DeepMind

Unbiased Gradient Estimation for Variational Auto-Encoders using Coupled Markov Chains

Francisco J. R. Ruiz, Michalis K. Titsias,
Taylan Cemgil, Arnaud Doucet

20/10/2020

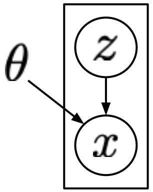
DeepMind

Motivation and Goal

Motivation

- Stochastic gradient descent (SGD) is a powerful tool in ML
- SGD obtains and follows (unbiased) estimates of the gradient
- For some models, these estimates are not available

Example: Variational Autoencoder



- The VAE is a deep probabilistic model

$$\mathcal{L}(\theta) := \log p_\theta(x) = \log\left(\int p_\theta(x,z)dz\right)$$

- Its gradient is intractable

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\theta(z\,|\,x)}\left[\nabla_\theta \log p_\theta(x,z)\right]$$

but it can be written as an expectation

Example: Energy-Based Model

- The EBM is a deep model with arbitrarily complex energy function

$$p_\theta(x) = \frac{\exp(E_\theta(x))}{Z_\theta}, \quad Z_\theta = \int \exp(E_\theta(x))\mathrm{d}x$$

- Its gradient is intractable

$$\nabla_\theta \log p_\theta(x) = \nabla_\theta E_\theta(x) - \mathbb{E}_{p_\theta(x')}\left[\nabla_\theta E_\theta(x')\right]$$

but it can be written as an expectation

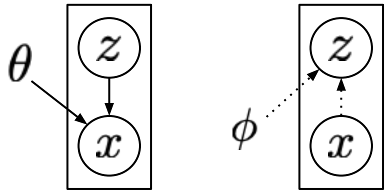
This Talk

- Form **unbiased gradient estimates** for complex models
- Main ideas: Extended latent space, MCMC couplings
- Focus on the VAE and show that unbiased gradients can boost the predictive performance

DeepMind

Preliminaries

Review: VAE / IWAE



- The VAE log-likelihood

$$\mathcal{L}(\theta) := \log p_\theta(x) = \log\left(\int p_\theta(x, z)\mathrm{d}z\right)$$

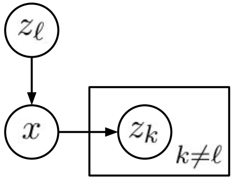
- Optimize the ELBO instead (a lower bound)

$$\mathcal{L}_{\mathrm{ELBO}}(\theta, \phi) = \mathbb{E}_{q_\phi(z\,|\,x)}\left[\log w_{\theta,\phi}(z)\right] \qquad w_{\theta,\phi}(z) := \frac{p_\theta(x, z)}{q_\phi(z\,|\,x)}$$

- Or optimize the IWAE (a tighter lower bound)

$$\mathcal{L}_{\mathrm{IWAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(z_{1:K}\,|\,x)}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K} w_{\theta,\phi}(z_k)\right)\right]$$

The IWAE as an ELBO in an Augmented Space



- Define the extended model

$$p_{\theta,\phi}(x, z_{1:K}, \ell) = \frac{1}{K} p_\theta(z_\ell) p_\theta(x \mid z_\ell) \prod_{\substack{k=1 \\ k \neq \ell}}^{K} q_\phi(z_k \mid x)$$

- Define the importance weights

$$w_{\theta,\phi}^{(k)} = w_{\theta,\phi}(z_k), \quad \widetilde{w}_{\theta,\phi}^{(k)} = \frac{w_{\theta,\phi}^{(k)}}{\sum_{k'=1}^{K} w_{\theta,\phi}^{(k')}}.$$

- Define the variational distribution on the extended space

$$q_{\theta,\phi}(z_{1:K}, \ell) = \text{Categorical}\left(\ell \mid \widetilde{w}_{\theta,\phi}^{(1)}, \ldots, \widetilde{w}_{\theta,\phi}^{(K)}\right) \prod_{k=1}^{K} q_\phi(z_k \mid x)$$

- The ELBO coincides with the IWAE bound

$$\mathcal{L}_{\text{ELBO}}^{\text{augmented}}(\theta, \phi) = \mathbb{E}_{q_{\theta,\phi}(z_{1:K}, \ell)}\left[\log \frac{p_{\theta,\phi}(x, z_{1:K}, \ell)}{q_{\theta,\phi}(z_{1:K}, \ell)}\right] = \mathbb{E}_{q_{\theta,\phi}(z_{1:K})}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K} w_{\theta,\phi}^{(k)}\right)\right]$$

The Roadmap to Unbiased Estimation

- Both the ELBO and IWAE are biased approximators of the gradient of the log-likelihood

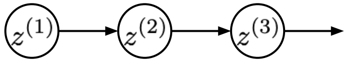
- If we could sample from the posterior, we could easily form an unbiased estimator

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\theta(z \mid x)} \left[\nabla_\theta \log p_\theta(x, z)\right]$$

(but cannot sample exactly in practice)

- MCMC couplings provide unbiased estimators by design

MCMC Couplings

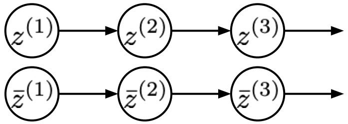


- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \mid z)$ that targets $\pi(z)$

MCMC Couplings



- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \mid z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \mid z, \bar{z})$

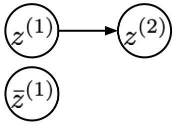
MCMC Couplings

- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \mid z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \mid z, \bar{z})$
  - Initialize the first chain $z^{(1)} \sim \mathcal{K}(z \mid z^{(0)})$

MCMC Couplings

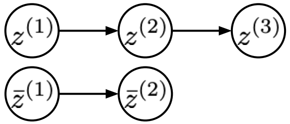


- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \,|\, z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \,|\, z, \bar{z})$
  - Initialize the first chain $z^{(1)} \sim \mathcal{K}(z \,|\, z^{(0)})$
  - Then sample both chains from the joint kernel $z^{(t+1)}, \bar{z}^{(t)} \sim \mathcal{K}(z, \bar{z} \,|\, z^{(t)}, \bar{z}^{(t-1)})$

MCMC Couplings

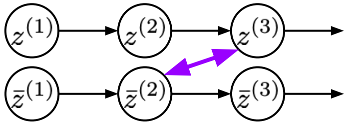


- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \,|\, z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \,|\, z, \bar{z})$
  - Initialize the first chain $z^{(1)} \sim \mathcal{K}(z \,|\, z^{(0)})$
  - Then sample both chains from the joint kernel $z^{(t+1)}, \bar{z}^{(t)} \sim \mathcal{K}(z, \bar{z} \,|\, z^{(t)}, \bar{z}^{(t-1)})$

MCMC Couplings

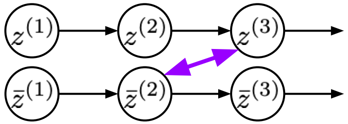


- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \mid z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \mid z, \bar{z})$
  - Initialize the first chain $z^{(1)} \sim \mathcal{K}(z \mid z^{(0)})$
  - Then sample both chains from the joint kernel $z^{(t+1)}, \bar{z}^{(t)} \sim \mathcal{K}(z, \bar{z} \mid z^{(t)}, \bar{z}^{(t-1)})$
- Define the **meeting time** $\tau = \inf\{t \geq 1 : z^{(t)} = \bar{z}^{(t-1)}\}$

MCMC Couplings



- Consider estimating an expectation of the form

$$H = \mathbb{E}_{\pi(z)}\left[h(z)\right]$$

- Typical MCMC approach: sample from a kernel $\mathcal{K}(\cdot \mid z)$ that targets $\pi(z)$
- Coupling MCMC: use **two** MCMC chains with the same stationary distribution that are *coupled*
  - There is a joint MCMC kernel $\mathcal{K}(\cdot, \cdot \mid z, \bar{z})$
  - Initialize the first chain $z^{(1)} \sim \mathcal{K}(z \mid z^{(0)})$
  - Then sample both chains from the joint kernel $z^{(t+1)}, \bar{z}^{(t)} \sim \mathcal{K}(z, \bar{z} \mid z^{(t)}, \bar{z}^{(t-1)})$
- Define the **meeting time** $\tau = \inf\{t \geq 1 : z^{(t)} = \bar{z}^{(t-1)}\}$
- Then, an unbiased estimator is

$$\mathbb{E}_{\pi(z)}\left[h(z)\right] \approx \hat{H}^{(t_0)}_{\text{Glynn}} \triangleq h(z^{(t_0)}) + \sum_{t=t_0+1}^{\tau-1} \left(h(z^{(t)}) - h(\bar{z}^{(t-1)})\right)$$

DeepMind

Unbiased Estimators on an Extended Space

Our Proposal

- Start with $\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\theta(z \mid x)} \left[\nabla_\theta \log p_\theta(x, z) \right]$
- Form the augmented model and augmented *proposal* distribution

$$p_{\theta,\phi}(x, z_{1:K}, \ell) = \frac{1}{K} p_\theta(z_\ell) p_\theta(x \mid z_\ell) \prod_{\substack{k=1 \\ k \neq \ell}}^{K} q_\phi(z_k \mid x)$$

$$q_{\theta,\phi}(z_{1:K}, \ell) = \text{Categorical}\left(\ell \mid \widetilde{w}_{\theta,\phi}^{(1)}, \ldots, \widetilde{w}_{\theta,\phi}^{(K)}\right) \prod_{k=1}^{K} q_\phi(z_k \mid x)$$

- Run a coupled MCMC kernel on the extended space, targeting the augmented posterior
  - How to form the kernel?

PIMH Algorithm (Non-Coupled Version)

Algorithm 1: Particle independent Metropolis-Hastings (PIMH) kernel, $\mathcal{K}_{\text{PIMH}}(\cdot, \cdot \mid z_{1:K}, \ell)$

Input: Current state of the chain, $(z_{1:K}, \ell)$

Output: Next state of the chain

```
Sample a candidate (z*_{1:K}, ℓ*) ~ q_{θ,φ}(·, ·)
Sample u ~ U([0, 1])
if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) then
    Return (z*_{1:K}, ℓ*)                    ▷ the proposal is accepted
else
    Return (z_{1:K}, ℓ)                      ▷ the proposal is rejected
end
```

PIMH Algorithm (Coupled Version)

Algorithm 4: Coupled PIMH kernel, $\mathcal{K}_{\text{C-PIMH}}((\cdot,\cdot),(\cdot,\cdot) \mid (z_{1:K},\ell),(\bar{z}_{1:K},\bar{\ell}))$

Input: Current state of both chains, $(z_{1:K},\ell)$ and $(\bar{z}_{1:K},\bar{\ell})$

Output: New state of both chains

```
Sample (z*_{1:K}, ℓ*) ~ q_{θ,φ}(·,·)
Sample u ~ U([0,1])
if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z*_{1:K}, ℓ*))          ▷ both chains accept the proposal
else if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u > α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z̄_{1:K}, ℓ̄))            ▷ the first chain accepts the proposal
else if u > α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z_{1:K}, ℓ), (z*_{1:K}, ℓ*))            ▷ the second chain accepts the proposal
else
    Return ((z_{1:K}, ℓ), (z̄_{1:K}, ℓ̄))              ▷ neither chain accepts the proposal
end
```

PIMH Algorithm (Coupled Version)

Algorithm 4: Coupled PIMH kernel, $\mathcal{K}_{\text{C-PIMH}}((\cdot,\cdot),(\cdot,\cdot) \mid (z_{1:K},\ell),(\bar{z}_{1:K},\bar{\ell}))$

Input: Current state of both chains, $(z_{1:K},\ell)$ and $(\bar{z}_{1:K},\bar{\ell})$

Output: New state of both chains

```
Sample (z*_{1:K}, ℓ*) ~ q_{θ,φ}(·,·)
Sample u ~ U([0,1])
if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z*_{1:K}, ℓ*))          ▷ both chains accept the proposal
else if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u > α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z̄_{1:K}, ℓ̄))            ▷ the first chain accepts the proposal
else if u > α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z_{1:K}, ℓ), (z*_{1:K}, ℓ*))            ▷ the second chain accepts the proposal
else
    Return ((z_{1:K}, ℓ), (z̄_{1:K}, ℓ̄))              ▷ neither chain accepts the proposal
end
```

PIMH Algorithm (Coupled Version)

Algorithm 4: Coupled PIMH kernel, $\mathcal{K}_{\text{C-PIMH}}((\cdot,\cdot),(\cdot,\cdot) \mid (z_{1:K},\ell),(\bar{z}_{1:K},\bar{\ell}))$

Input: Current state of both chains, $(z_{1:K},\ell)$ and $(\bar{z}_{1:K},\bar{\ell})$

Output: New state of both chains

```
Sample (z*_{1:K}, ℓ*) ~ q_{θ,φ}(·,·)
Sample u ~ U([0,1])
if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z*_{1:K}, ℓ*))          ▷ both chains accept the proposal
else if u ≤ α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u > α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z*_{1:K}, ℓ*), (z̄_{1:K}, ℓ̄))          ▷ the first chain accepts the proposal
else if u > α(z*_{1:K}, ℓ* | z_{1:K}, ℓ) and u ≤ α(z*_{1:K}, ℓ* | z̄_{1:K}, ℓ̄) then
    Return ((z_{1:K}, ℓ), (z*_{1:K}, ℓ*))          ▷ the second chain accepts the proposal
else
    Return ((z_{1:K}, ℓ), (z̄_{1:K}, ℓ̄))          ▷ neither chain accepts the proposal
end
```

PIMH Algorithm (Coupled Version)

- After collecting samples, obtain the unbiased gradient estimator as

$$\nabla_\theta \log p_\theta(x) \approx h(z_{1:K}^{(t_0)}) + \sum_{t=t_0+1}^{\tau-1} \left(h(z_{1:K}^{(t)}) - h(\bar{z}_{1:K}^{(t-1)}) \right)$$

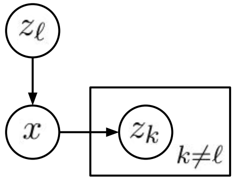
The function *h* is

$$h(z_{1:K}) = \sum_{k=1}^{K} \tilde{w}_{\theta,\phi}^{(k)} \nabla_\theta \log p_\theta(x, z_k)$$

Our Contributions

- Our MCMC algorithm is based on ISIR (rather than PIMH)
- We propose an extension of ISIR, called DISIR that *significantly* reduces the variance of the estimator
- We derive the sufficient conditions that guarantee an unbiased estimator of finite variance in finite time
- Our estimator is based on a lagged coupling estimator, which further reduces the variance

Importance Sampling in High-Dimensional Spaces

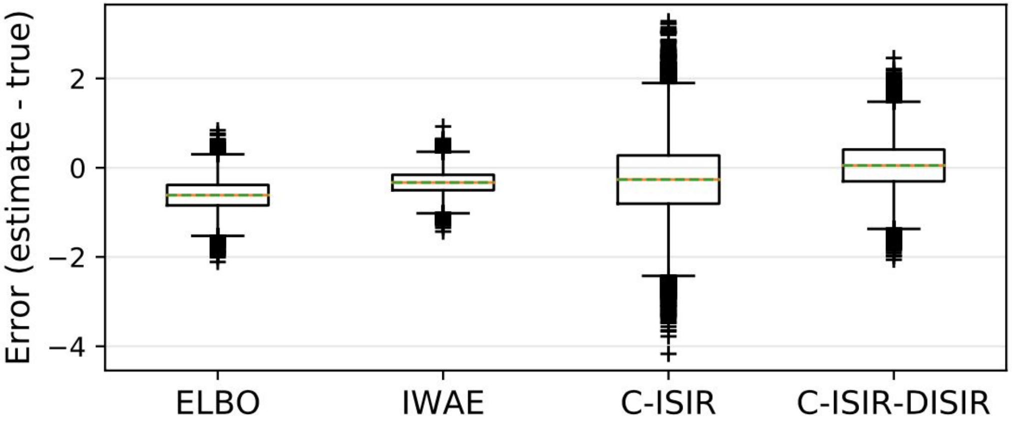


- IS typically fails in high dimensions, when one weight dominates the others
- We augment the dimensionality with *K-1* particles
  - So IS should perform poorly (and the MCMC chains would never meet)
  - However, performance actually improves with dimensionality (and meeting occurs earlier) as the model and proposals become closer to each other when *K* increases

DeepMind

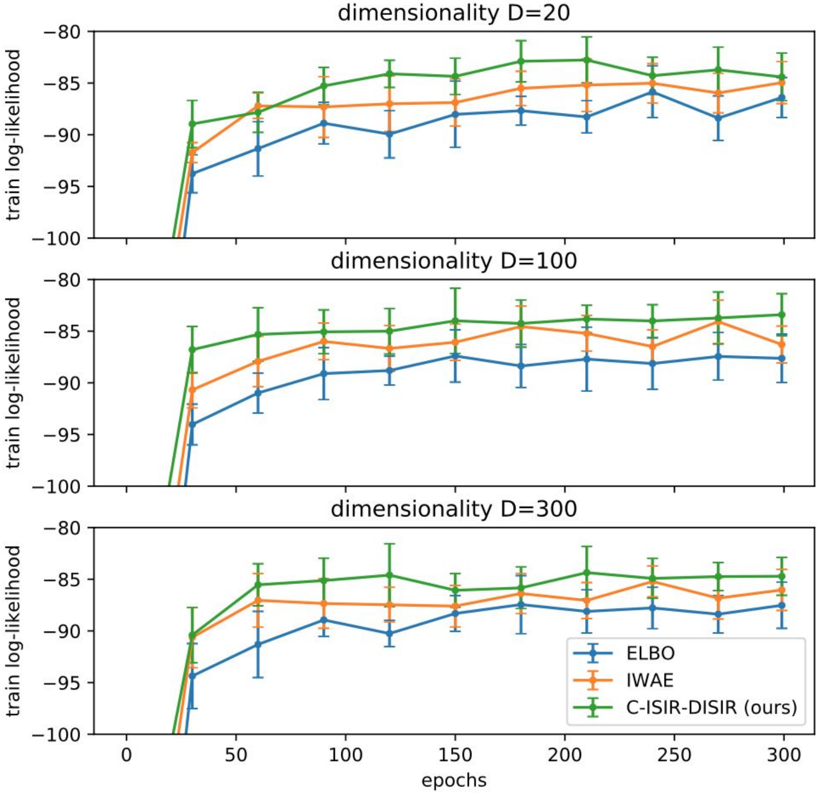
Experiments and Results

PPCA: Analysis of Unbiasedness



VAE on Binarized MNIST

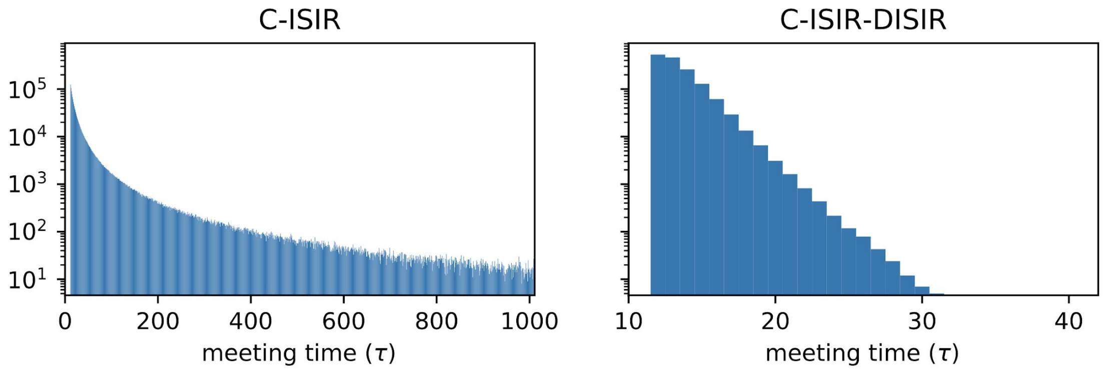
train log-likelihood



test log-likelihood

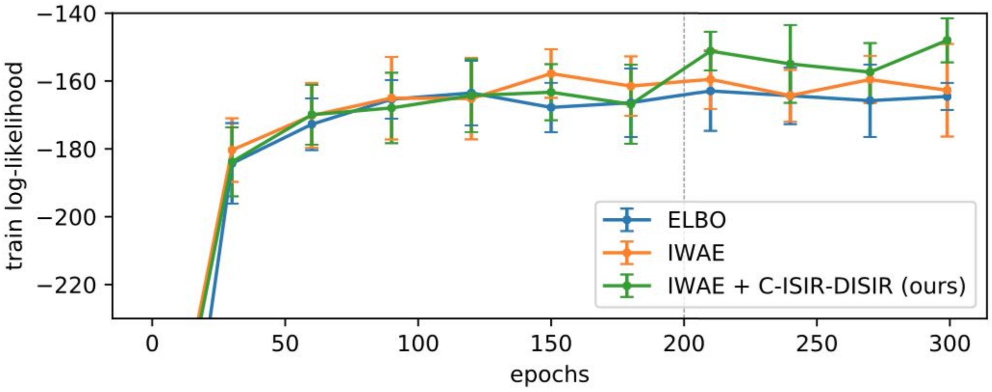
	dimensionality of z 20	100	300
ELBO	-90.05 ± 0.21	-89.96 ± 0.14	-90.63 ± 0.12
IWAE	-88.06 ± 0.08	-88.07 ± 0.06	-89.05 ± 0.08
C-ISIR-DISIR	$\mathbf{-87.29 \pm 0.08}$	$\mathbf{-86.75 \pm 0.10}$	$\mathbf{-88.10 \pm 0.08}$

Analysis of the Meeting Time



VAE on Fashion-MNIST and CIFAR-10

train log-likelihood (fashion-MNIST)



test log-likelihood

	Fashion-MNIST	CIFAR-10
ELBO	-173.36 ± 0.40	-152.06 ± 0.30
IWAE	-170.50 ± 0.30	-149.72 ± 0.39
IWAE + C-ISIR-DISIR	$\mathbf{-168.19 \pm 0.32}$	$\mathbf{-148.40 \pm 0.27}$

Conclusions

- The combination of latent space augmentation and coupling estimators gives practical unbiased gradients
- Unbiased gradient estimation improves the model's performance for VAEs
- The computational time is higher, but we can use this method to refine model fits
- Future work on improving coupling estimators will also reduce the computational complexity