上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

# Lecture 24: Variational Autoencoders

Tao Huang

John Hopcroft Center, School of Computer Science, Shanghai Jiao Tong University

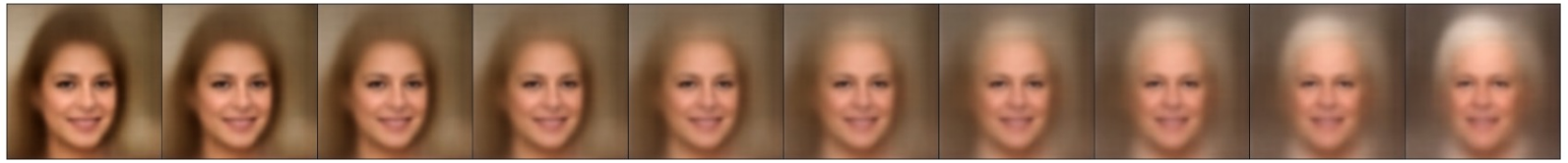
https://taohuang.info/cs3317

https://oc.sjtu.edu.cn/courses/89538

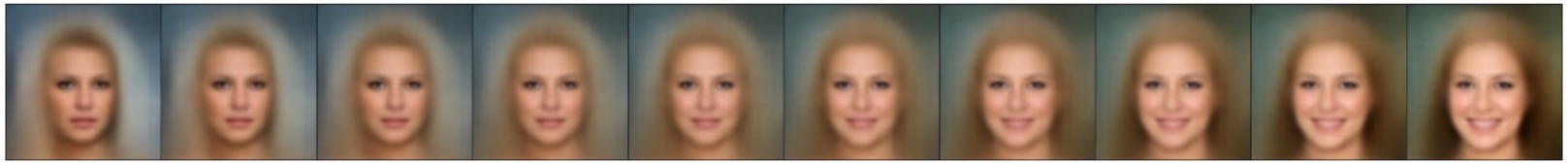
AI tools assisted in generating some figures in these slides. All such content has been reviewed, and the instructor is responsible for its accuracy.

# What a Latent Space Looks Like

- We want to generate varying images by interpolating the latents of two images



Interpolation from a non-gray-haired person to a gray-haired person



Interpolation from a non-smiling person to a smiling person

*By the end of this lecture, you will know exactly what is happening — and why autoregressive models cannot do it.*

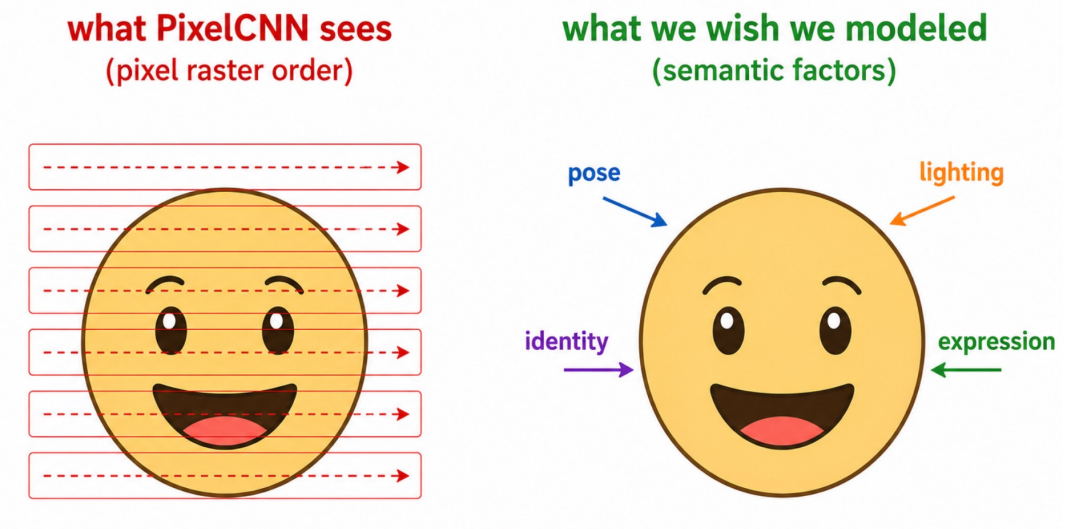
# Where We Left Off

**L25 won on tractability:**

- $p(x) = \prod_i p(x_i \mid x_{<i})$ → exact likelihood

**L25 lost on three things:**

1. Sampling is sequential — O(N) forward passes
2. No semantic latent — no $z$ to interpolate, edit, condition on
3. Pixel ordering is arbitrary — top-left → bottom-right
   has no meaning for natural images.

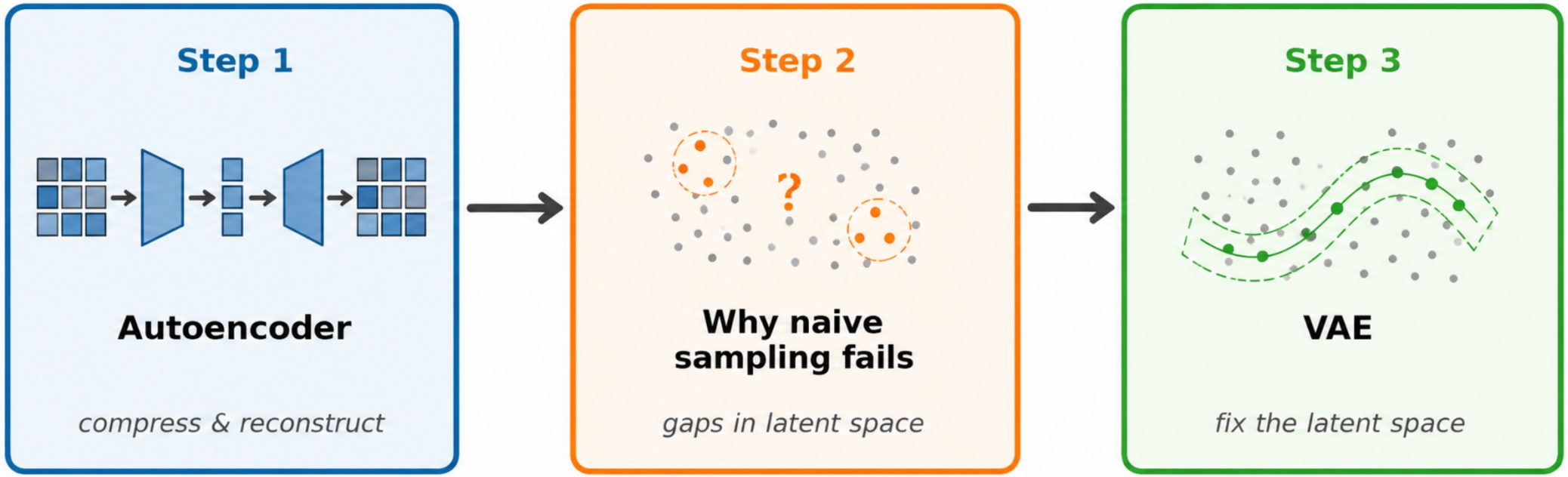


# Objectives

*By the end of this lecture, you will be able to:*

- Understand the autoencoder — encoder, bottleneck, decoder.
- See why a plain autoencoder can't generate new samples.
- Fix it: the VAE — encoder produces a distribution, not a point.
- Sample in one forward pass — and the trade-off (blurry samples).

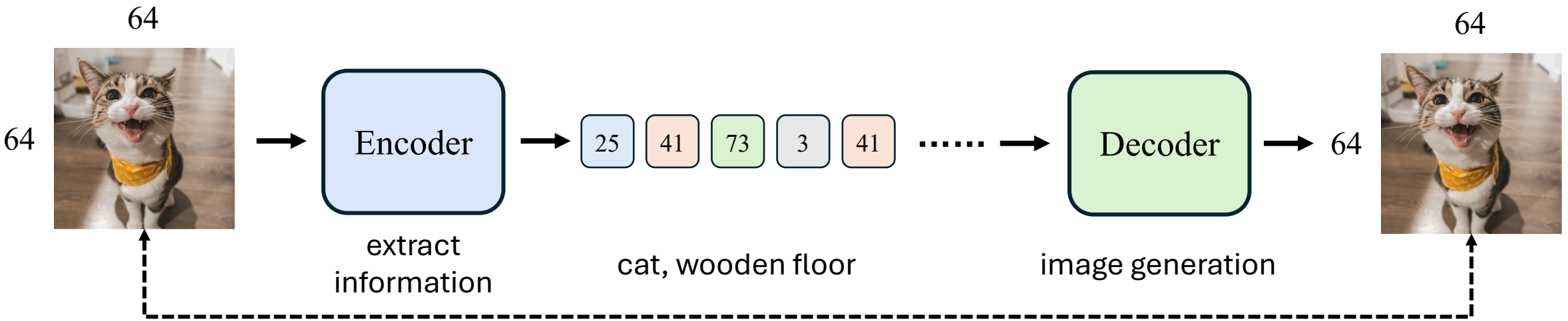
# Roadmap


Step 1
Autoencoder
compress & reconstruct
Step 2
?
Why naive sampling fails
gaps in latent space
Step 3
VAE
fix the latent space

# 1. The Autoencoder

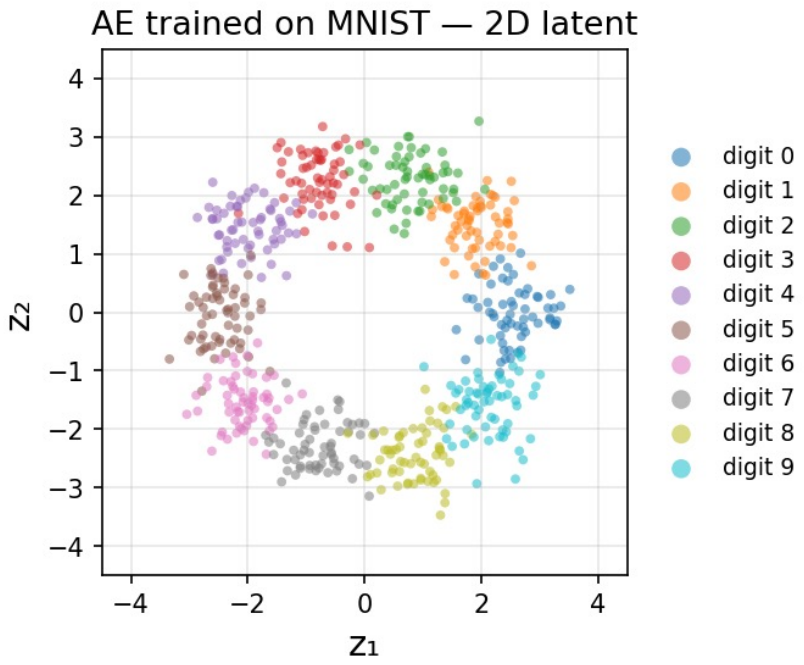
# What is an Autoencoder?

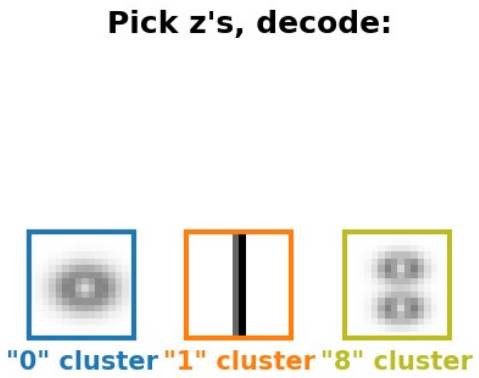
- A neural network trained to copy its input to its output — **through a narrow bottleneck.**
- Encoder $f_\theta$: $x \to z$ (low-dim). Decoder $g_\varphi$: $z \to \hat{x}$.
- *Loss:* $\|x - \hat{x}\|^2$ *— just reconstruction.*



# What the Bottleneck Brings

- Bottleneck dim ≪ input dim → the network is forced to compress.
- **To reconstruct well, z must capture what matters about x.**
- Example: AE with 2D latent on MNIST — digits cluster by identity.

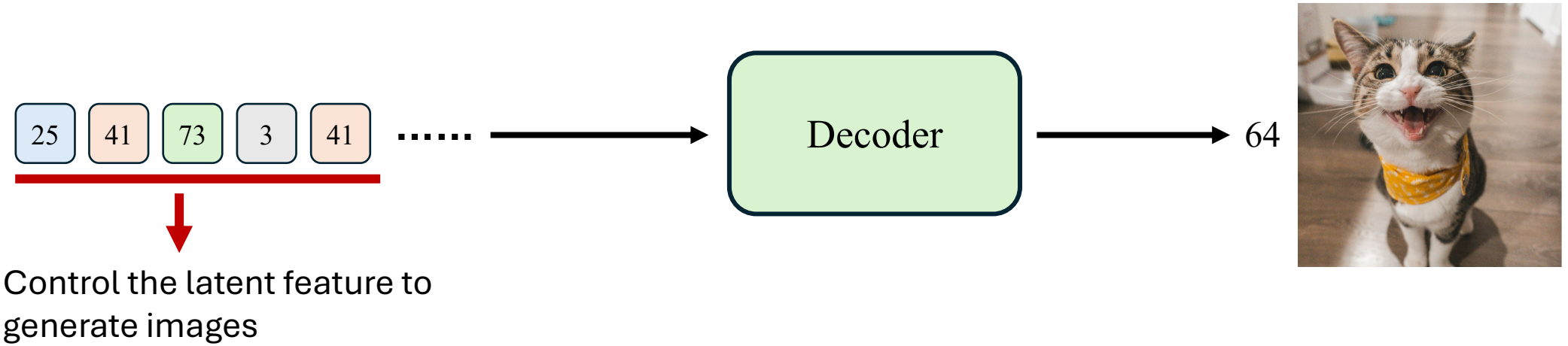


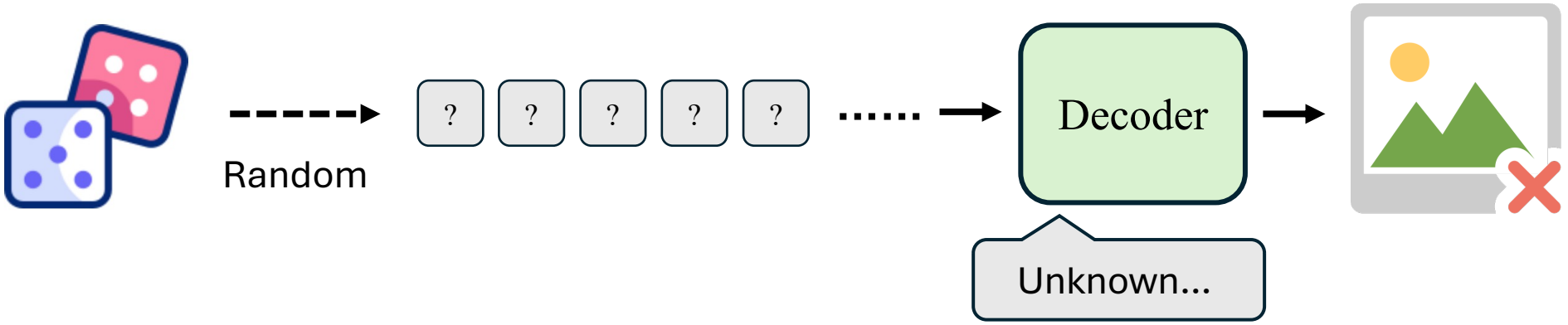

*z captures digit identity.*

# Can We Generate by Sampling z?

- Idea: z is low-dim. Pick a random z, decode it, get a new x.

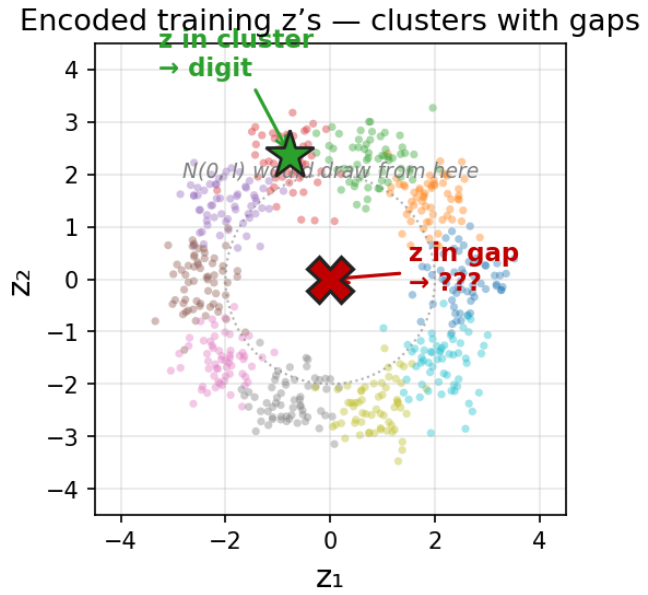


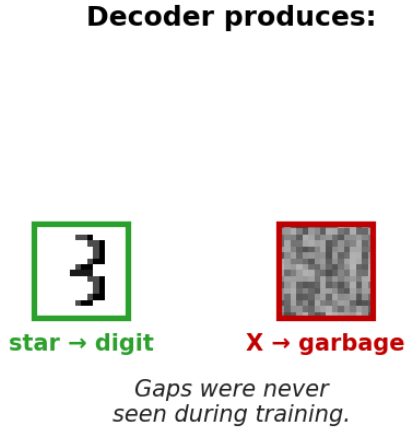

## How to generate different images?



# Why Naïve Sampling Fails

- Encoder only ever produces z's near training points.
- **Big regions of latent space are never seen during training.**
- Sample a z from one of these unseen regions → decoder produces garbage.




***The decoder was never told what z's in the gaps mean.***

# 🧠 Think

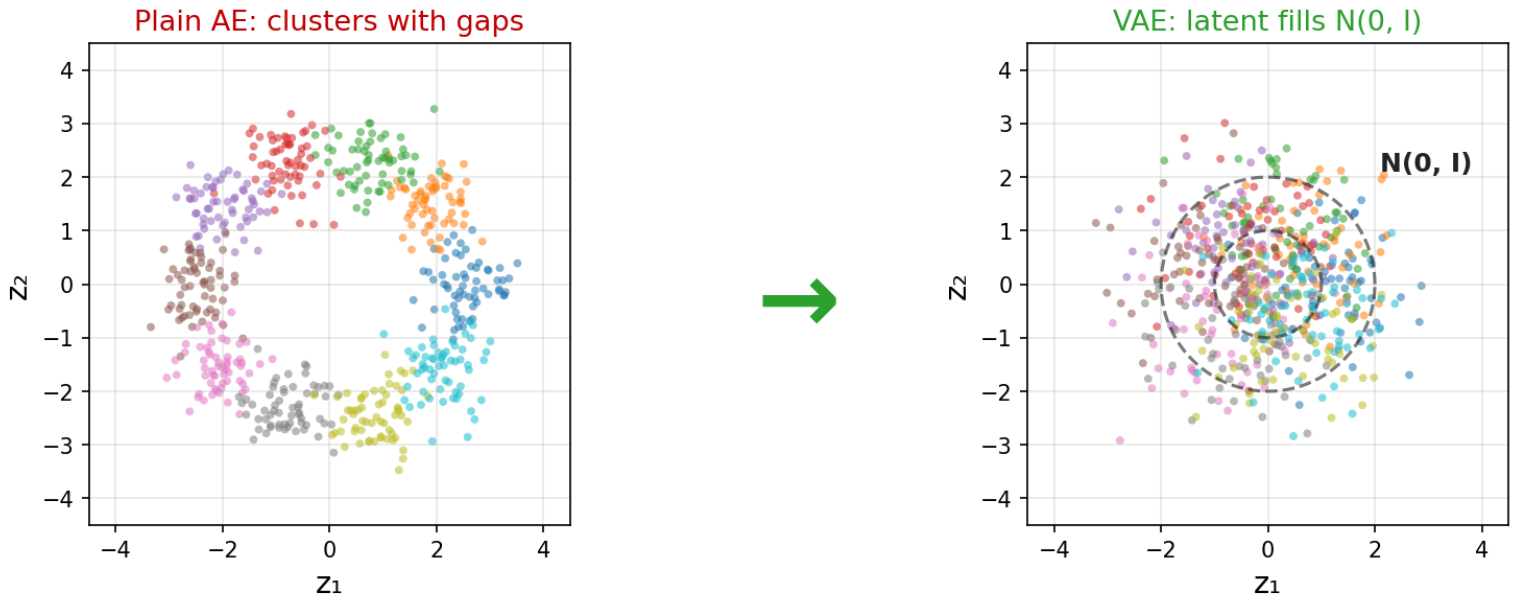
- **We want to sample new x's.**
- The decoder only behaves well in regions of z it has seen during training.

- **What's one thing you could change about how the encoder is trained to fix this?**

# 2. From AE to VAE

# The Fix

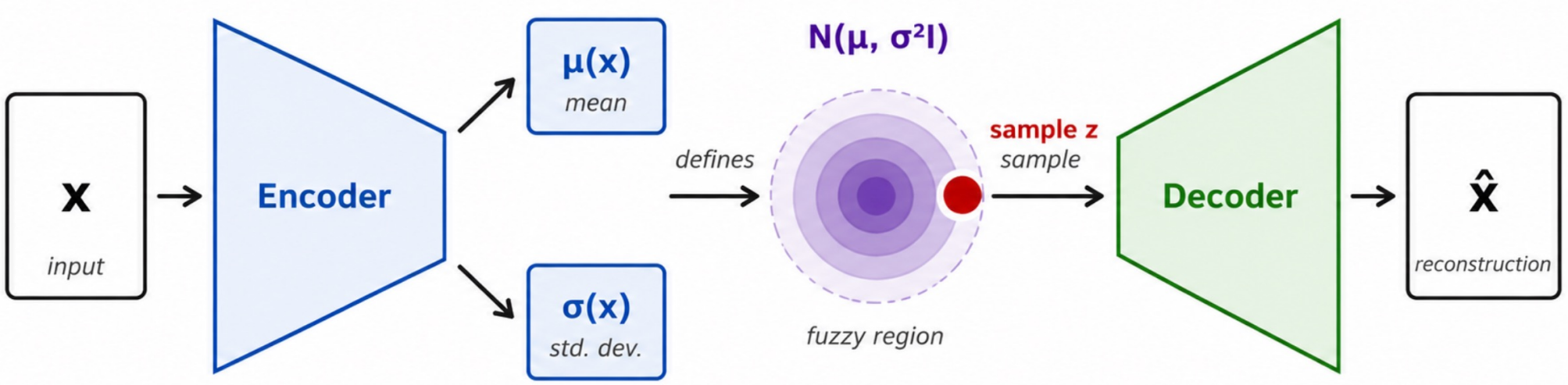
- **Force the encoder's outputs to fill a known distribution.**
- Specifically: train the encoder so the collection of z's it produces **looks like N(0, I).**




*Now sampling z ~ N(0, I) lands in regions the decoder has seen.*

# Encoder Outputs a Distribution

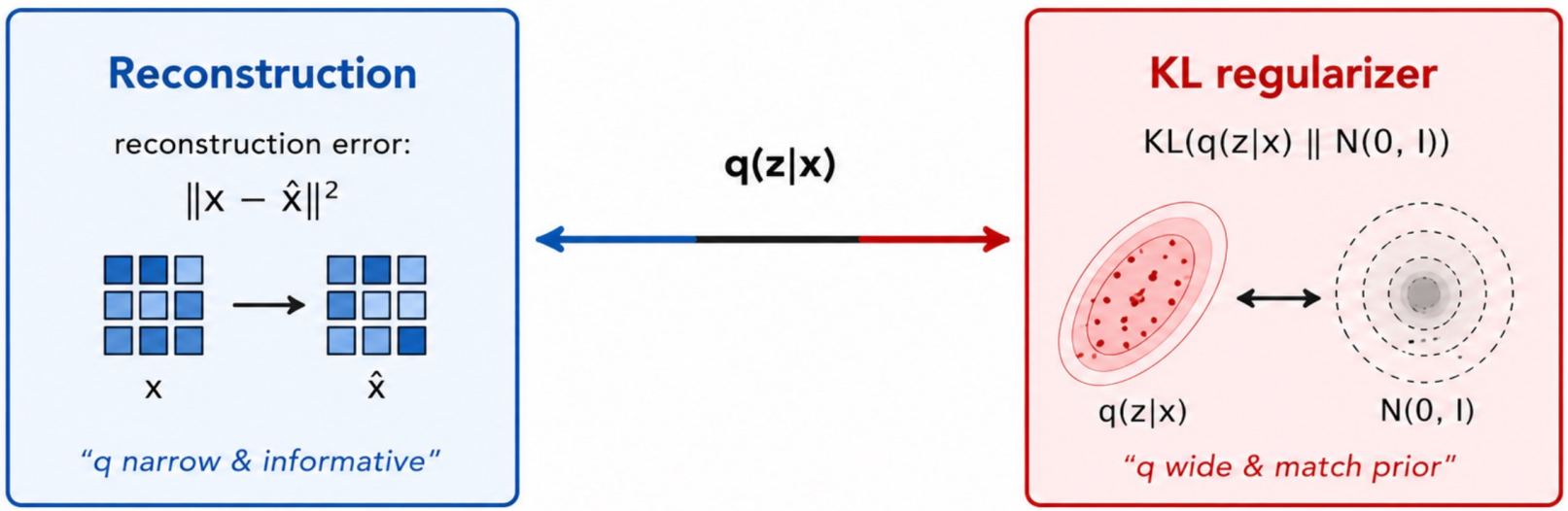
- Plain AE: encoder(x) = z  (a single point).
- **VAE: encoder(x) = ($\mu$, $\sigma^2$) — defines q(z|x) = N($\mu$, $\sigma^2$I).**
- At training time: sample z ~ q(z|x), then decode.



# Two Losses

- **Reconstruction: $\|x - \hat{x}\|^2$ (same as plain AE).**
- **KL regularizer: pulls each Gaussian $q(z|x)$ toward the prior $N(0, I)$.**

**Loss = reconstruction + KL.**



# ELBO

$$L = \mathbb{E}_q[\log p(x \mid z)] - \mathrm{KL}(q(z \mid x) \,\|\, p(z))$$

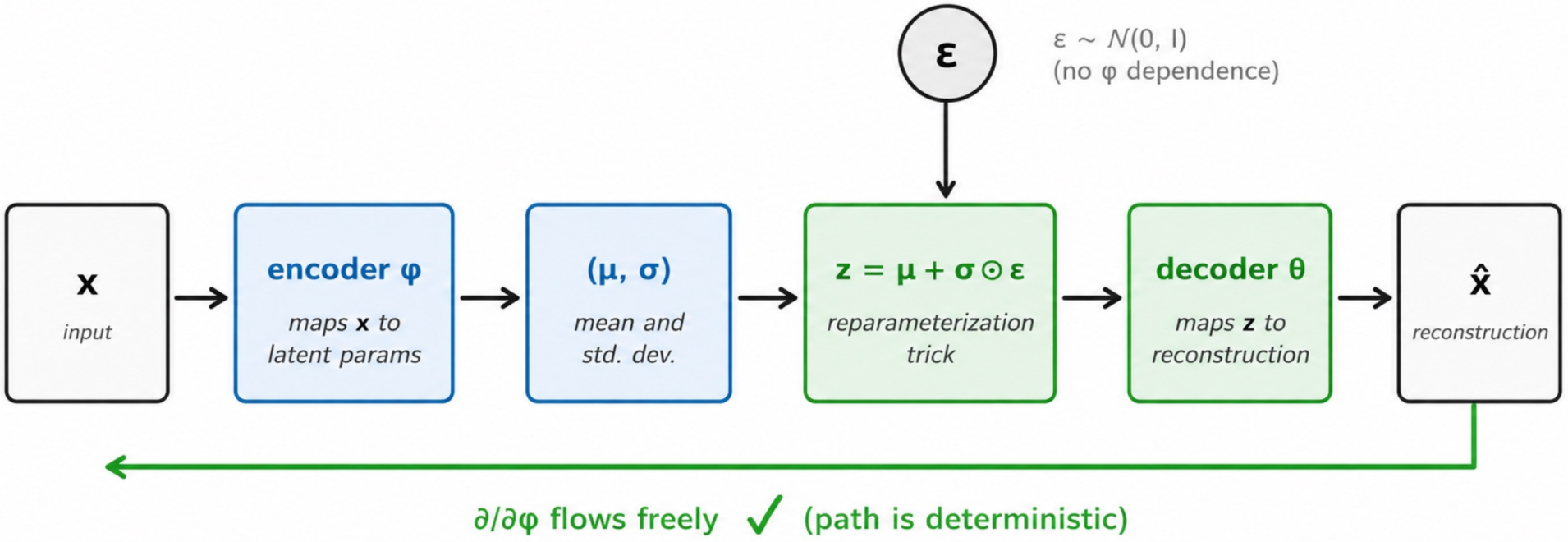
↑ *reconstruction quality* ↑ *how far encoder strays from prior*

- *Reconstruction quality minus how far the encoder strays from the prior.*

- L is a lower bound on log p(x). Maximizing L ≈ maximizing likelihood.
- **Hence: Evidence Lower BOund.**

# The Reparameterization Trick

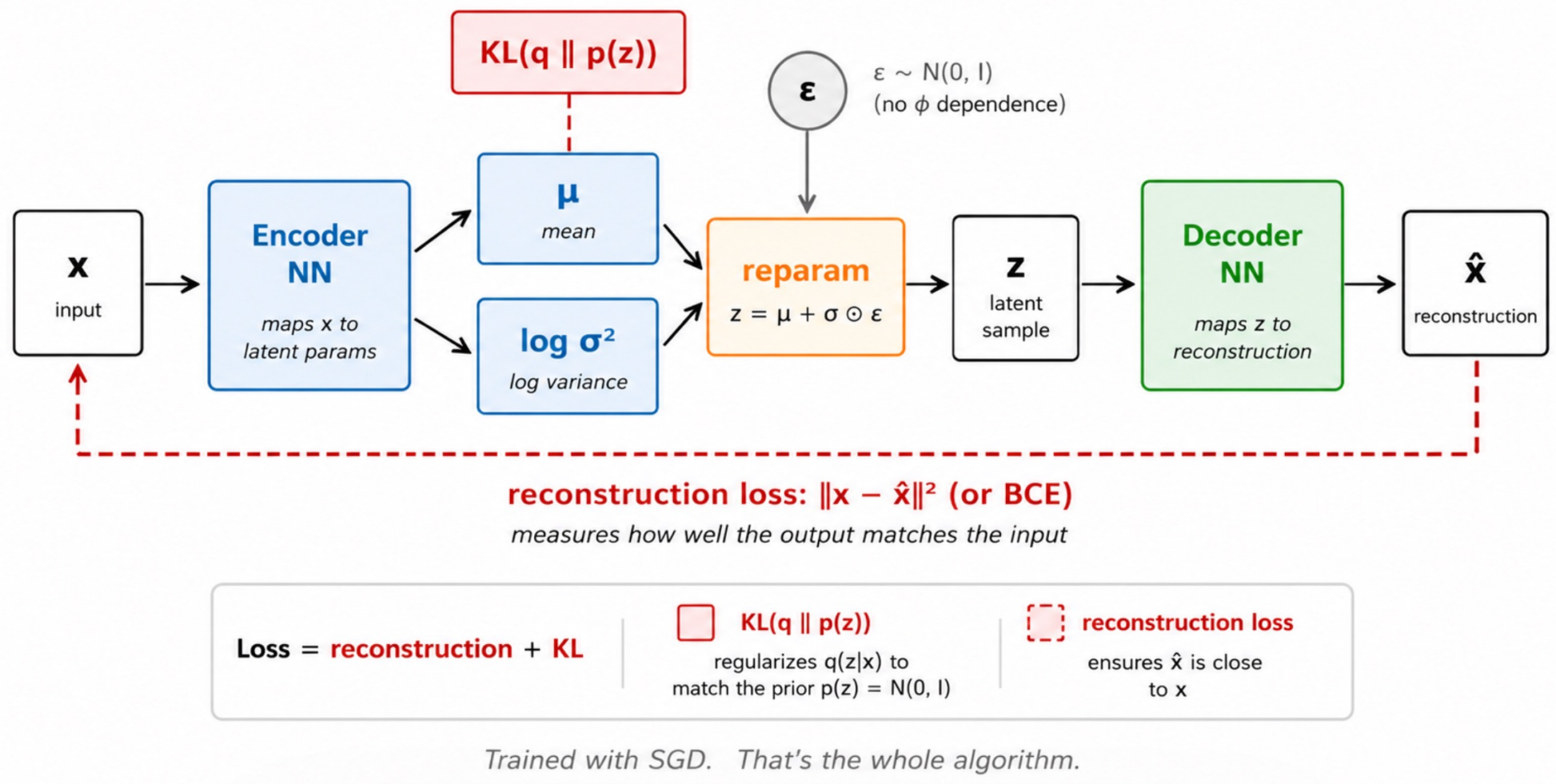
- Problem: we sample $z \sim N(\mu, \sigma^2 I)$.
- **Random samples have no gradient through μ and σ. Backprop is broken.**
- **Fix: write $z = \mu + \sigma \odot \varepsilon$ with $\varepsilon \sim N(0, I)$.**



# 3. The VAE in Practice

# The Full VAE

Encoder → (μ, log $\sigma^2$) → reparam → z → Decoder → x̂.



# Sampling: One Forward Pass

- **At test time: discard the encoder.**
- **z ~ N(0, I);  $\hat{x}$ = decoder(z).  One forward pass. Done.**

Autoregressive (PixelCNN)

VAE (single forward pass)

**wall-clock: 0.0s**

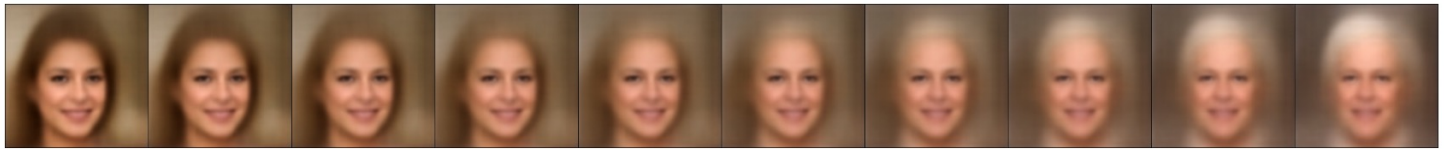
0% complete

**wall-clock: 0.0s**

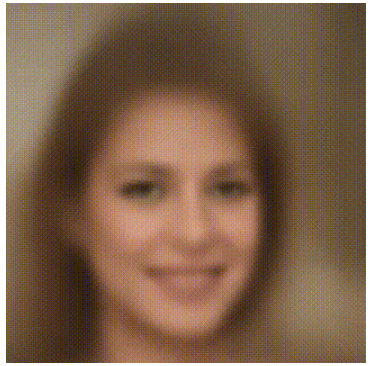
0% complete

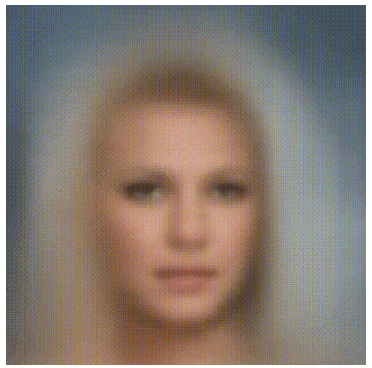
# Latent Space Interpolation

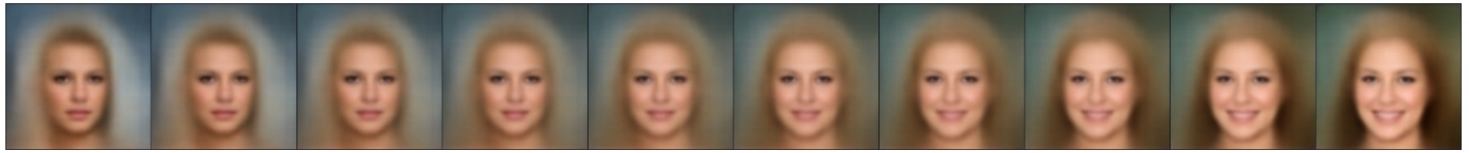
- Encode two real images $x_A, x_B \rightarrow z_A, z_B$.
- Decode along $z_t = (1-t) \cdot z_A + t \cdot z_B$.
- **Result: a smooth semantic morph — pose, expression, identity blend continuously.**



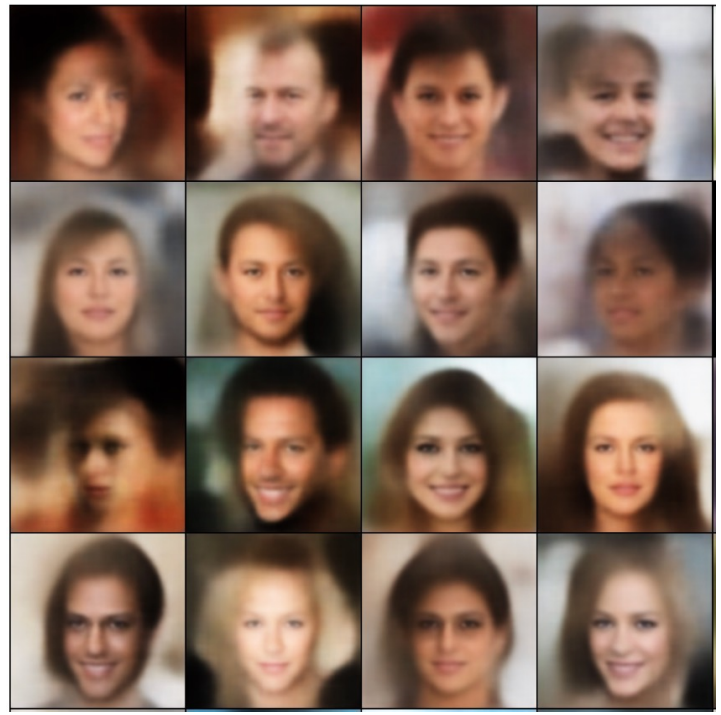
Interpolation from a non-gray-haired person to a gray-haired person







Interpolation from a non-smiling person to a smiling person



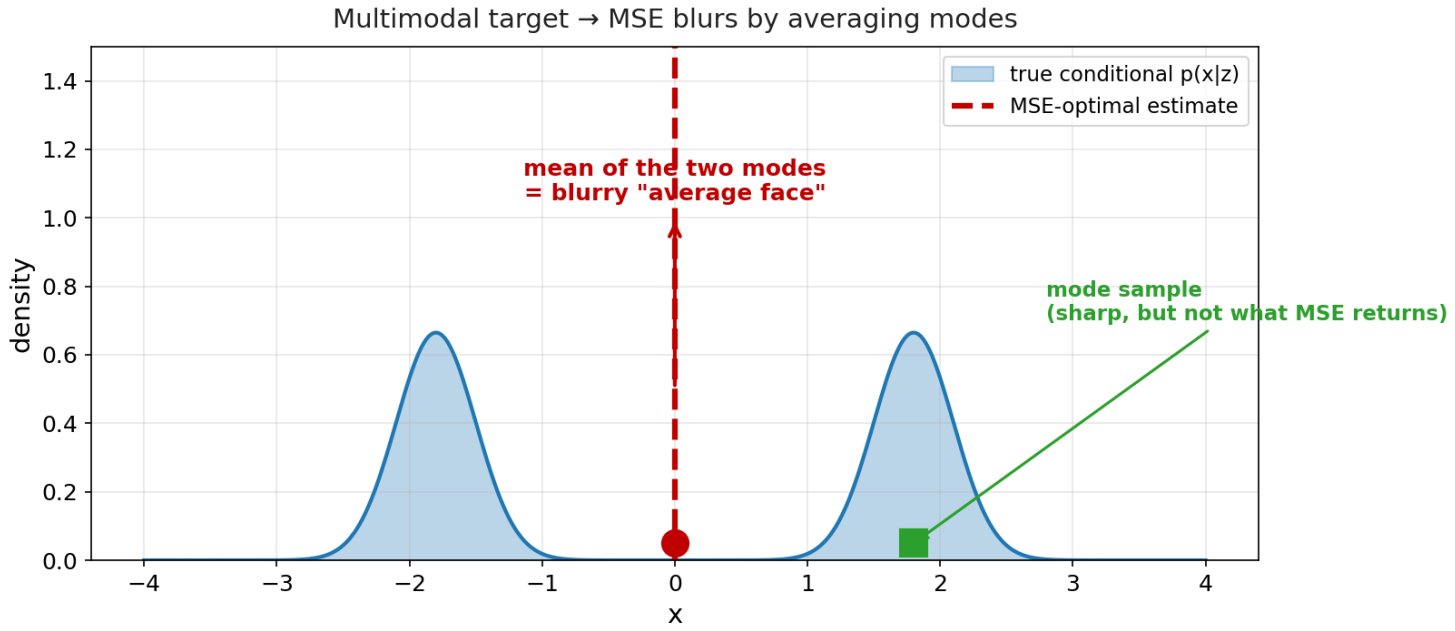
Random Sampling



https://github.com/reshalfahsi/latent-space-vae

# Why VAE Samples Are Blurry

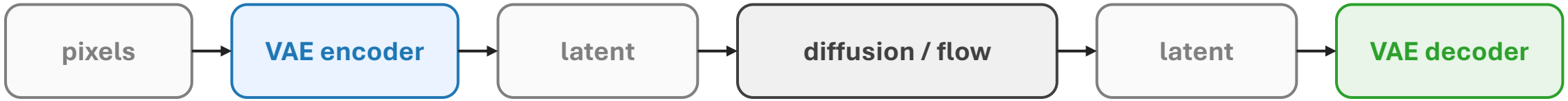
- **VAE samples are visibly soft / blurry. Why?**
- Reconstruction loss = $\|x - \hat{x}\|^2$. For a multimodal target, the loss-minimizer **is the mean of the modes — a blur.**
- *If x|z could be either of two faces, MSE returns the average — neither face.*



# VAEs in 2026

**VAEs as components: everywhere.**

- *Stable Diffusion, FLUX, SDXL, Sora, Movie Gen* — all use a VAE encoder/decoder to **compress pixels** into a latent space, where a diffusion / flow model does the actual generation.



# Summary

- Autoencoder: compress and reconstruct
- Naïve sampling fails: decoder only knows trained z's
- VAE fix: encoder outputs Gaussian, KL pulls toward N(0, 1)
- One forward pass: fast — but blurry samples