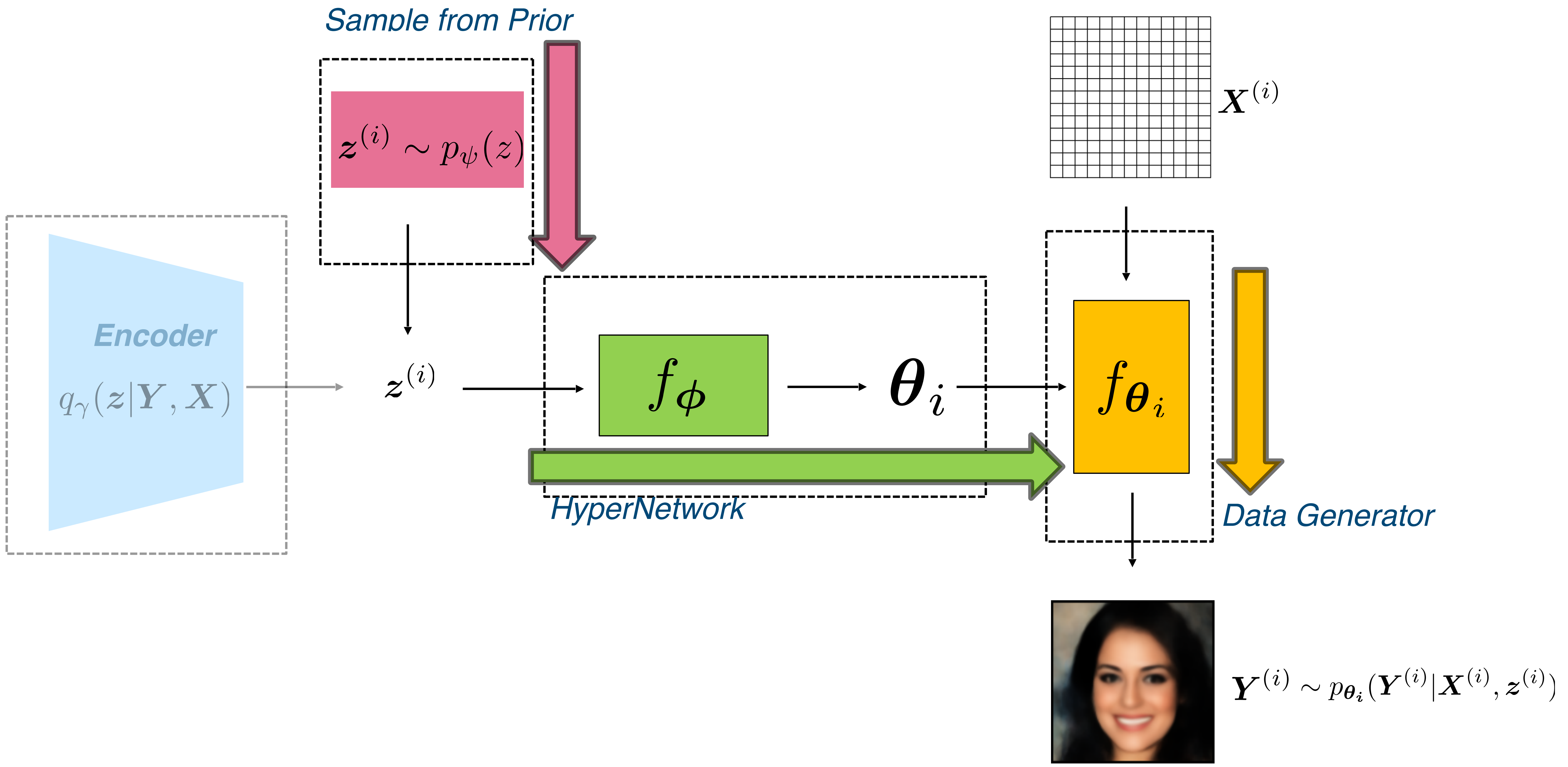
VAMoH

Image Generation



VAMoH

Image Generation



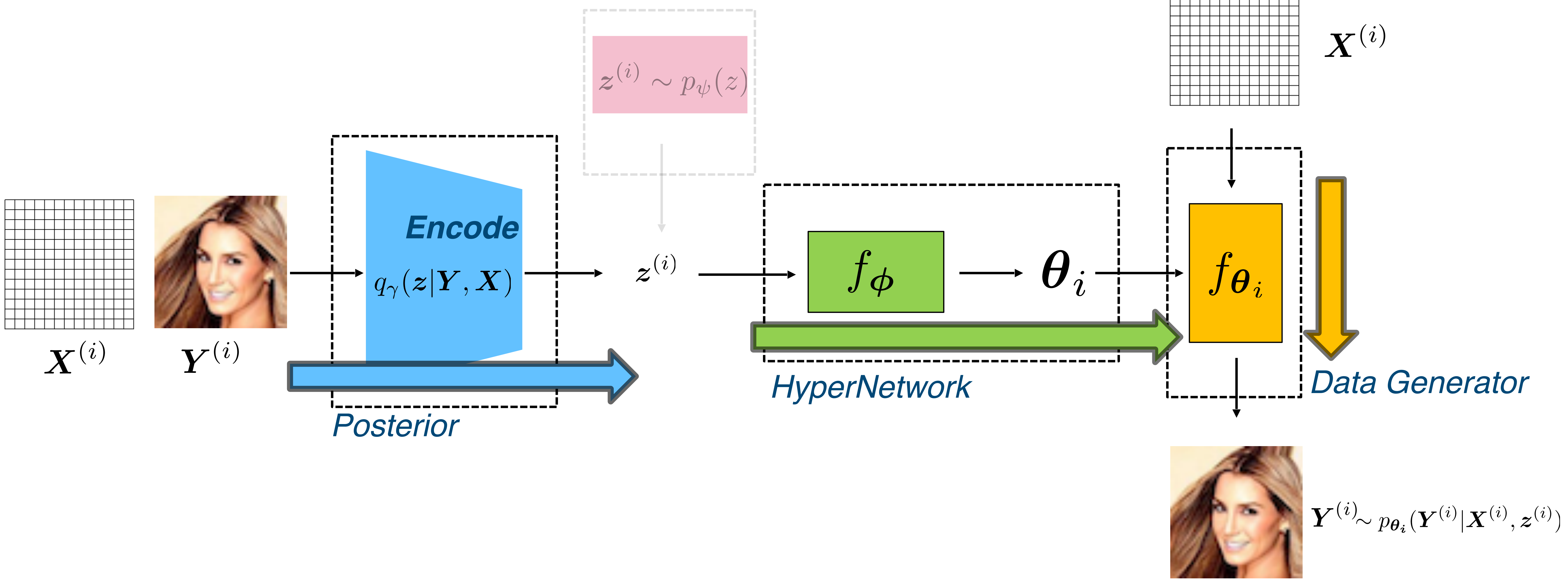
VAMoH

Optimization

How to learn all these steps end-to-end from data?

VAMoH

Optimization



Generative model:

$$p(Y, z \mid X) = p_\theta(Y \mid X, z)p_\psi(z)$$

Aim: Learning latent variables z given data (intractable)

$$p(z|Y,X)$$

How: Learn an approximation

$$q_\gamma(z|Y,X) \approx p(z|Y,X)$$

VAMoH

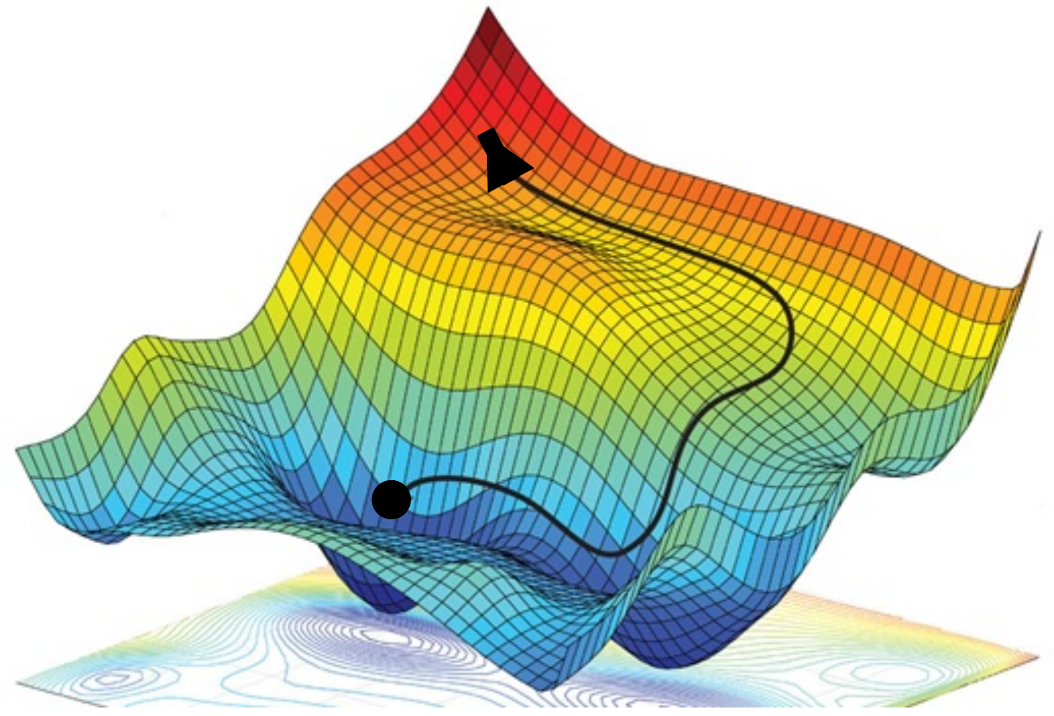
Optimization

- For a single data sample $(\boldsymbol{X}, \boldsymbol{Y})$

$$\max_{\phi,\gamma} \mathcal{L}(\phi, \gamma; \boldsymbol{Y}, \boldsymbol{X}) = \underbrace{\boxed{\max_{\phi,\gamma} \mathbb{E}_{q_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{Y},\boldsymbol{X})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z})]}}_{\text{Reconstruction}} - \underbrace{\boxed{D_{KL}(q_\gamma(\boldsymbol{z}|\boldsymbol{Y}, \boldsymbol{X}) \| p_\psi(\boldsymbol{z}))}}_{\text{Regularization}}$$

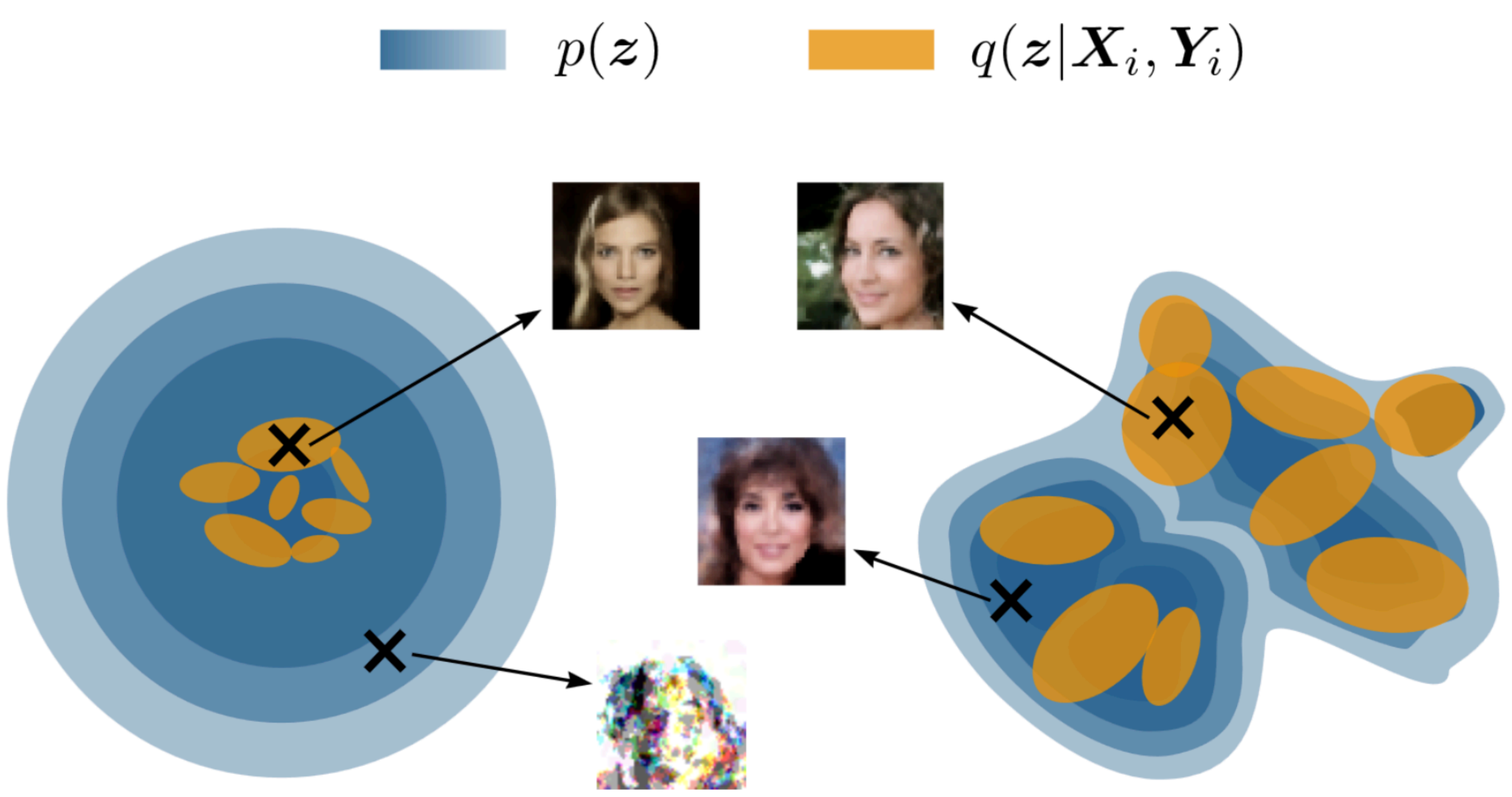
- For all samples in our dataset $\left(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}\right), i \in [N]$

$$\max_{\phi,\gamma} \sum_{i=1}^{N} \mathcal{L}(\phi, \gamma; \boldsymbol{Y}^{(i)}, \boldsymbol{X}^{(i)})$$



VAMoH

‘*Holes*’ problem



Regularization Term:

$$\min_{\gamma} D_{KL}\left(q_\gamma(\boldsymbol{z} \mid \boldsymbol{Y}, \boldsymbol{X}) \| p_\psi(\boldsymbol{z})\right)$$

→

We need to align the approximate posterior with the prior.

$p_\psi(\boldsymbol{z}) \quad q_\gamma(z)$

→

Problem:

If the prior is too simple, it hinders generation quality.

→

Solution:

Learn a more complex $p_\psi(z)$

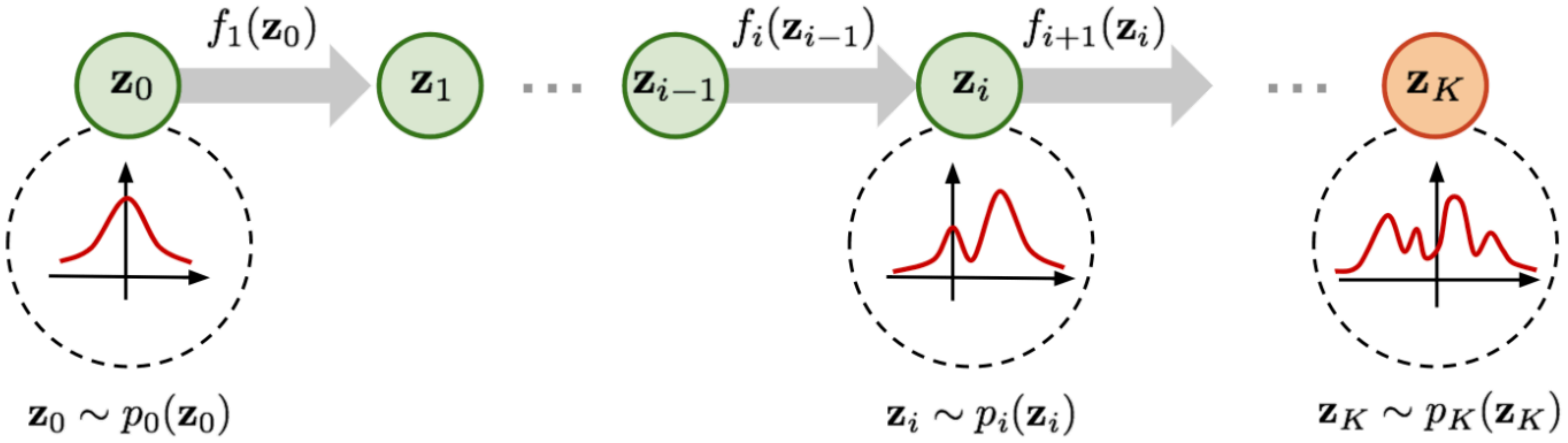
$$\min_{\gamma,\psi} D_{KL}(q_\gamma(\boldsymbol{z}|\boldsymbol{Y}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{z}))$$

VAMoH

Flow-based prior

- More expressive prior using RealNVP (Real-valued, Non-Volume Preserving) Flow.

[23]

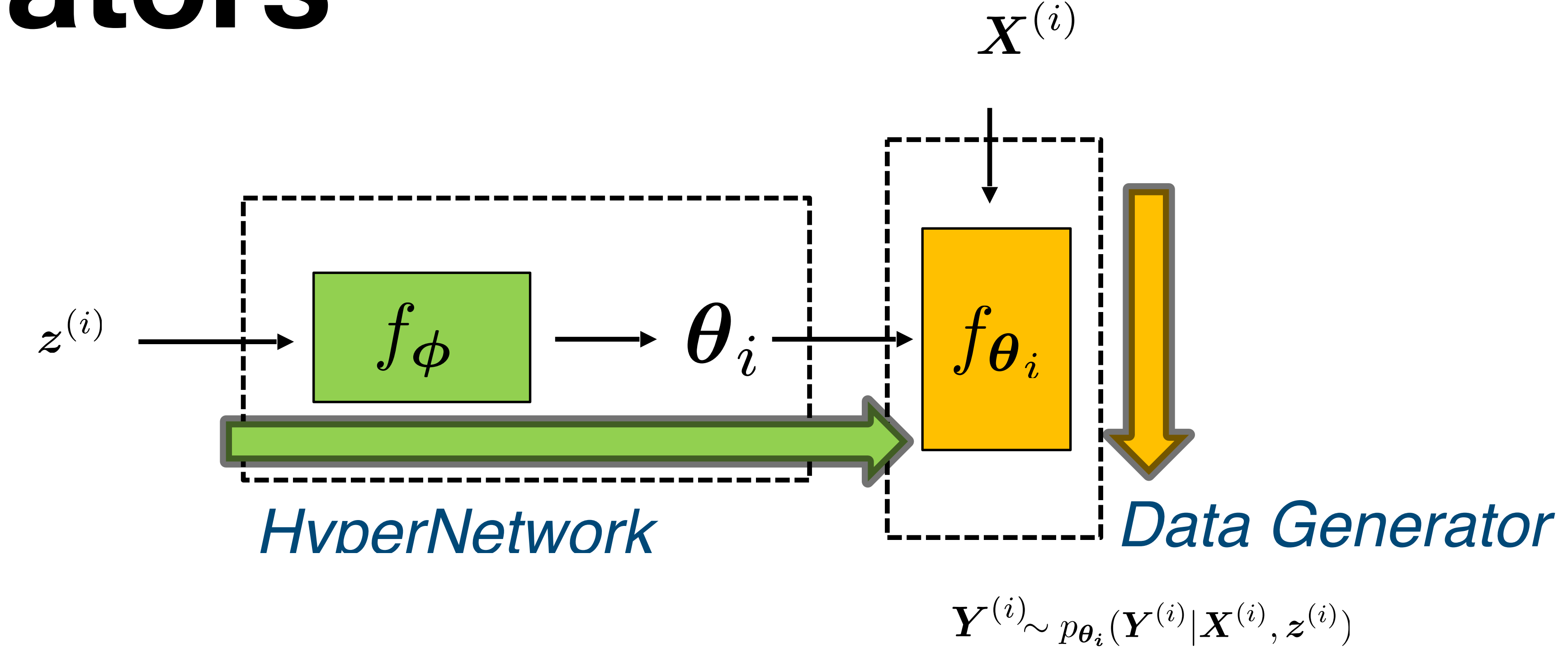


$$\boldsymbol{z}^{(i)} \sim p_\psi(z)$$

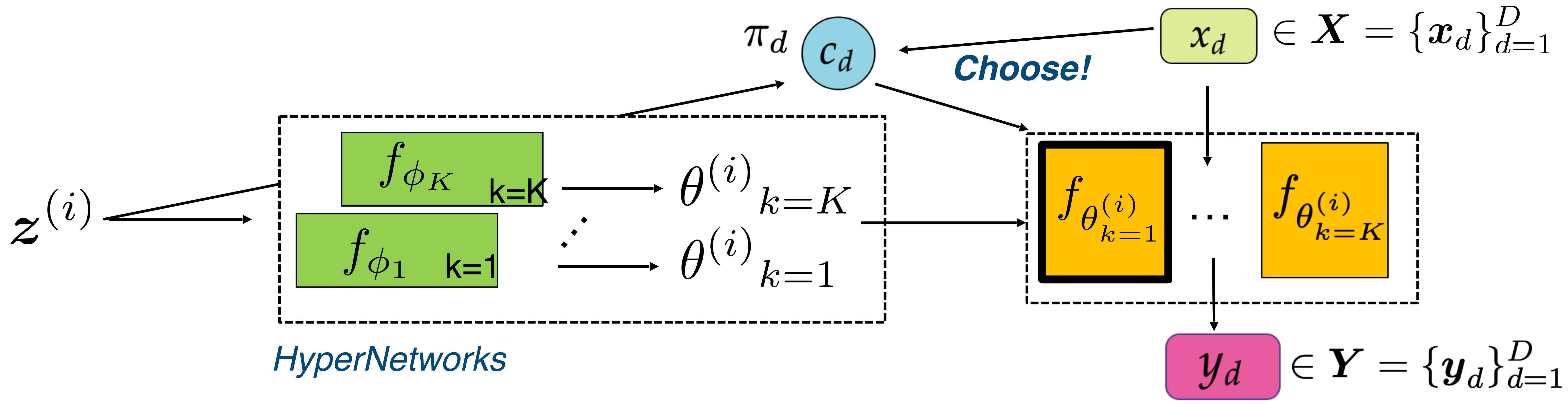
VAMoH

Mixture of HyperGenerators

Single HyperGenerator



Mixture of HyperGenerators



VAMoH

Mixture of HyperGenerators

Image Reconstruction with Mixture of HyperGenerators

VAMoH

ELBO

- For a single data sample $(\boldsymbol{X}, \boldsymbol{Y})$

$$\mathcal{L}(\boldsymbol{Y}, \boldsymbol{X}; \psi, \phi, \boldsymbol{\gamma}) = \sum_{d=1}^{D} \mathbb{E}_{q_{\gamma_z}(\boldsymbol{z}|\boldsymbol{Y},\boldsymbol{X})} \left[\sum_{k=1}^{K} \log p_{\boldsymbol{\theta}_k} \left(\boldsymbol{y}_d \mid \boldsymbol{x}_d \right) \cdot \pi_{dk} \right] - D_{KL} \left(q_{\gamma_z}(\boldsymbol{z} \mid \boldsymbol{X}, \boldsymbol{Y}) \| p_{\psi_z}(\boldsymbol{z}) \right)$$

$$- D_{KL} \left(q_{\gamma_c}(\boldsymbol{C} \mid \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{Y}) \| p_{\psi_c}(\boldsymbol{C} \mid \boldsymbol{z}, \boldsymbol{X}) \right)$$

- For all samples in our dataset $\left(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}\right), i \in [N]$

$$\max_{\phi,\gamma,\psi} \sum_{i=1}^{N} \mathcal{L}(\phi, \gamma, \psi; \boldsymbol{Y}^{(i)}, \boldsymbol{X}^{(i)})$$

Reconstruction the features of the observed pixels

KL of the continuous latent variable

KL of the discrete latent variable

Experiments

Baselines

Model	Approach	Training Procedure	Generation	Reconstruction, Imputation, Super Resolution
GASP (2021) [5]	GAN	Minimax	Forward Pass	❌

Experiments

Baselines

Model	Approach	Training Procedure	Generation	Reconstruction, Imputation, Super Resolution
GASP (2021) [5]	GAN	Minimax	Forward Pass	✗
Functa (2022) [6]	Flow-based	Bilevel optimization	+ Extra Generative Model	Optimization procedure(s) per sample

Experiments

Baselines

Model	Approach	Training Procedure	Generation	Reconstruction, Imputation, Super Resolution
GASP (2021) [5]	GAN	Minimax	Forward Pass	
Functa (2022) [6]	Flow-based	Bilevel optimization	+ Extra Generative Model	Optimization procedure(s) per sample 

$$\min_{\phi} -\log p(\phi) + \lambda \sum_{i \in \mathcal{I}} \|f_{\phi}(\mathbf{x}_i) - \mathbf{f}_i\|_2^2$$

Experiments
Baselines

Model	Approach	Training Procedure	Generation	Reconstruction, Imputation, Super Resolution
GASP (2021) [5]	GAN	Minimax	Forward Pass	✗
Functa (2022) [6]	Flow-based	Bilevel optimization	+ Extra Generative Model	Optimization procedure(s) per sample
VaMoH (ours)	VAE-based	**Single optimization**	**Forward Pass**	**Forward pass**

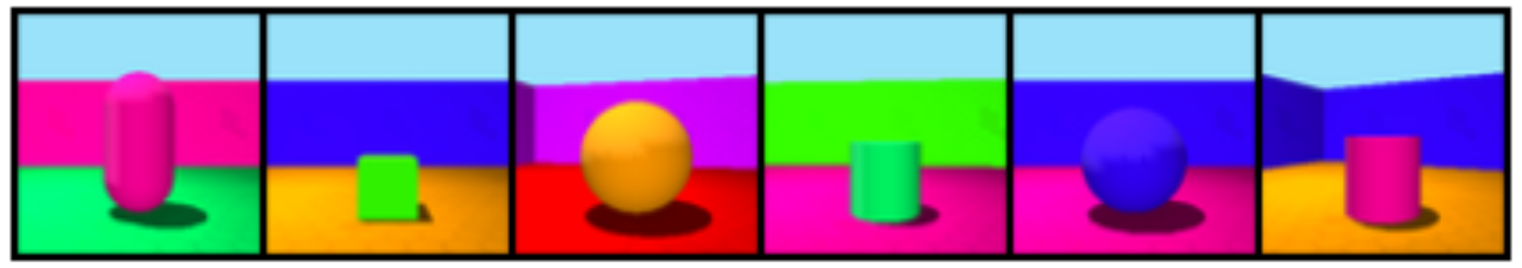
VAMoH provides a probabilistic generative model that is efficient, robust, and expressive for modeling distribution over functions.

Experiments

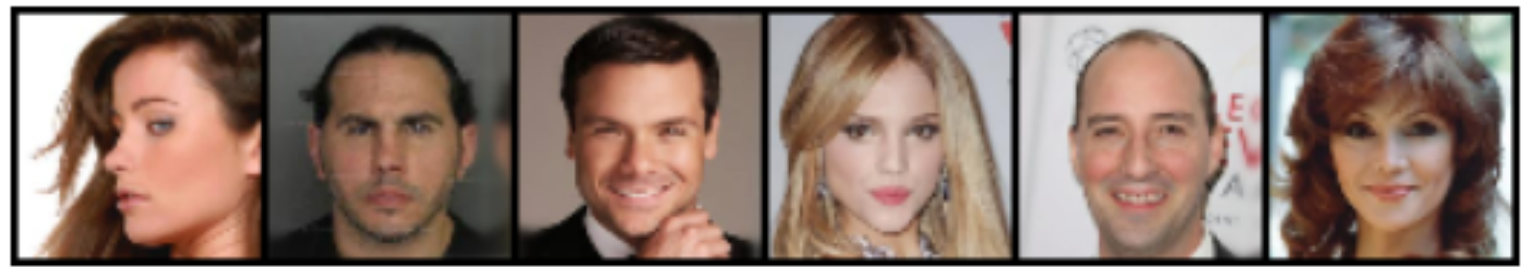
Datasets

PolyMNIST (28x28)

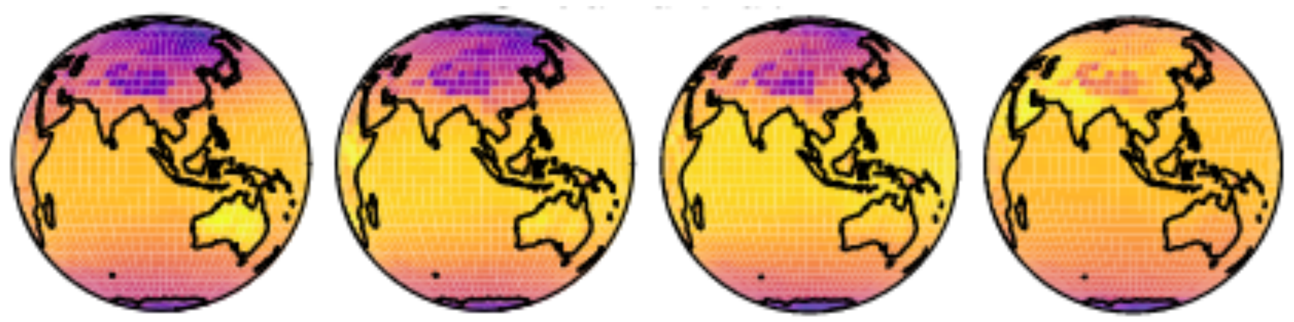
Shapes3D (64x64)



CelebA-HQ (64x64)



ERA5 (Polar)

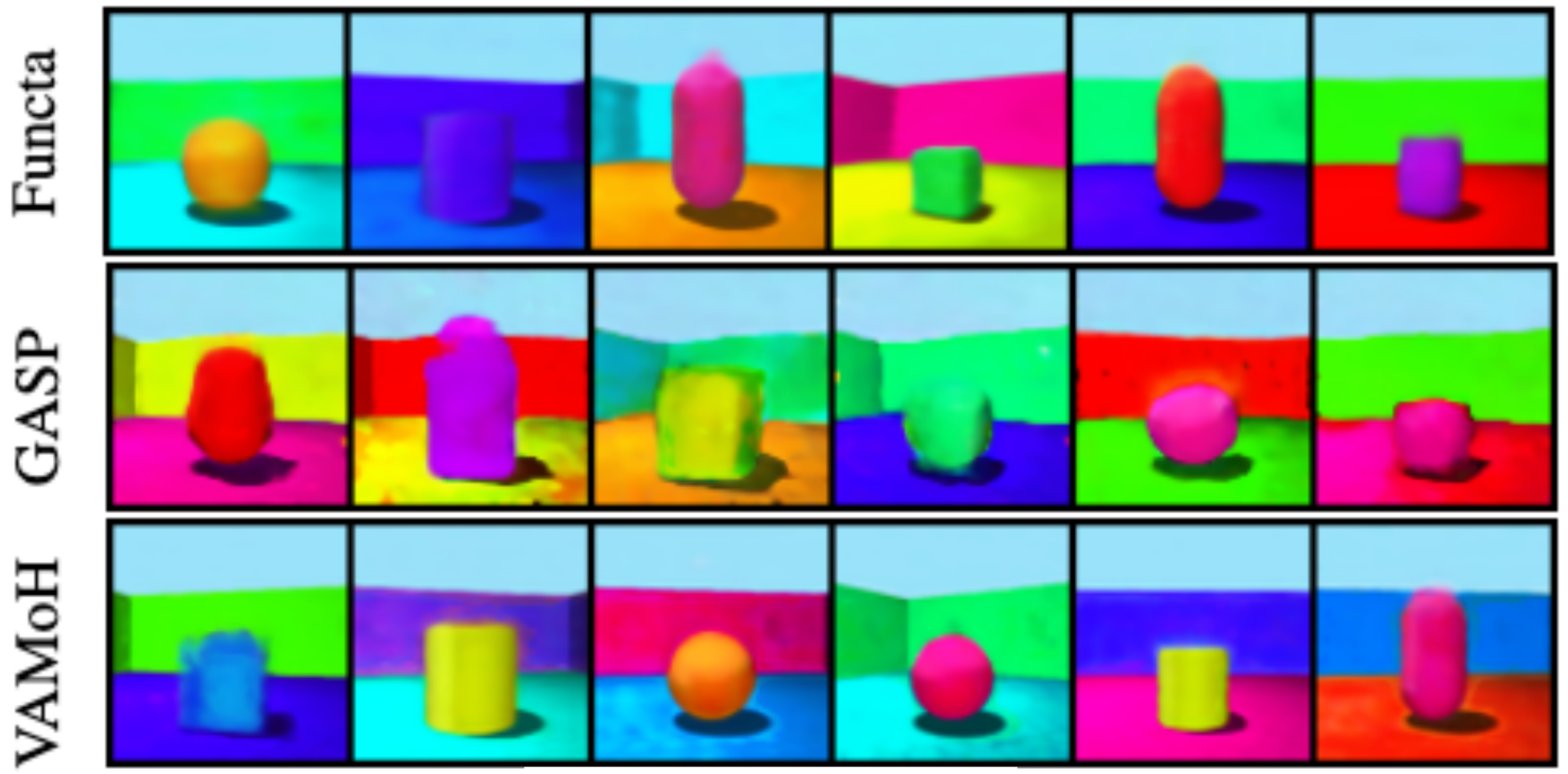


ShapeNET (Voxels)

Experiments

Generation

CelebA-HQ

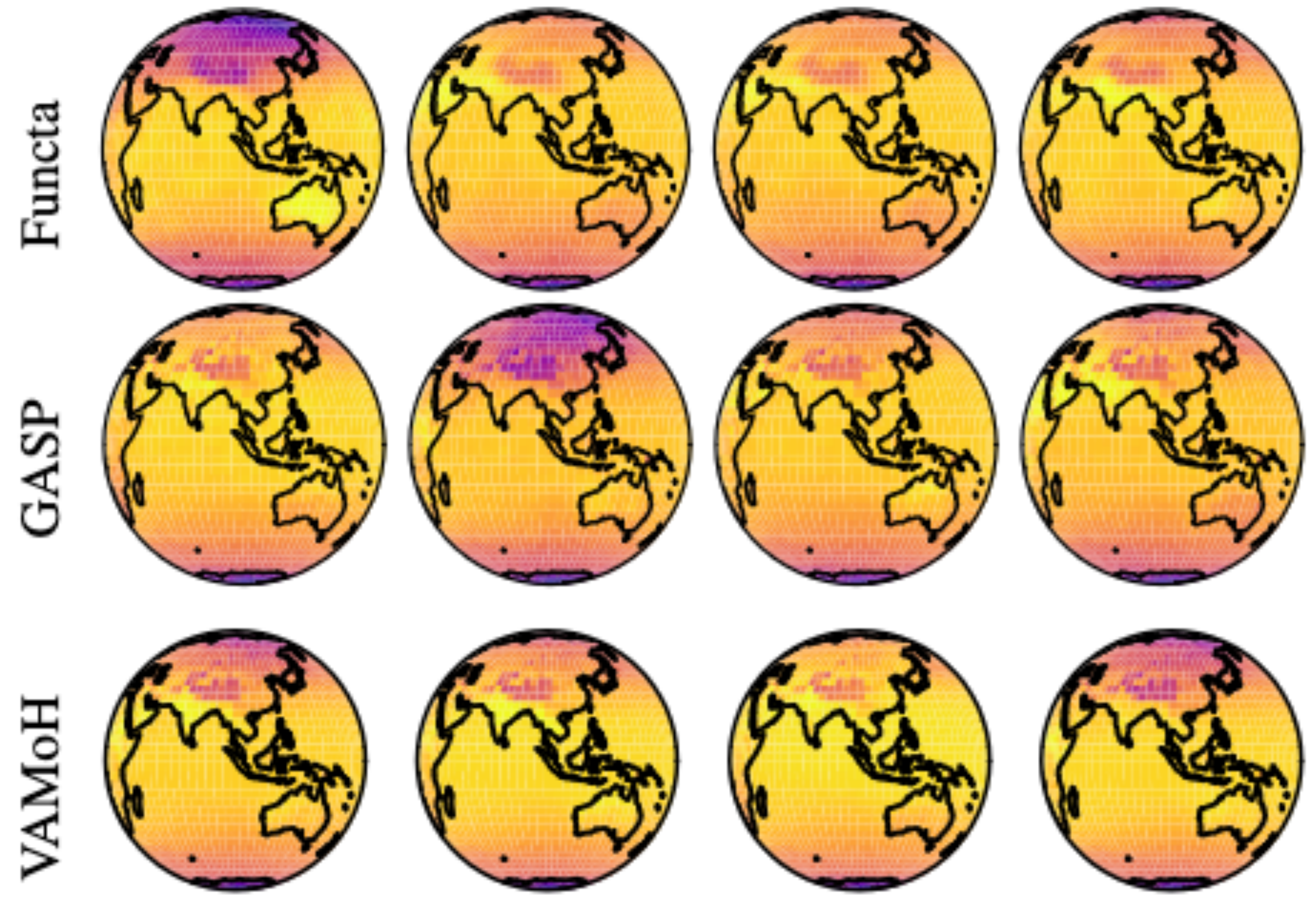


Shapes3D

Experiments

Generation

PolyMNIST



ERA5

ShapeNET

Experiments

Reconstructions

Experiments

Inference times

Table 2: Comparison of inference time (seconds) for reconstruction task of VaMoH and Functa. On the right-most two columns, we show the speed improvement of VaMoH compared to Functa (3) which is trained with 3 gradient steps as suggested in the original paper [Dupont et al., 2022b] and Functa (10) which is trained with 10 gradient step to obtain the results of Functa depicted in Figures 16,17. Please note that these experiments are run on the same GPU device.

	Model Inference Time (secs)			Speed Improvement	
Dataset	VaMoH	Functa (3)	Functa (10)	vs. Functa (3)	vs. Functa (10)
POLYMNIST	**0.00453**	0.01648	0.05108	**x 3.64**	**x 11.28**
SHAPES3D	**0.00536**	0.01759	0.05480	**x 3.28**	**x 10.22**
CELEBA HQ	**0.00757**	0.01733	0.05381	**x 2.29**	**x 7.11**
ERA5	**0.00745**	0.01899	0.05932	**x 2.55**	**x 7.96**
SHAPENET	**0.00689**	0.02095	0.06576	**x 3.04**	**x 9.54**

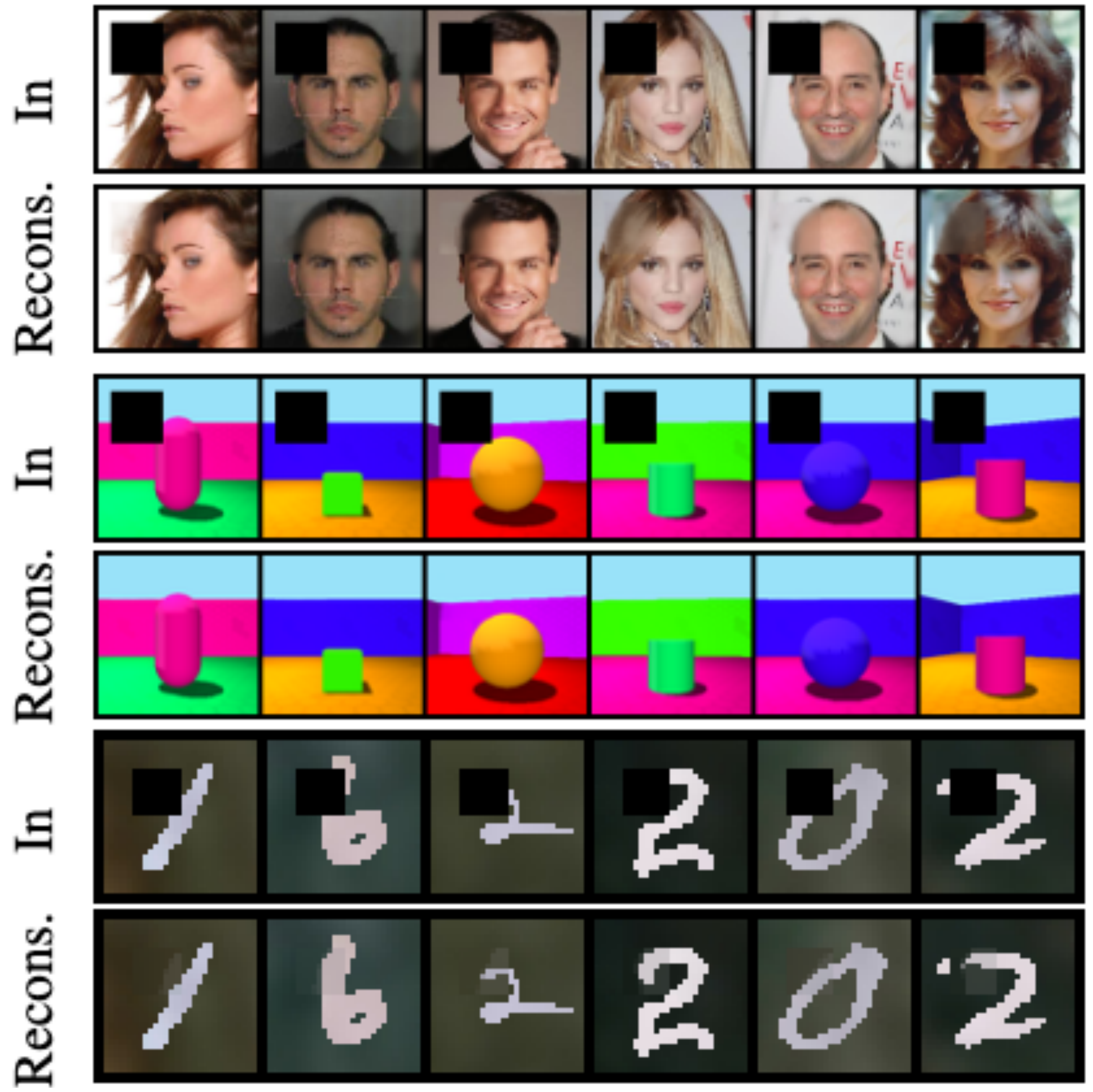
Reconstruction

	Model Inference Time (secs)			Speed Improvement	
Dataset	VaMoH	Functa (3)	Functa (10)	vs. Functa (3)	vs. Functa (10)
POLYMNIST	**0.00455**	0.01649	0.05109	**x 3.62**	**x 11.23**
SHAPES3D	**0.00544**	0.01768	0.05489	**x 3.25**	**x 10.09**
CELEBA HQ	**0.00833**	0.01729	0.05377	**x 2.08**	**x 6.46**
ERA5	**0.00790**	0.01997	0.06030	**x 2.53**	**x 7.63**
SHAPENET	**0.01440**	0.02089	0.06569	**x 1.45**	**x 4.56**

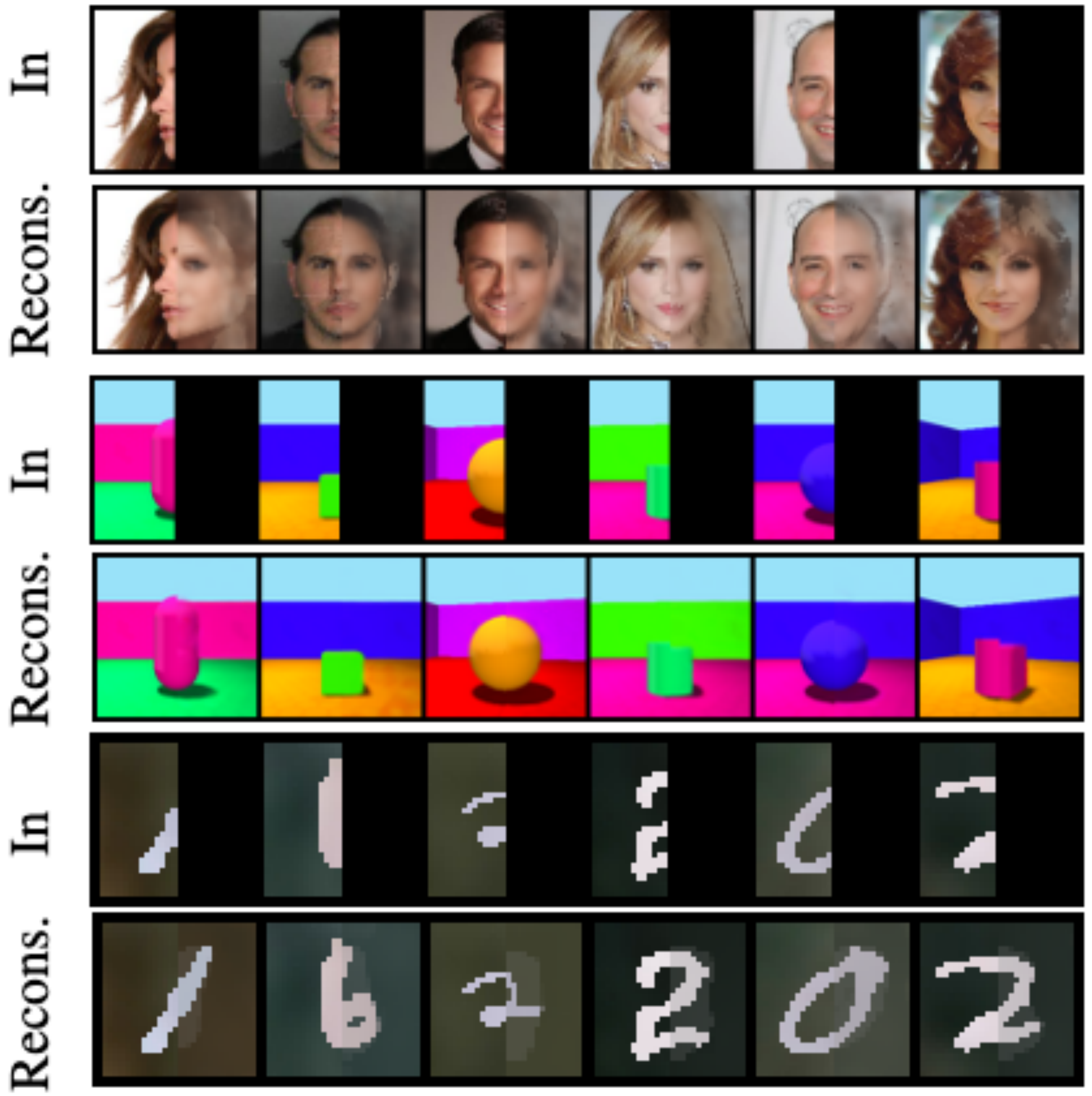
Super-reconstruction

Experiments

Image completion



Missing a patch (in-painting)



Missing half of the image

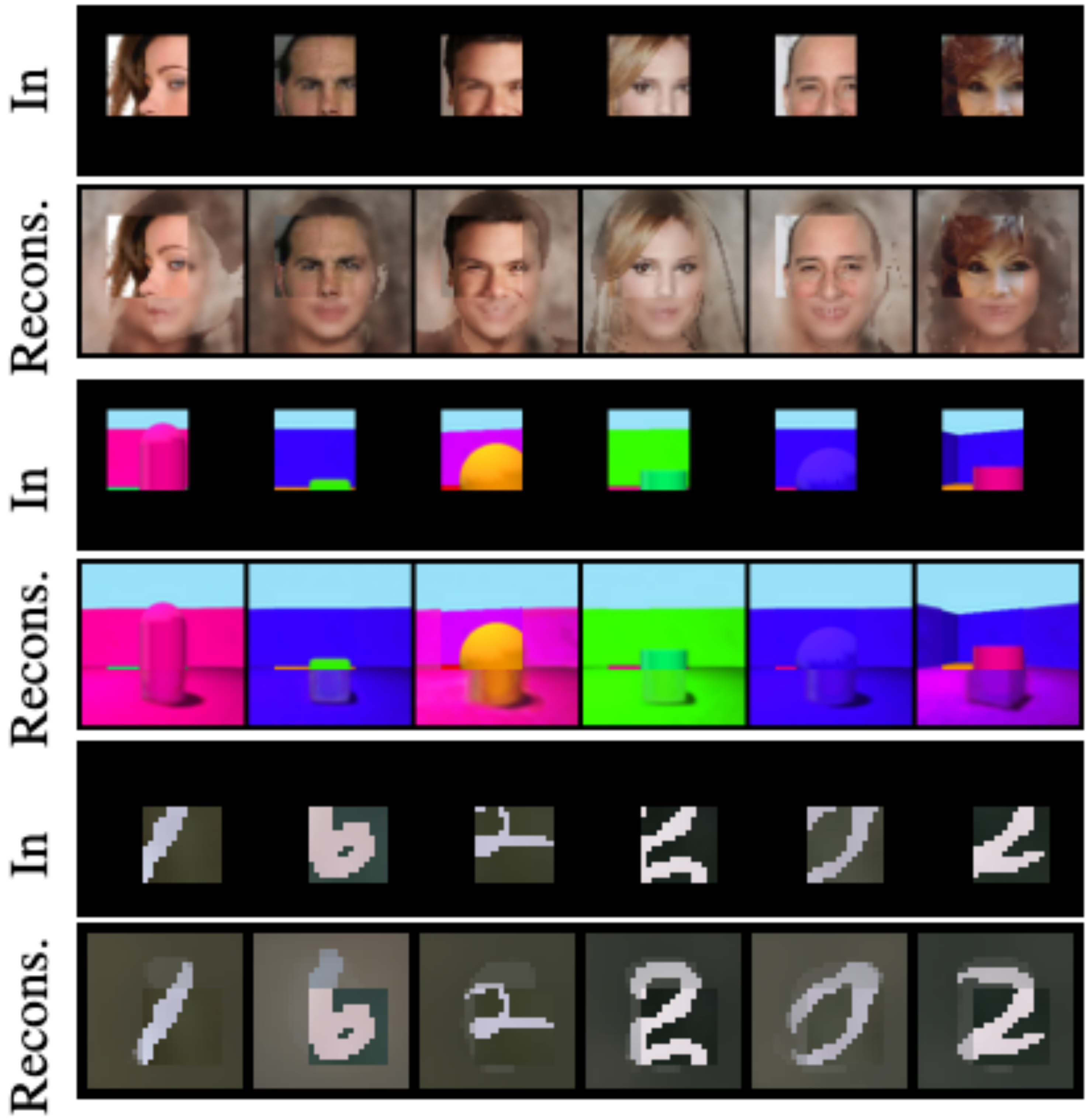


Image out-painting

Conclusion

Thanks to learning distributions of functions, our proposed VAMoH can easily perform:

- Generation.
- Reconstruction.
- Conditional generation.
- Super resolution (interpolation).

While being:

✓ Robust to partially observed data.

✓ Expressive for generating high-quality data.

✓ Efficient in terms of inference.

Further details

VARIATIONAL MIXTURE OF HYPERGENERATORS FOR LEARNING DISTRIBUTIONS OVER FUNCTIONS

Batuhan Koyuncu*
Saarland University
Saarbrücken, Germany

Pablo Sánchez-Martín
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Ignacio Peis
Universidad Carlos III de Madrid
Madrid, Spain

Pablo M. Olmos
Universidad Carlos III de Madrid
Madrid, Spain

Isabel Valera
Saarland University
Saarbrücken, Germany

[Paper]

[25] Koyuncu et at., 2023

References

- [1] 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.

- [2] Caterini, A. L., Doucet, A., & Sejdinovic, D. (2018). Hamiltonian variational auto-encoder. *Advances in Neural Information Processing Systems, 31.*

- [3] Salimans, T., Kingma, D., & Welling, M. (2015, June). Markov chain monte carlo and variational inference: Bridging the gap. In *International conference on machine learning* (pp. 1218-1226). PMLR.

- [4] 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.

- [5] Dupont, E., Whye Teh, Y. & Doucet, A.. (2022). Generative Models as Distributions of Functions. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research* 151:2989-3015.

- [6] Dupont, E., Kim, H., Eslami, S. A., Rezende, D. J., & Rosenbaum, D. (2022, June). From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning* (pp. 5694-5725). PMLR.

References

- [7] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- [8] Cremer, C., Li, X., & Duvenaud, D. (2018, July). Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning* (pp. 1078-1086). PMLR.
- [9] Bernardo, J. M. (1979). Expected information as expected utility. *the Annals of Statistics*, 686-690.
- [10] 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*.
- [11] 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.
- [12] Child, R. (2020). Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*.

References

- [13] Nazabal, A., Olmos, P. M., Ghahramani, Z., & Valera, I. (2020). Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, *107*, 107501.
- [14] 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.
- **[15] 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.**
- [16] Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical review E*, *69*(6), 066138.
- [17] Gong, W., Li, Y., & Hernández-Lobato, J. M. (2020). Sliced kernelized Stein discrepancy. *arXiv preprint arXiv:2006.16531*.
- [18] Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

References

- [19] Betancourt, M., & Girolami, M. (2015). Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, *79*(30), 2-4.
- [20] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, *33*, 7462-7473.
- [21] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4460-4470).
- [22] Sitzmann, V., Zollhöfer, M., & Wetzstein, G. (2019). Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, *32*.
- [23] Ha, D., Dai, A. M., & Le, Q. V. HyperNetworks. In International Conference on Learning Representations.
- [24] Wu, W., Qi, Z., & Fuxin, L. (2019). Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (pp. 9621-9630).
- [25] Koyuncu, B., Sanchez-Martin, P., Peis, I., Olmos, P. M., & Valera, I. (2023). Variational Mixture of HyperGenerators for Learning Distributions Over Functions. In Proceedings of the 40th International Conference on Machine Learning, 2023.

Thank you!

ipeis@tsc.uc3m.es