From Probability to Variational Autoencoders

ECE 57000 — November 6, 2026

David I. Inouye

Probability distributions are nonnegative and normalized

	Nonnegative	Total probability is one
Discrete mass	$p(\boldsymbol{x}) \geq 0$	$\sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1$
Continuous density	$p(\boldsymbol{x}) \geq 0$	$\int p(\boldsymbol{x})\, d\boldsymbol{x} = 1$

Marginalize: remove $\boldsymbol{z}$ from a joint distribution by summing or integrating it out.

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) \quad (\text{discrete codes}), \qquad p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z} \quad (\text{continuous codes}).$$

Zero values are allowed. A density can exceed one; its **integral** must equal one.

Conditioning and Bayes' rule rearrange the same joint distribution

Conditional distributions (where the denominators are positive):

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z})}, \qquad p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x})}.$$

Joint decomposition / chain rule:

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x} \mid \boldsymbol{z}) = p(\boldsymbol{x})p(\boldsymbol{z} \mid \boldsymbol{x}).$$

Bayes' rule:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}.$$

The variables keep their meanings: $\boldsymbol{x}$ is an image; $\boldsymbol{z}$ is a code.

KL compares distributions by averaging a log-density ratio

The **Kullback–Leibler divergence** is

$$D_{\mathrm{KL}}(q\|p) = \mathbb{E}_{\mathbf{z}\sim q}\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right].$$

Average under q: disagreement where q puts weight matters.

1. **Nonnegative:** $D_{\mathrm{KL}}(q\|p) \geq 0$.
2. **Zero when the distributions agree:** $D_{\mathrm{KL}}(q\|q) = \mathbb{E}_{\mathbf{z}\sim q}[\log 1] = 0$.

It is directional: generally $D_{\mathrm{KL}}(q\|p) \neq D_{\mathrm{KL}}(p\|q)$.

For two outcomes, $q = (0.75, 0.25)$ and $p = (0.5, 0.5)$ give $D_{\mathrm{KL}}(q\|p) \approx 0.131$; setting $p = q$ gives zero.

We will use KL to compare **code distributions**, including their centers and spreads.

Probabilistic autoencoders

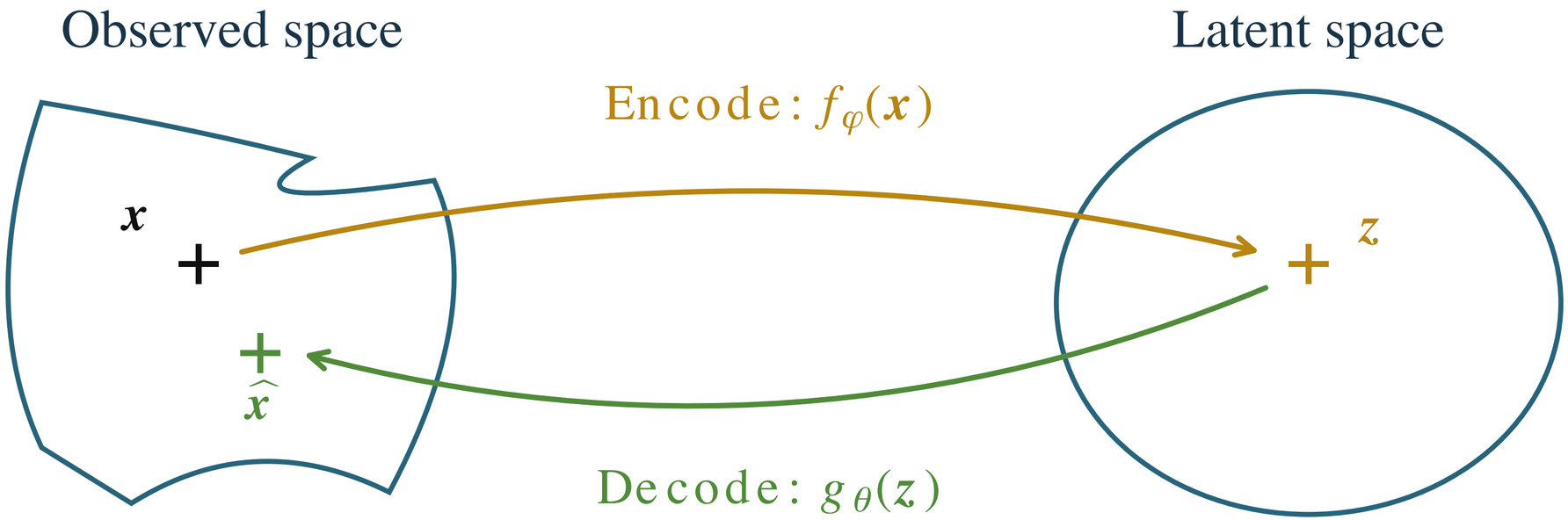
Replace point outputs with distributions

We can make both encoder and decoder probabilistic

1. **Probabilistic AE:** replace each point output with a distribution.
2. **VAE generative model:** add an assumption about where latent codes come from.
3. **VAE training:** fit the generative model using the encoder to infer codes.

First, encode a point into latent space, then decode back into the **same observed space**.

A deterministic AE maps one point to one point

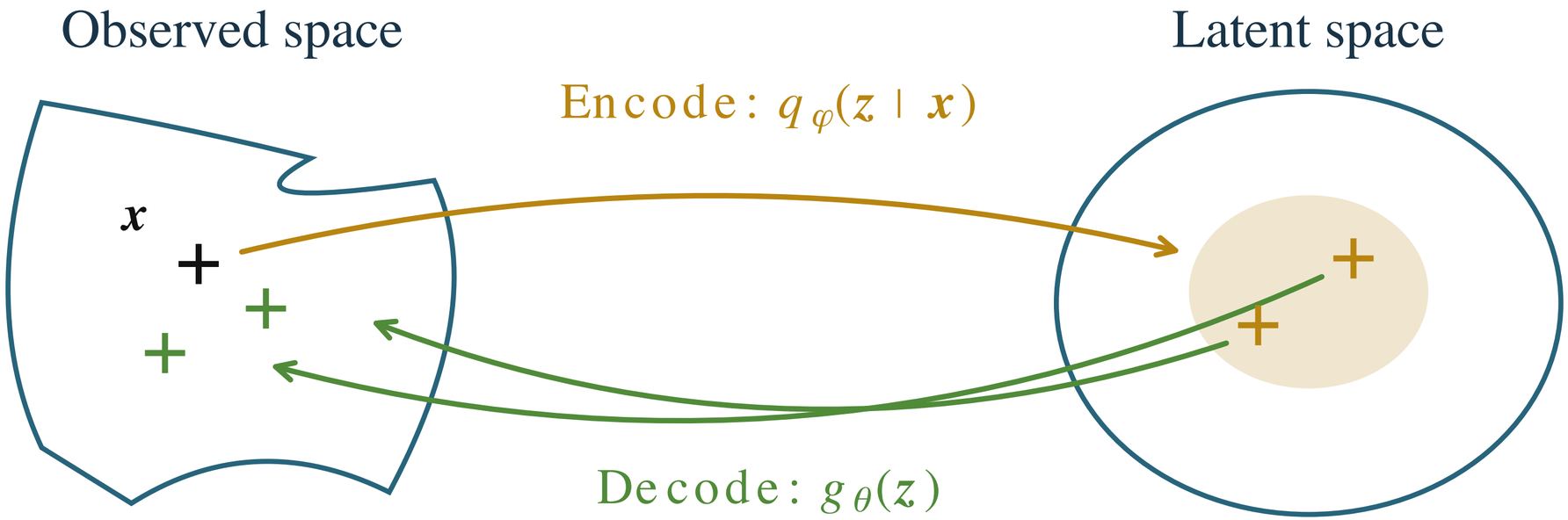


The reconstruction returns to the same observed space.

$$\boldsymbol{z} = f_\phi(\boldsymbol{x}), \qquad \widehat{\boldsymbol{x}} = g_\theta(\boldsymbol{z}).$$

Each **+** is one example or code. The outlines show the spaces where they live.

A probabilistic encoder maps an image to a distribution of codes

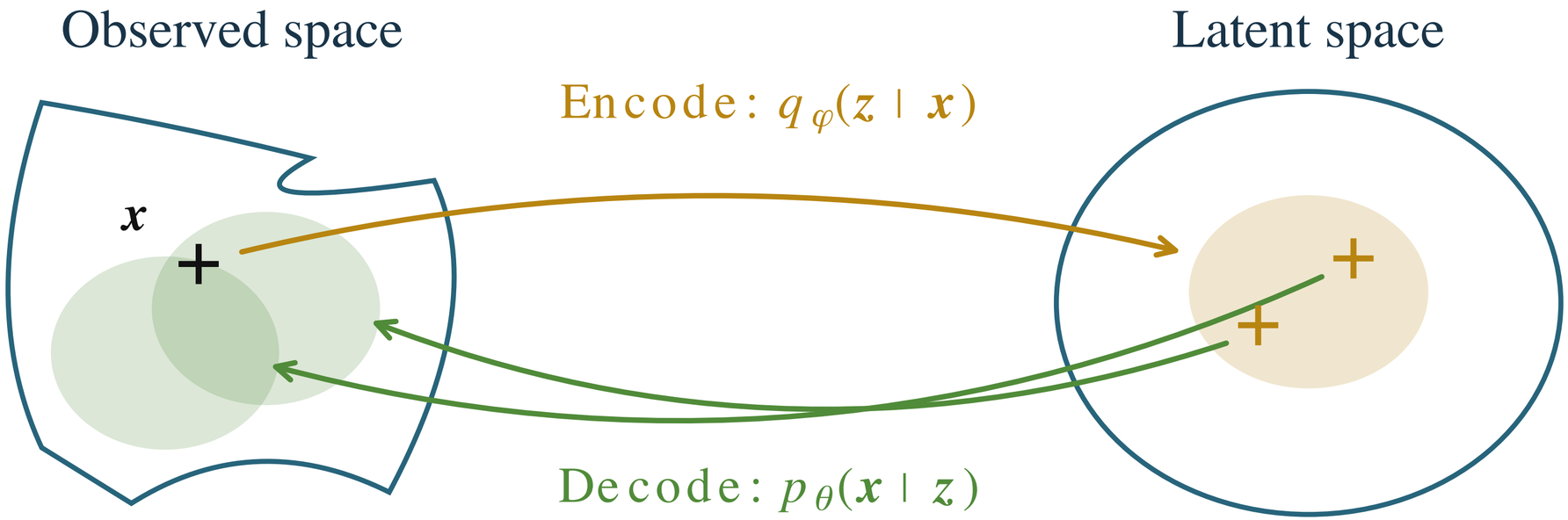


Sample latent codes; each decodes to a point.

Replace $z = f_\phi(x)$ with a conditional distribution $q_\phi(z \mid x)$.

The yellow circle is a code distribution; its **+ marks are sampled codes**. The decoder still returns one point per code to the original observed space.

A probabilistic decoder maps each code to a distribution of images



Training: evaluate the original + under each green distribution.

Replace $\widehat{x} = g_\theta(z)$ with $p_\theta(x \mid z)$.

During training, measure **the likelihood of the original input image** under the distributions represented by the green circles. We sample latent codes, **not new images from the decoder**.

Gaussian mappings let the networks predict the distribution means

For a first example, hold both covariance matrices fixed at the identity:

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; f_\phi(\boldsymbol{x}), I) \,.$$

$$p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; g_\theta(\boldsymbol{z}), I) \,.$$

- f_ϕ predicts the center of the latent distribution.
- g_θ predicts the center of the image distribution.
- We can also learn the spreads; first keep them fixed to see the mechanism.

A probabilistic AE can train with expected reconstruction loss

For one observed image, average over sampled codes:

$$\mathcal{J}_{\mathrm{PAE}}(\boldsymbol{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\cdot|\boldsymbol{x})}[-\log p_\theta(\boldsymbol{x} \mid \mathbf{z})].$$

With the fixed Gaussian mappings, one noise draw gives the estimate

$$\tfrac{1}{2}\|\boldsymbol{x} - g_\theta(f_\phi(\boldsymbol{x}) + \boldsymbol{\epsilon})\|_2^2 + C, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I).$$

Evaluate the original $\boldsymbol{x}$ under the green decoder distribution. **Sample codes, not output images.**

Where does the noise enter these two autoencoders?

> **Both models reconstruct the original clean image.**
> **Which description matches the models we just defined?**
> **A. Denoising AE: noisy input; probabilistic AE: sampled latent code.**
> **B. Denoising AE: sampled latent code; probabilistic AE: noisy input.**
> **C. Both models must use the noisy input as their target.**

A. We changed the input for denoising, then changed the mappings for probabilistic autoencoding.

VAEs: a generative model

Pair a latent prior with a probabilistic decoder

A prior and decoder define an assumed image-generation process

- We **assume a generative process** as a way to model images; it need not match how real images are created.
- Pair a probabilistic decoder with a **prior** on latent codes to define a two-stage sampling process:

1. Sample a latent code: $\mathbf{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, I)$.
2. Sample an image: $\mathbf{x} \sim p_\theta(\boldsymbol{x} \mid \mathbf{z})$.

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})\, p_\theta(\boldsymbol{x} \mid \boldsymbol{z}).$$

We will show why an encoder is useful in a few slides.

Fitting the generative model means averaging over possible codes

For an observed image, the model assigns density

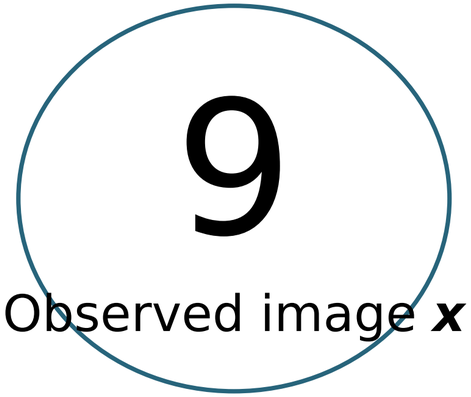
$$p_\theta(\boldsymbol{x}) = \int p(\boldsymbol{z})\, p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\, d\boldsymbol{z}.$$

- We would like to maximize $\sum_i \log p_\theta(\boldsymbol{x}_i)$.
- The code that produced an observed image is unknown.
- With a nonlinear decoder, this integral is generally difficult to compute exactly.

The encoder helps identify codes that could explain each observed image.

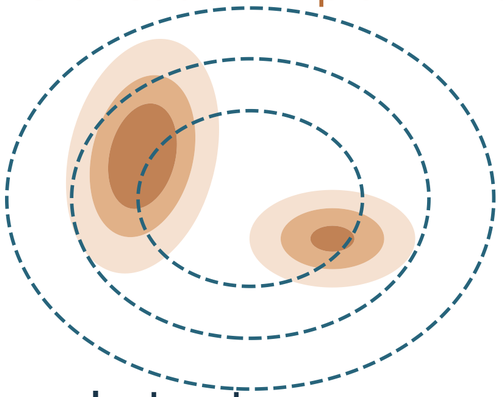
The VAE encoder approximates the codes that explain an image

Observe an image



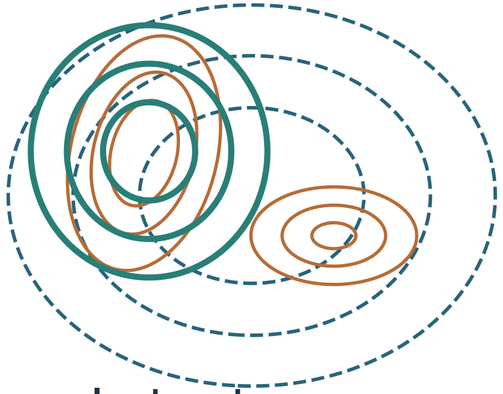
Prior → posterior
$p(\boldsymbol{z}) \rightarrow p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$

Several codes explain this 9



Gaussian approximation
$q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$

Other modes can be missed



Schematic: dashed blue = prior · orange = complex posterior · teal = Gaussian approximation

The **prior** $p(\boldsymbol{z})$ is shared; the **posterior** $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ depends on this image.

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))\big) \quad \text{approximates } p_\theta(\boldsymbol{z} \mid \boldsymbol{x}).$$

Training a VAE

Use the encoder to fit the generative model

Encoder samples let us estimate a usable training objective

For a fixed image, draw codes from $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and define the **evidence lower bound (ELBO)**

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\cdot \mid \boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x}, \mathbf{z}) - \log q_\phi(\mathbf{z} \mid \boldsymbol{x}) \right].$$

- Evaluate the joint with $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z}) p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$.
- These densities are available; the difficult marginal $p_\theta(\boldsymbol{x})$ is absent.
- Average sampled values to estimate $\mathcal{L}$.

Training objective: maximize $\mathcal{L}$ over $ heta,$ (equivalently, minimize $-\mathcal{L}$).

Next, derive why it is a lower bound on image log likelihood.

Multiplying and dividing by the encoder separates the ELBO and its gap

Fix $\boldsymbol{x}$; abbreviate $q = q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$. Every expectation below is under q.

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \mathbb{E}_q[\log p_\theta(\boldsymbol{x})] && \text{(fixed image)} \\
&= \mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right] && \text{(chain rule)} \\
&= \mathbb{E}_q\left[\log \left(\frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\phi(\mathbf{z}|\boldsymbol{x})}\,\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right)\right] && \text{(multiply by } q/q\text{)} \\
&= \mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\phi(\mathbf{z}|\boldsymbol{x})}\right] + \mathbb{E}_q\left[\log \frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right] && \text{(split the log)} \\
&= \mathcal{L}(\theta, \phi; \boldsymbol{x}) + D_{\mathrm{KL}}(q_\phi(\cdot \mid \boldsymbol{x}) \,\|\, p_\theta(\cdot \mid \boldsymbol{x})) && \text{(definitions)}
\end{aligned}
$$

The ELBO gives a lower bound on image log likelihood

For one image, the following identity connects inference and generation:

$$\log p_\theta(\boldsymbol{x}) = \underbrace{\mathcal{L}(\theta, \phi; \boldsymbol{x})}_{\text{ELBO}} + \underbrace{D_{\mathrm{KL}}(q_\phi(\cdot \mid \boldsymbol{x}) \,\|\, p_\theta(\cdot \mid \boldsymbol{x}))}_{\text{gap from approximate inference}}.$$

Because KL is nonnegative, $\mathcal{L}(\theta, \phi; \boldsymbol{x}) \leq \log p_\theta(\boldsymbol{x})$.

- Fit θ and ϕ by maximizing this **evidence lower bound**.
- The bound is tight when the encoder equals the true posterior.