NVAE: A Deep Hierarchical Variational Autoencoder

Can better architecture make VAEs great again? (NeurIPS 2020)

Arash Vahdat, Jan Kautz

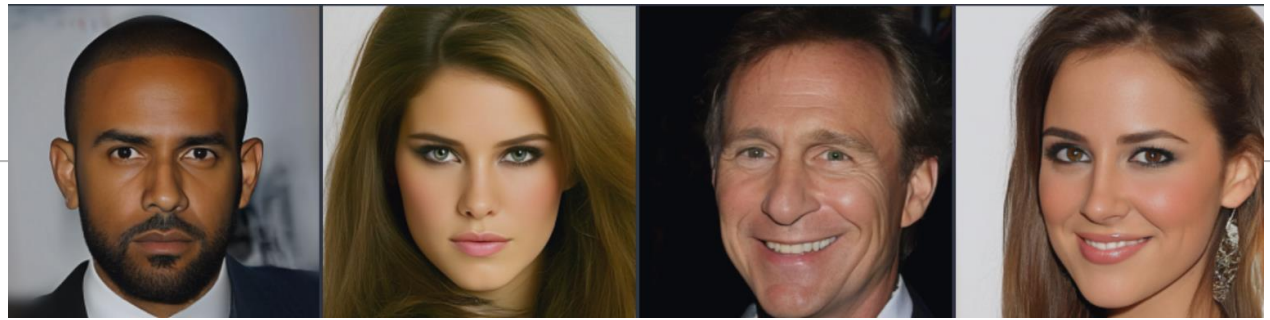
NVIDIA

Presented by: Amogh Gupta

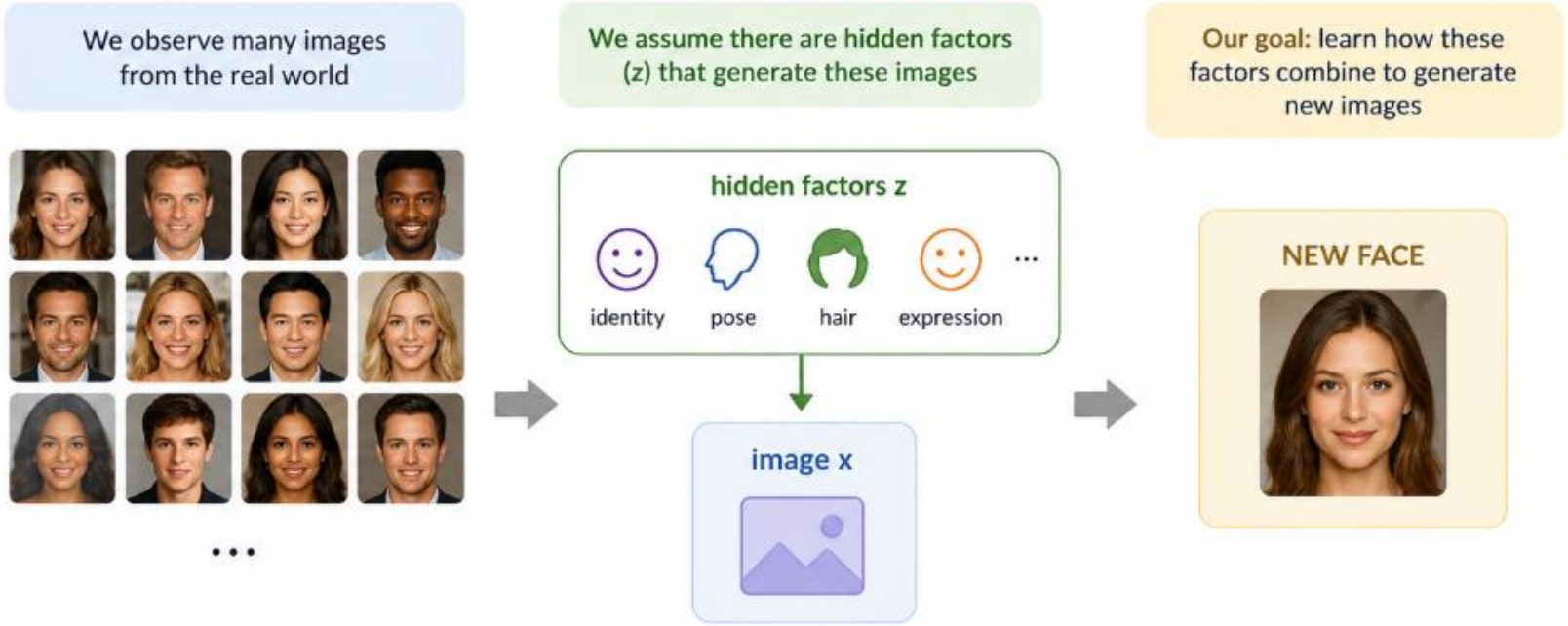
The University of North Carolina at Chapel Hill

COMP 690 : Zhongzheng Ren

Generating Realistic Images



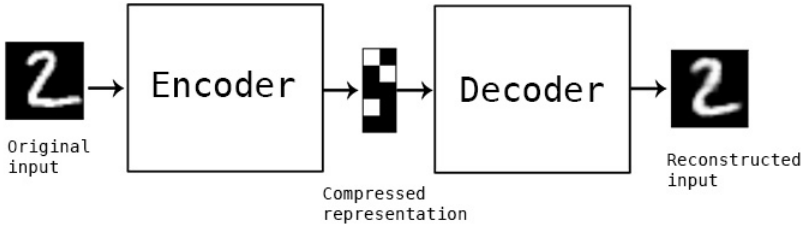
What is a VAE trying to do?

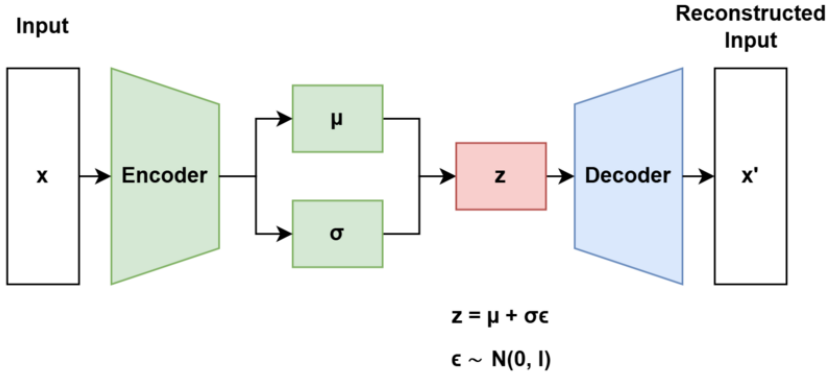


hidden factors z → image x
z = {pose, identity, hair, expression, ...}

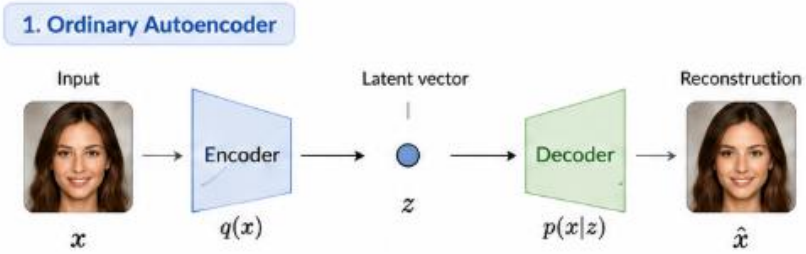
$$p(x) = \int p(x|z)p(z)\,dz$$

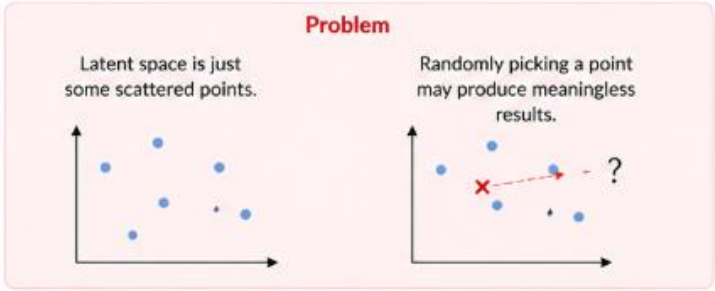
Autoencoders -> VAEs

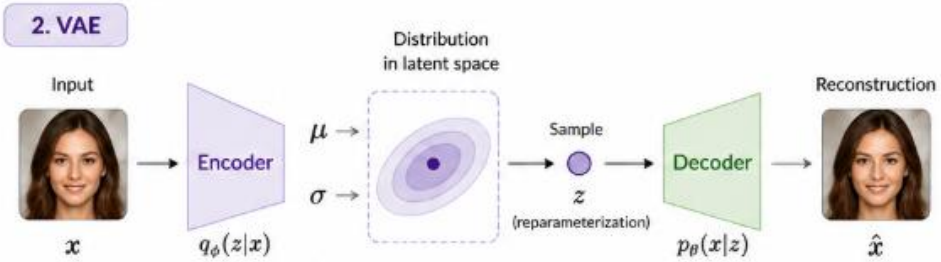


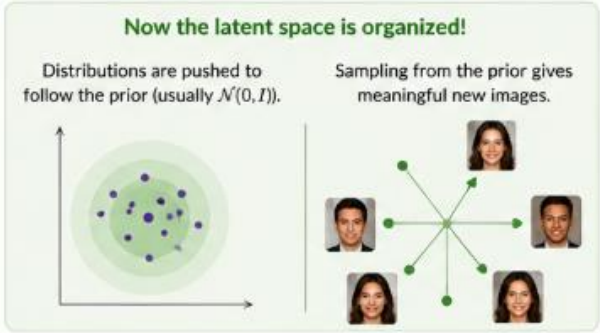


Latent Space of VAE









$$q_\phi(z|x) \approx p(z)$$

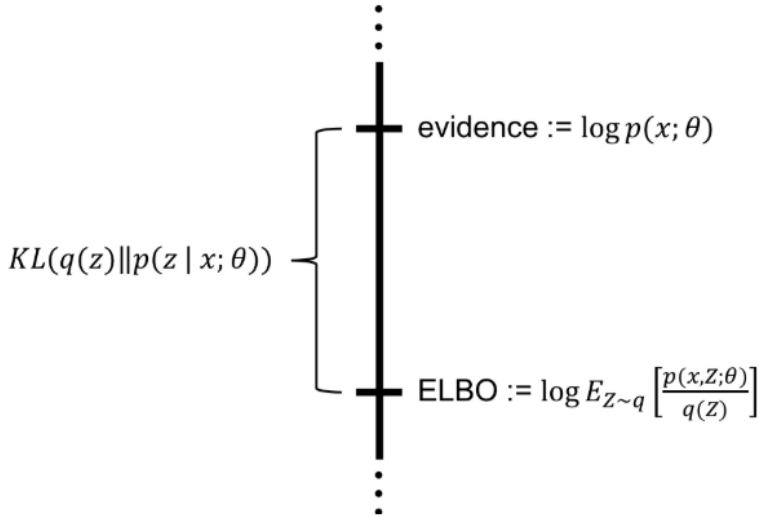
$$p(z) = \mathcal{N}(0, I)$$

$$KL(q_\phi(z|x)||p(z))$$

VAE Loss

$$\mathcal{L}_{\text{VAE}} = \underbrace{E_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{reconstruct the image}} - \underbrace{KL(q_\phi(z|x)||p(z))}_{\text{keep latent space organized}}$$

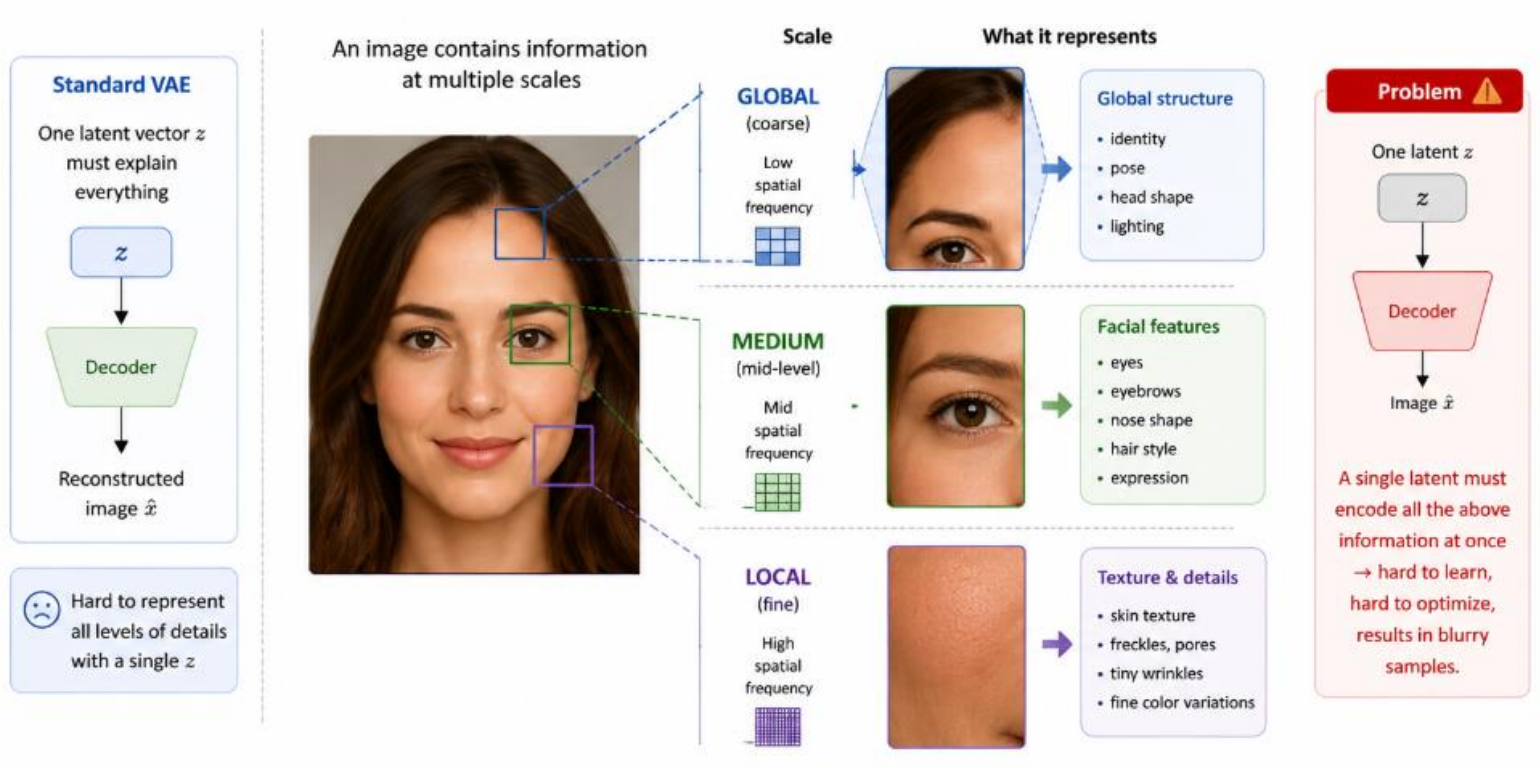
VAE Loss



This can be derived as follows:

$$
\begin{aligned}
KL(q(z) \;||\; p(z \mid x;\theta)) &:= E_{Z\sim q}\left[\log \frac{q(Z)}{p(Z \mid x;\theta)}\right] \\
&= E_{Z\sim q}\left[\log q(Z)\right] - E_{Z\sim q}\left[\log \frac{p(x,Z;\theta)}{p(x;\theta)}\right] \\
&= E_{Z\sim q}\left[\log q(Z)\right] - E_{Z\sim q}\left[\log p(x,Z;\theta)\right] + E_{Z\sim q}\left[\log p(x;\theta)\right] \\
&= \log p(x;\theta) - E_{Z\sim q}\left[\log \frac{p(x,Z;\theta)}{q(z)}\right] \\
&= \text{evidence} - \text{ELBO}
\end{aligned}
$$

Why does standard VAE struggle?



Hierarchical Latent Variables

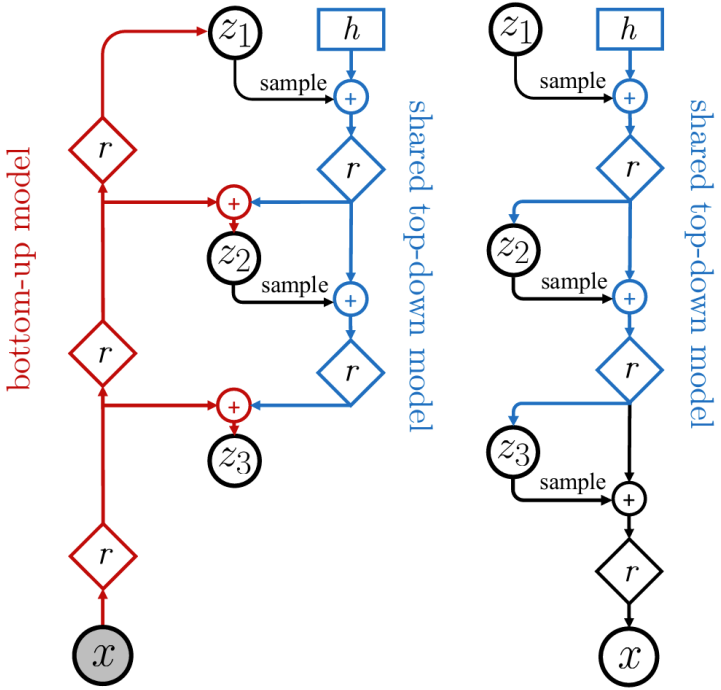
$$\boldsymbol{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_L\}$$

$$p(\boldsymbol{z}) = \prod_l p(\boldsymbol{z}_l|\boldsymbol{z}_{<l}) \qquad q(\boldsymbol{z}|\boldsymbol{x}) = \prod_l q(\boldsymbol{z}_l|\boldsymbol{z}_{<l}, \boldsymbol{x})$$

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{x}) := \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - \text{KL}(q(\boldsymbol{z}_1|\boldsymbol{x})||p(\boldsymbol{z}_1)) - \sum_{}^{L} \mathbb{E}_{q(\boldsymbol{z}_{<l}|\boldsymbol{x})}\left[\text{KL}(q(\boldsymbol{z}_l|\boldsymbol{x}, \boldsymbol{z}_{<l})||p(\boldsymbol{z}_l|\boldsymbol{z}_{<l}))\right]$$

Instead of a single latent zz, NVAE introduces a hierarchy, where different levels capture information at different spatial scales.
The joint latent distribution is modeled hierarchically, allowing higher-level latents to influence the distribution of lower-level latents.

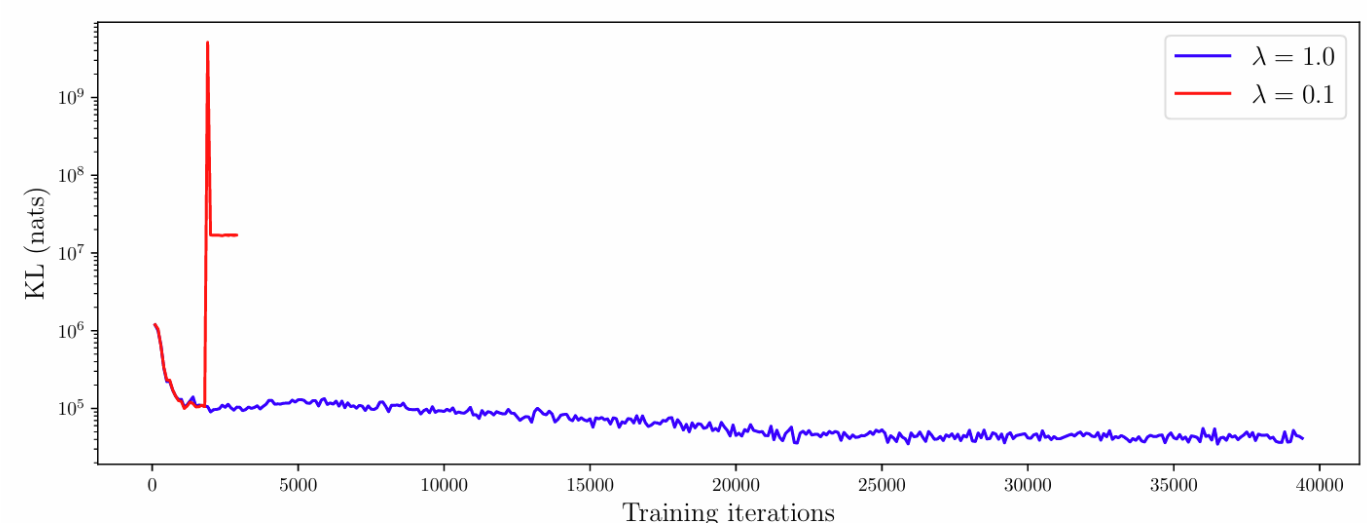
NVAE Architecture



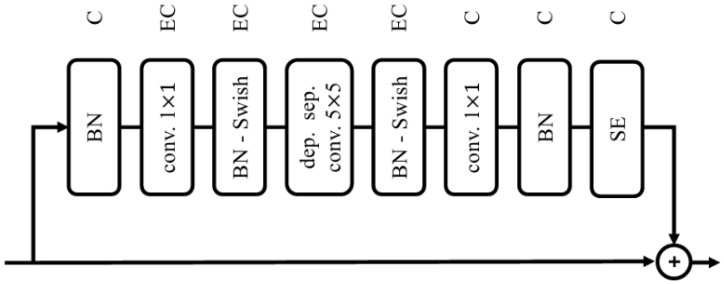
(a) Bidirectional Encoder (b) Generative Model

NVAE Architecture

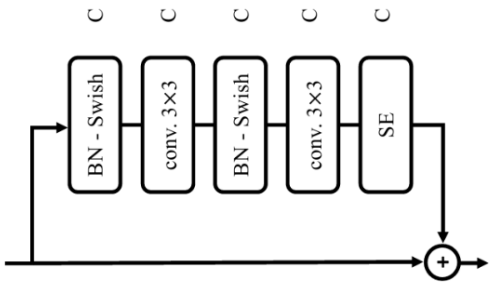
More Latent variables -> More expressive Model -> Training Unstable



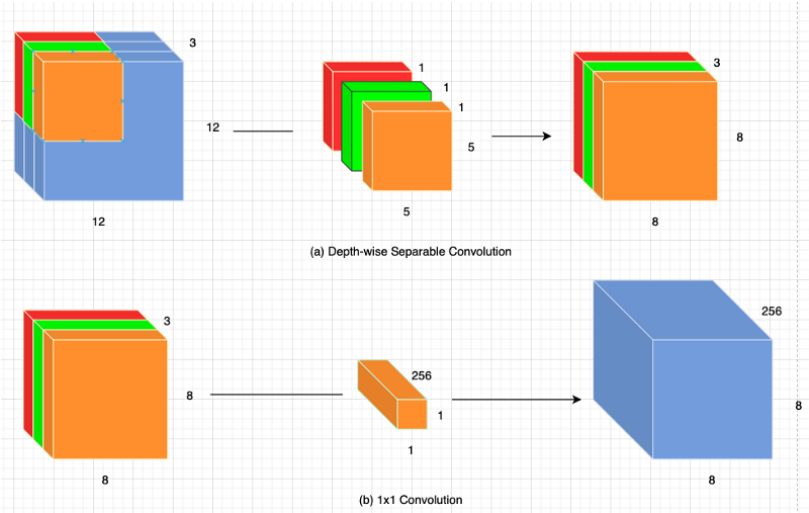
NVAE: Better Cells



(a) Residual Cell for NVAE Generative Model



(b) Residual Cell for NVAE Encoder



Residual Normal Distributions

Normally, the prior and posterior are parameterized independently.

Prior: N(μ,σ) Posterior: N(μq,σq)

$$\boxed{\mu_q = \mu + \Delta\mu}$$

$$\boxed{\sigma_q = \sigma\Delta\sigma}$$

NVAE instead says:

N(μ,σ) -> Posterior N(μ + Δμ, σ × Δσ)

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

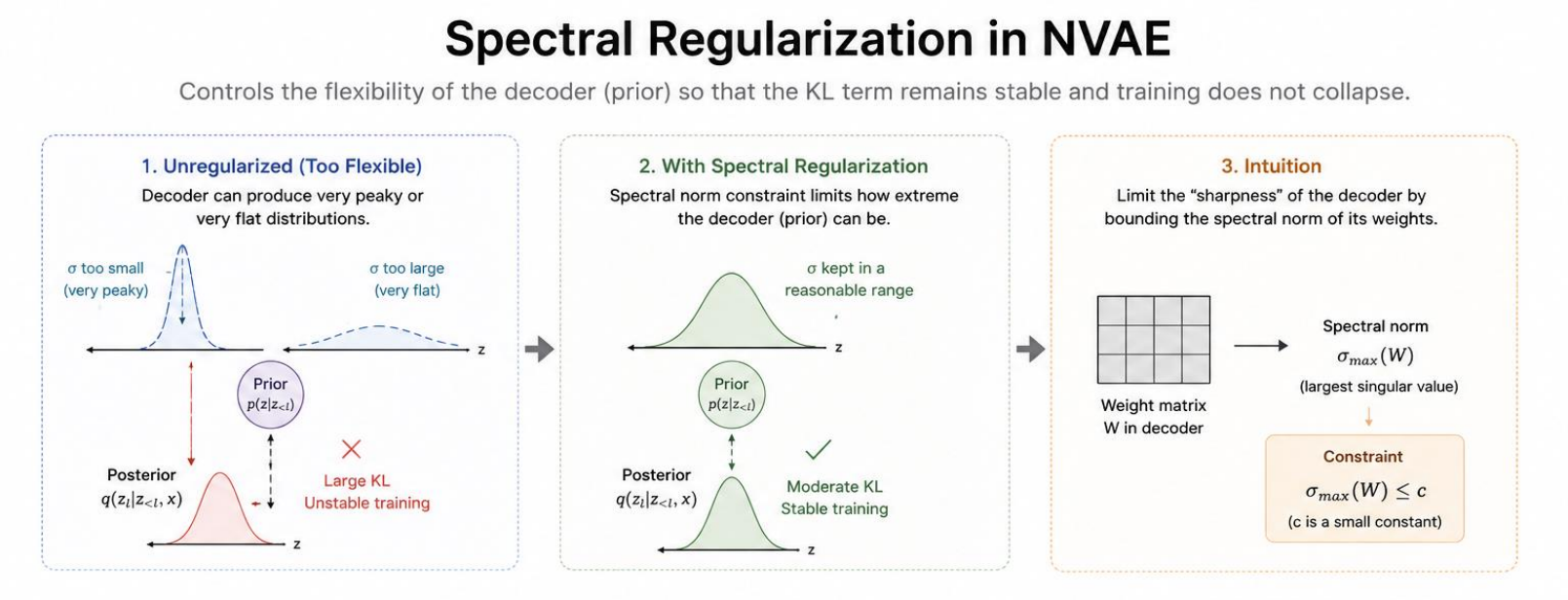
Instead of asking the encoder:

"What should the absolute mean and variance of my posterior be?"

NVAE asks:

"How much should I modify the prior to obtain the posterior?"

Spectral Regularization



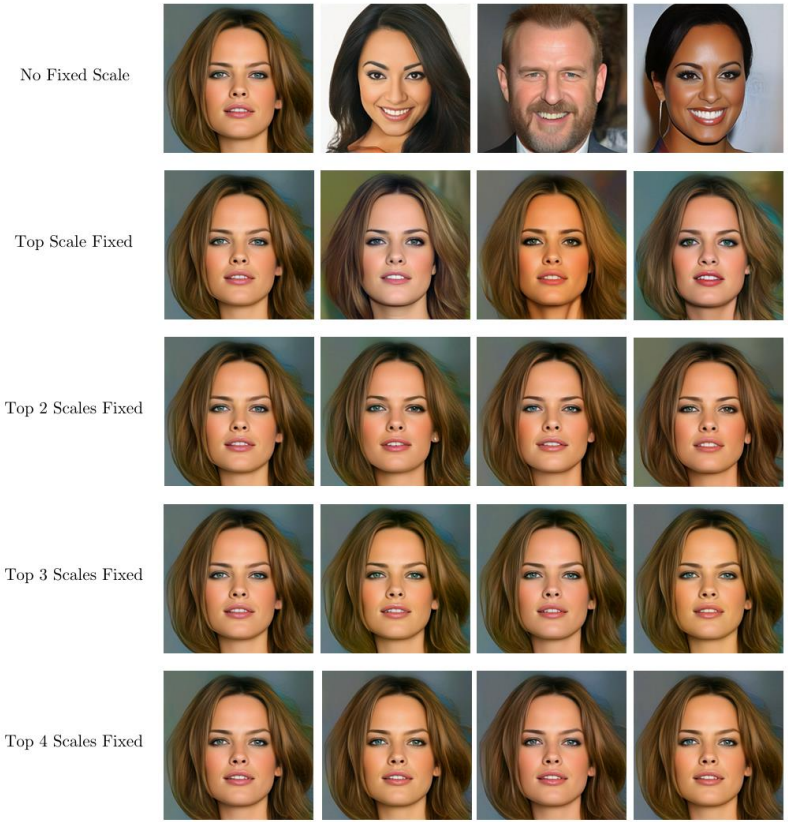
For each weight matrix WW, we constrain its largest singular value.
Spectral regularization keeps the decoder smooth, preventing extreme prior distributions and making the hierarchical KL easier to optimize.

Results

Table 1: Comparison against the state-of-the-art likelihood-based generative models. The performance is measured in bits/dimension (bpd) for all the datasets but MNIST in which negative log-likelihood in nats is reported (lower is better in all cases). NVAE outperforms previous non-autoregressive models on most datasets and reduces the gap with autoregressive models.

Method	MNIST 28×28	CIFAR-10 32×32	ImageNet 32×32	CelebA 64×64	CelebA HQ 256×256	FFHQ 256×256
NVAE w/o flow	**78.01**	2.93	-	2.04	-	0.71
NVAE w/ flow	78.19	**2.91**	3.92	**2.03**	**0.70**	**0.69**
VAE Models with an Unconditional Decoder						
BIVA [36]	78.41	3.08	3.96	2.48	-	-
IAF-VAE [4]	79.10	3.11	-	-	-	-
DVAE++ [20]	78.49	3.38	-	-	-	-
Conv Draw [42]	-	3.58	4.40	-	-	-
Flow Models without any Autoregressive Components in the Generative Model						
VFlow [59]	-	2.98	-	-	-	-
ANF [60]	-	3.05	3.92	-	0.72	-
Flow++ [61]	-	3.08	**3.86**	-	-	-
Residual flow [50]	-	3.28	4.01	-	0.99	-
GLOW [62]	-	3.35	4.09	-	1.03	-
Real NVP [63]	-	3.49	4.28	3.02	-	-
VAE and Flow Models with Autoregressive Components in the Generative Model						
δ-VAE [25]	-	2.83	3.77	-	-	-
PixelVAE++ [35]	78.00	2.90	-	-	-	-
VampPrior [64]	78.45	-	-	-	-	-
MAE [65]	77.98	2.95	-	-	-	-
Lossy VAE [66]	78.53	2.95	-	-	-	-
MaCow [67]	-	3.16	-	-	0.67	-
Autoregressive Models						
SPN [68]	-	-	3.85	-	0.61	-
PixelSNAIL [34]	-	2.85	3.80	-	-	-
Image Transformer [69]	-	2.90	3.77	-	-	-
PixelCNN++ [70]	-	2.92	-	-	-	-
PixelRNN [41]	-	3.00	3.86	-	-	-
Gated PixelCNN [71]	-	3.03	3.83	-	-	-

Does the hierarchy mean anything?



Strengths and Weaknesses

1. Hierarchy captures information at different spatial scales

1. NVAE achieves highly competitive likelihood and sample-quality results

1. The hierarchical design allows information to be distributed across multiple latent groups rather than forcing everything into one bottleneck.

The hierarchy is largely hand-designed
(***Can the model learn its own hierarchy?)***

Hierarchical latents are not guaranteed to be semantically disentangled

ELBO/likelihood optimization can reward pixel-level fidelity without necessarily producing the most perceptually realistic samples.

Open Questions and Future Ideas

1. Learnable Hierarchy

Instead of manually defining the number and resolution of latent groups, let the model learn the optimal hierarchy.
The model could dynamically decide which information belongs at each level and which latent variables are actually needed.

2. Hierarchical VAE + Diffusion

Use NVAE's hierarchy to provide multi-scale latent representations, then perform diffusion in the learned latent space.
High-level latents could model global structure, while diffusion focuses computation on fine-grained details.
This could potentially combine VAE efficiency and structured latents with diffusion-quality generation.

Question:

Could we generate or edit an image by modifying only the relevant latent levels, instead of running the entire generative model? Maybe something like localized editing??

Key Takeaways

NVAE uses a **deep hierarchy of latent variables** rather than a single latent space.

Different levels capture information at different scales: **global** → **local** → **fine-grained details**.

Residual parameterization and carefully designed normalization help maintain stable training.

Generates high-quality, diverse images while retaining the probabilistic advantages of VAEs.

Shows that VAEs don't necessarily have to sacrifice generation quality for **structured, interpretable latent representations**.

THANK YOU