IWAE vs. VAE

Tighter Bounds, Richer Latents?

Cedric Damais, Yacine Benihaddadene, Amine Mike El Maalouf, Leon Ayral, Oscar Le Dauphin

GAIDM
EPITA

January 21, 2026

The Goal: Latent Variable Inference

The Objective: Given data x, we want to learn the posterior distribution of the latent variables z:

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

- $p(x|z)$: Likelihood (Decoder)
- $p(z)$: Prior (e.g., $\mathcal{N}(0, I)$)
- $p(x)$: **Marginal Likelihood (The Evidence)**

The Problem: Intractability

We can't simply use Bayes Formula because p(x) is intractable. To calculate the denominator $p(x)$, we must marginalize out z:

$$p(x) = \int p(x|z)p(z)\, dz$$

Standard VAE: The Objective

Goal: Maximize marginal log-likelihood $\log p(x)$.

Since $p(x)$ is intractable, VAE ($K = 1$) maximizes the **ELBO**:

$$\log p(x) \geq \mathcal{L}_{\text{VAE}} = \mathbb{E}_{z\sim q}\left[\log \frac{p(x,z)}{q(z|x)}\right]$$

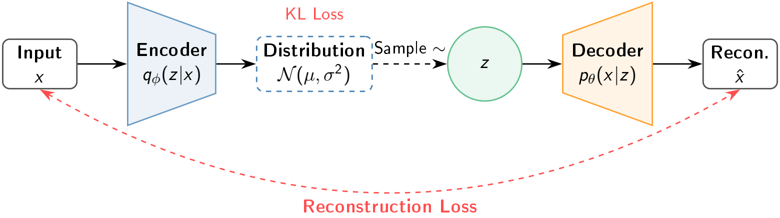
The Problem: The "Gap"

The bound is strictly lower than the evidence due to the KL divergence:

$$\log p(x) - \mathcal{L}_{\text{VAE}} = \text{KL}(q(z|x)||p(z|x))$$

- If $q(z|x)$ is too simple $\rightarrow$ **Loose Bound**.
- Loose bound $\rightarrow$ Risk of **Posterior Collapse**.

Model Architecture



Novelty 1: Strictly Tighter Bounds

Theorem (Burda et al., 2015): The IWAE bound $\mathcal{L}_K$ is monotonically increasing with K.

$$\mathcal{L}_1 \leq \mathcal{L}_2 \leq \cdots \leq \mathcal{L}_K \leq \cdots \leq \log p(x)$$

Why?

Using **Jensen's Inequality** on the concave log function:

$$\mathbb{E}\left[\log\left(\frac{1}{K}\sum w_i\right)\right] \leq \log\left(\mathbb{E}\left[\frac{1}{K}\sum w_i\right]\right) = \log p(x)$$

Implication:

- As $K \to \infty$, the estimator converges to the true marginal likelihood $\log p(x)$.
- Even with a finite K, we are guaranteed a better objective

Novelty 2: Richer Implicit Posteriors

The IWAE Solution: Weighting as Filtering

Normalized Weight: $\tilde{w}_i = \frac{w_i}{\sum_{j=1}^{k} w_j}$ where $w_i = \frac{p(x, z_i)}{q(z_i|x)}$

$$\underbrace{\tilde{q}(z|x)}_{\substack{\text{Implicit} \\ \text{Multi-modal}}} \approx \sum_{i=1}^{k} \tilde{w}_i \delta(z - z_i)$$

$$\nabla \mathcal{L}_k = \mathbb{E}_\epsilon \left[\sum_{i=1}^{k} \tilde{w}_i \nabla_\theta \log w(x, z_i, \theta) \right]$$

Why Sampling More Helps? (The Mechanics)

Case $K = 1$ (Risky)

If we draw just **one** bad sample z_1:

$$w_1 \approx 0$$

$$\log(w_1) \to -\infty$$

Gradient Explodes

Case $K > 1$ (Hedging)

If we draw many bad samples, but just **one good one**:

$$\log(0 + 0 + \cdots + w_{\text{good}})$$

$$\approx \log(w_{\text{good}}) > -\infty$$

Stable Training

The Insight: The sum acts as a safety net. The Encoder is allowed to make mistakes, as long as it gets it right *once*.

The Mechanism: Importance Weighting

1. The Definition

- Calculate raw weight:

$$w_i = \frac{p(x, z_i)}{q(z_i|x)}$$

- Normalize:

$$\tilde{w}_i = \frac{w_i}{\sum_{j=1}^{k} w_j}$$

2. The Intuition

Sample z_i is **Good**
$\rightarrow p(x, z_i)$ is High
$\rightarrow \tilde{\mathrm{w}}_{\mathrm{i}} \approx 1$

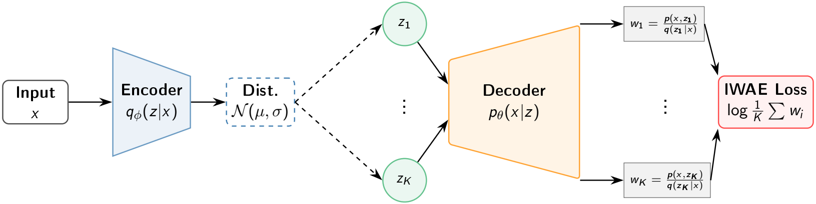
Sample z_j is **Bad**
$\rightarrow p(x, z_j)$ is Low
$\rightarrow \tilde{\mathrm{w}}_{\mathrm{j}} \approx 0$

Key Result: The Filter Effect

The gradient update effectively **ignores** bad samples:

k

IWAE Architecture



Experimental Setup

- **Objective:** Compare VAE ($K = 1$) vs IWAE ($K > 1$).
- **Constraint: Identical Architecture** for fair comparison.

Parameter	**Value**
Dataset	MNIST (Binarized)
Encoder	MLP ($784 \to 200 \to 200 \to 50$)
Decoder	MLP ($50 \to 200 \to 200 \to 784$)
Optimizer	Adam ($lr = 1e - 3$)
K (Samples)	**1, 5, 20, 30, 50, 100**

Implementation Detail

Used `torch.logsumexp` to avoid numerical underflow.

Result 1: Estimated Log-Likelihood vs Number of Samples (K)

- **Metric:** Estimated Log-Likelihood (Higher is better).

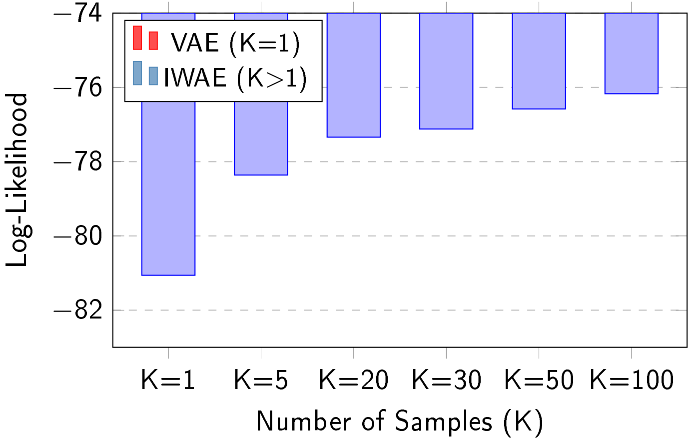
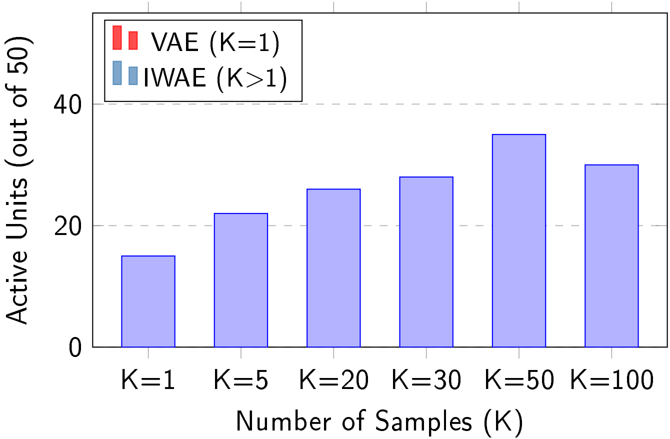


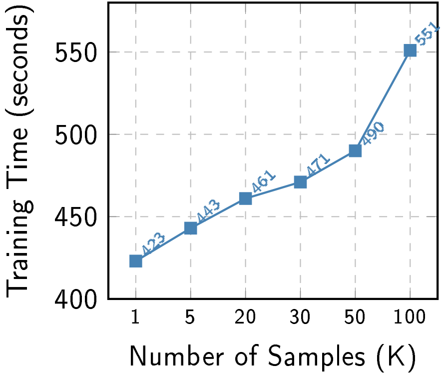
Figure 1: Estimated Log-Likelihood on MNIST Test Set (Higher is Better)

Result 2: Active Latent Units vs Number of Samples (K)

- **Metric:** Active Units (Dimensions where KL $> \epsilon$, with $\epsilon = 0.01$).



Result 3: Impact of K on Training Time



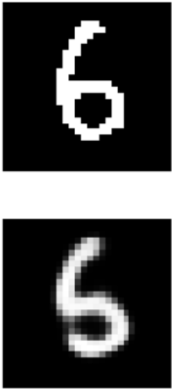
Trade-off Analysis:

- **Compute:** Training time scales **sub-linearly** with K (423s $\rightarrow$ 551s).
- **Log-Likelihood:** Diminishing returns as K increases.
- **Gradient SNR:** Decreases at high K (encoder neglect).

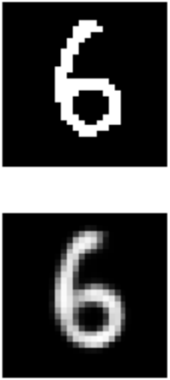
1. Introduction & Theory
2. The IWAE Solution
3. Methodology
4. Results
   - Log-Likelihood
   - Latent Utilization
   - K-Analysis
   - Sample Quality
5. Conclusion

Result 4: Sample Quality

VAE (K=1)



IWAE (K=50)



IWAE samples typically show sharper strokes and fewer "averaged" blurry digits.

Trade-offs Summary

Metric	VAE (K=1)	IWAE (K=50)
Bound Tightness	Loose	**Tight**
Latent Usage	Risk of Collapse	**Rich**
Bound Variance	High	**Low**
Encoder Gradient SNR	**High**	Low
Sample Quality	Blurry	**Sharp**
Compute Cost	**Low**	High ($\times K$)

Thanks!

Q & A