Generative AI and LLM

GenAI Principles and Latent Variable Models
CS5202

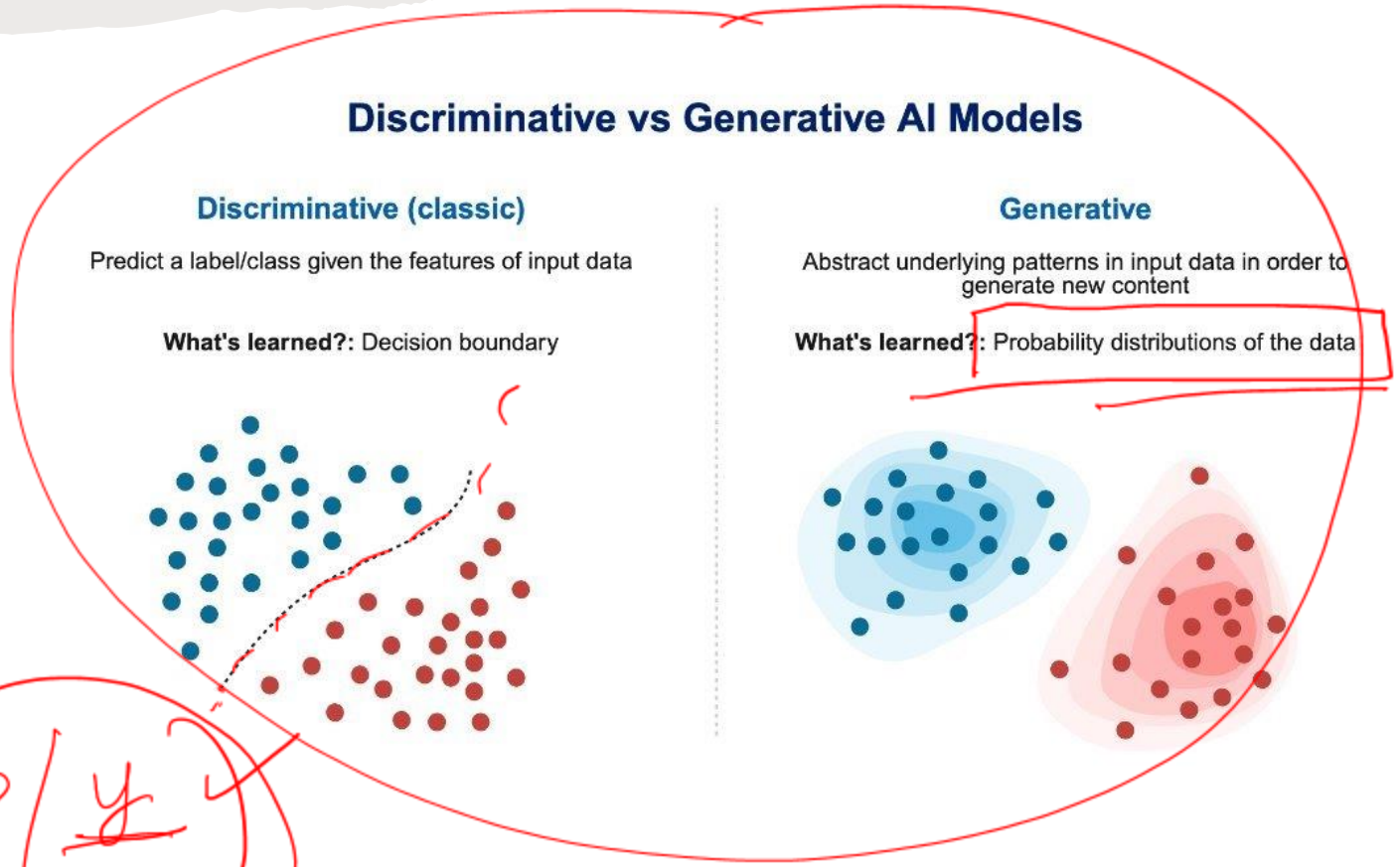
Course Instructor : Dr. Nidhi Goyal

Lecture Plan

- Generative AI Principles
- Latent Variables
- Mathematical Foundations of GenAI
- Entropy
- Cross entropy
- KL divergence

Generative AI

- GenAI Principles:
  - Learn data distribution
  - Generate new samples
- Discriminative → P(y | x)
- Generative → P(x)



What does it mean to “learn data”?

- Data is not just points
- Data comes from an underlying distribution
- Generative models aim to learn this distribution
- Learn **how likely** each data point is
- Once learned → sample new data

$$x \sim p_{\text{data}}(x)$$

$$p_\theta(x) \approx p_{\text{data}}(x)$$

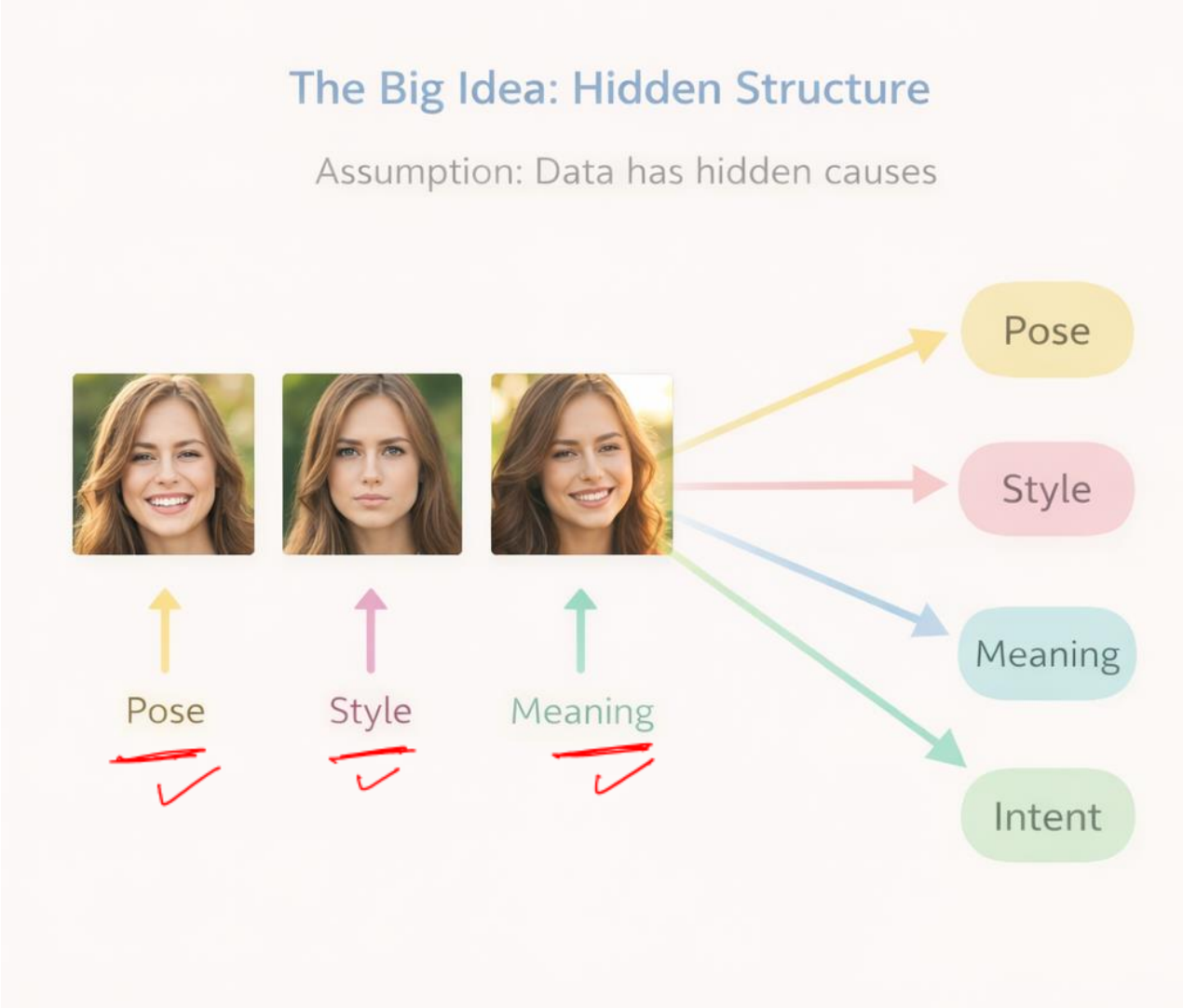
Theta are model parameters

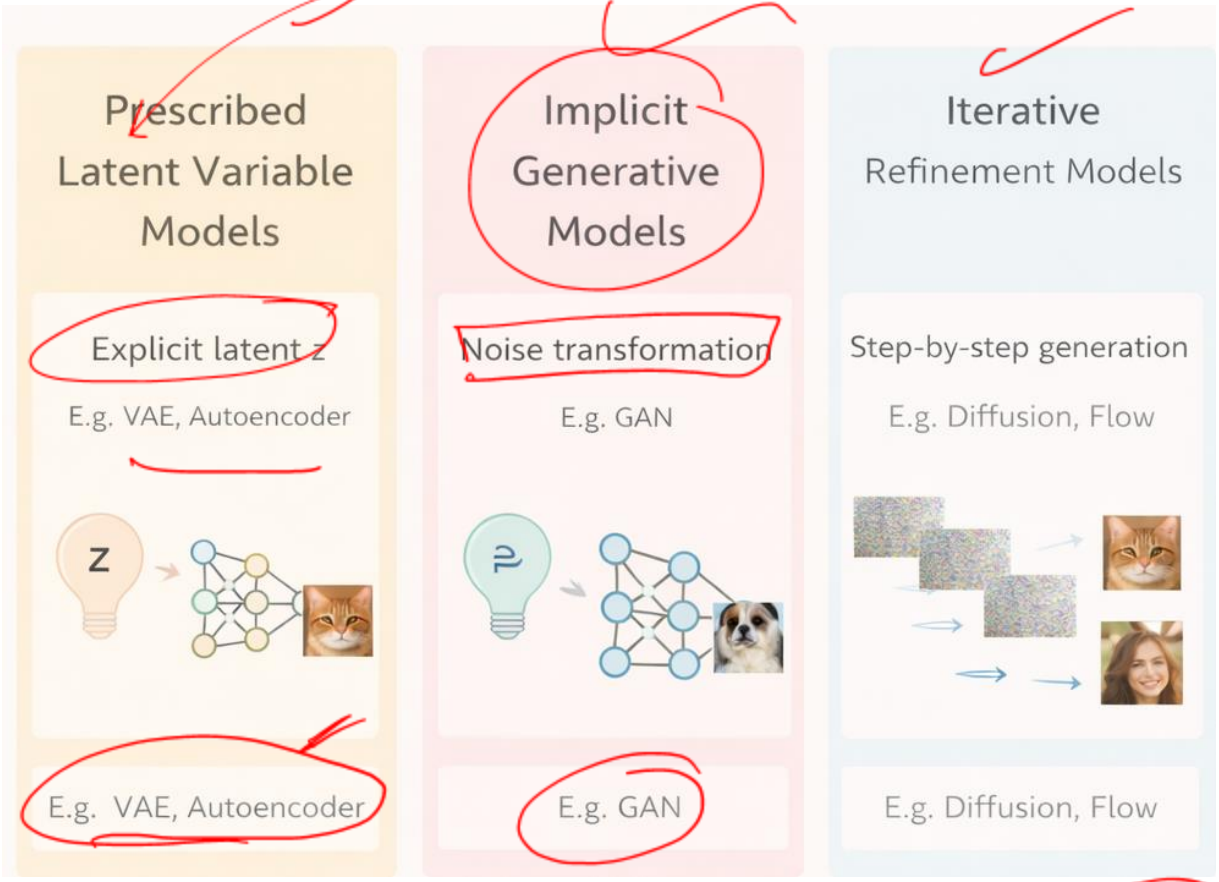
The Generative Modeling Problem

- **Goal:** Given dataset D = {x_1, x_2, ..., x_n}, learn distribution p(x)
- High dimensional data (images: millions of pixels)
- Complex dependencies
- Cannot model directly

The Big Idea: Hidden Structure

- Data has hidden causes
- Introduce Latent Variables
  - Pose
  - Style
  - Meaning
  - Intent





Different approaches to model p(x)

Prescribed latent variable models/likelihood based models

- Explicit latent z with known prior p(z)
- Learn p(x|z)
- Examples: VAE, Autoencoder

Implicit generative models

- Learn transformation from noise to data
- No explicit p(x)
- Examples: GAN

Iterative refinement models

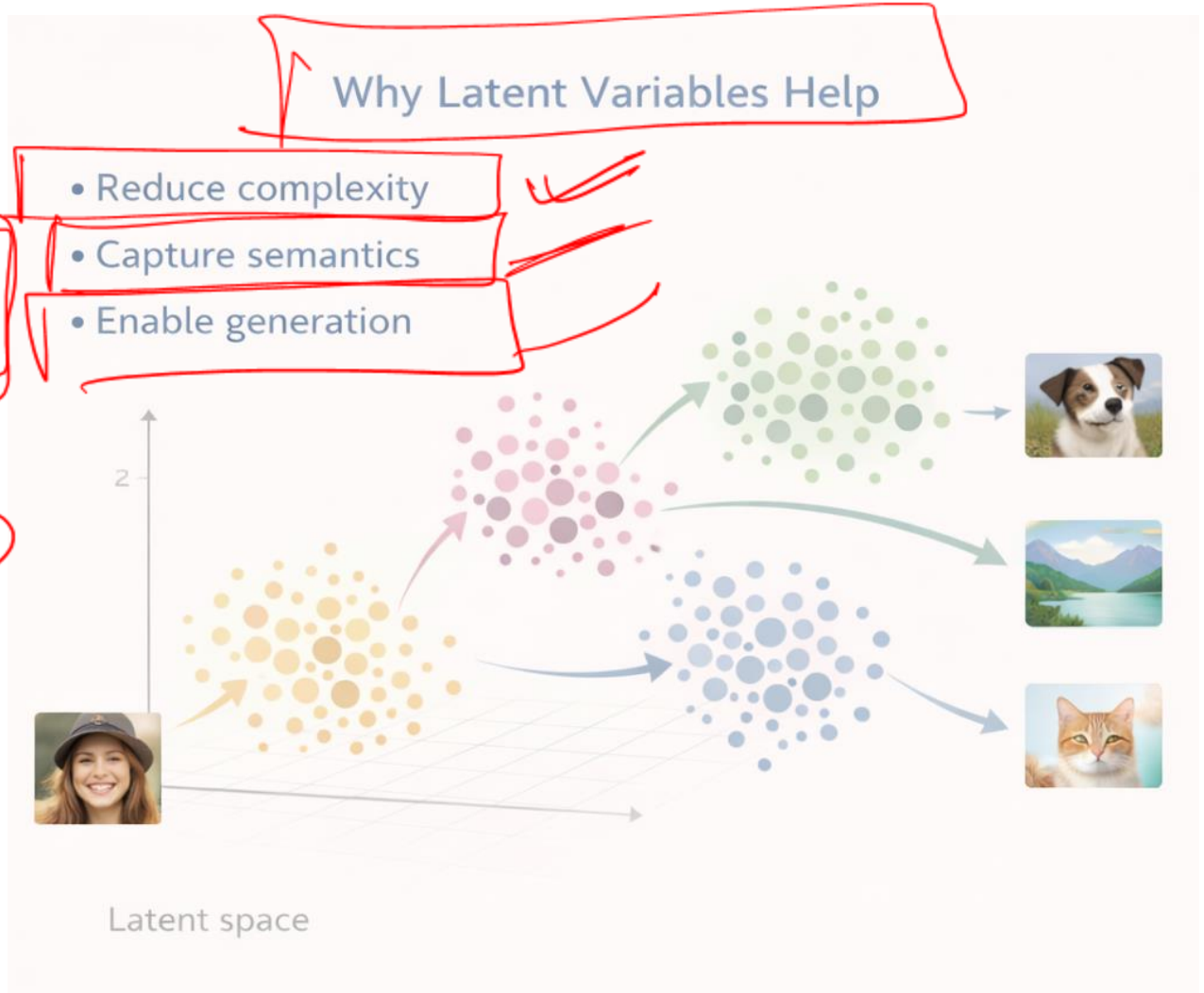
- Sequential generation/denoising
- Examples: Diffusion, Flow

Mathematical Prerequisites
Latent Variables

- What are latent variables?

$$z \sim p(z) \qquad x \sim p_\theta(x \mid z)$$

  - Hidden/unobserved variables that explain data
  - z represents compressed information about x
- Why Latent Variables Help
  - Reduce complexity
  - Capture semantics
  - Enable generation



The Core Challenge

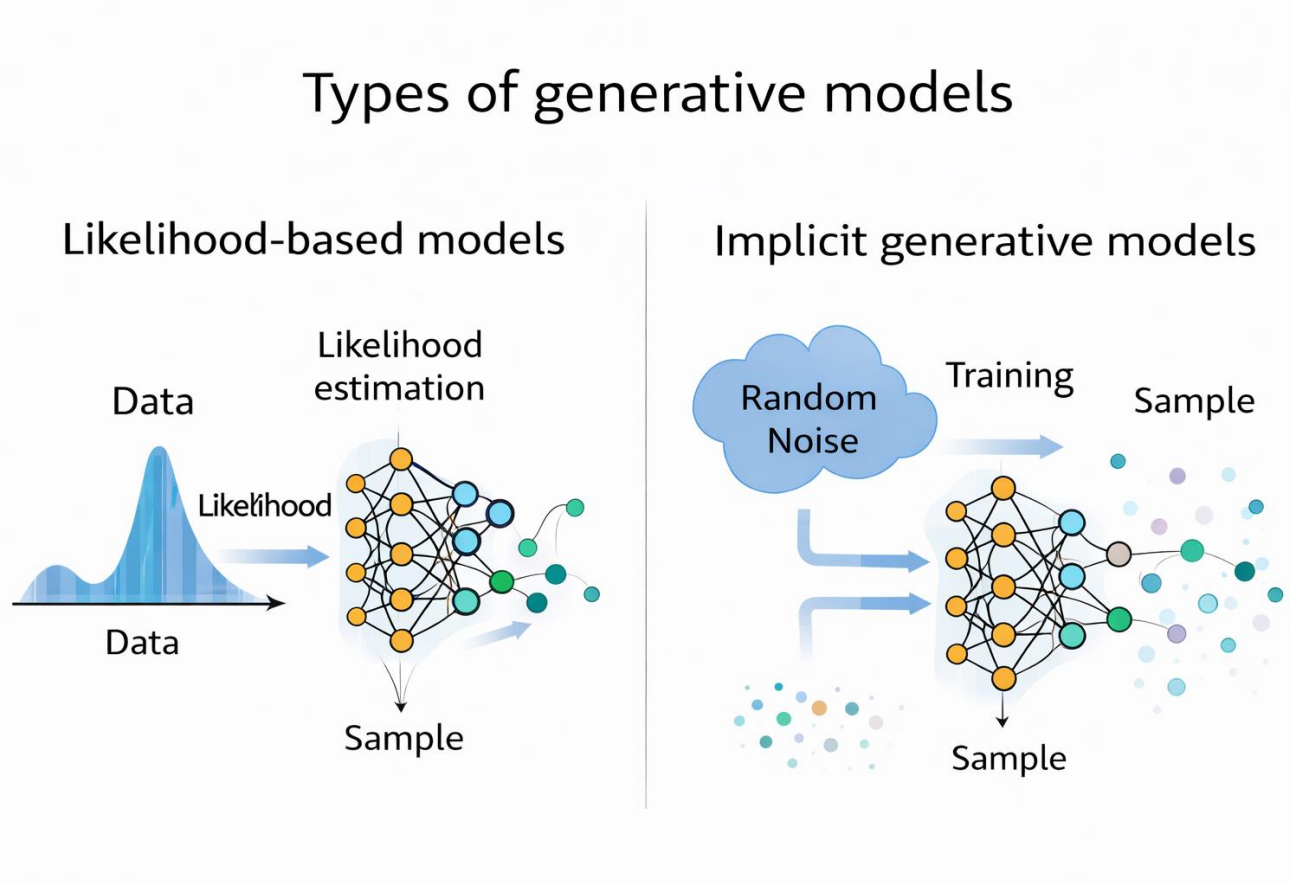
- To compute likelihood:

$$p(x) = \int p_\theta(x \mid z)\, p(z)\, dz$$

- This integral is **intractable**
- Leads to **approximation methods**

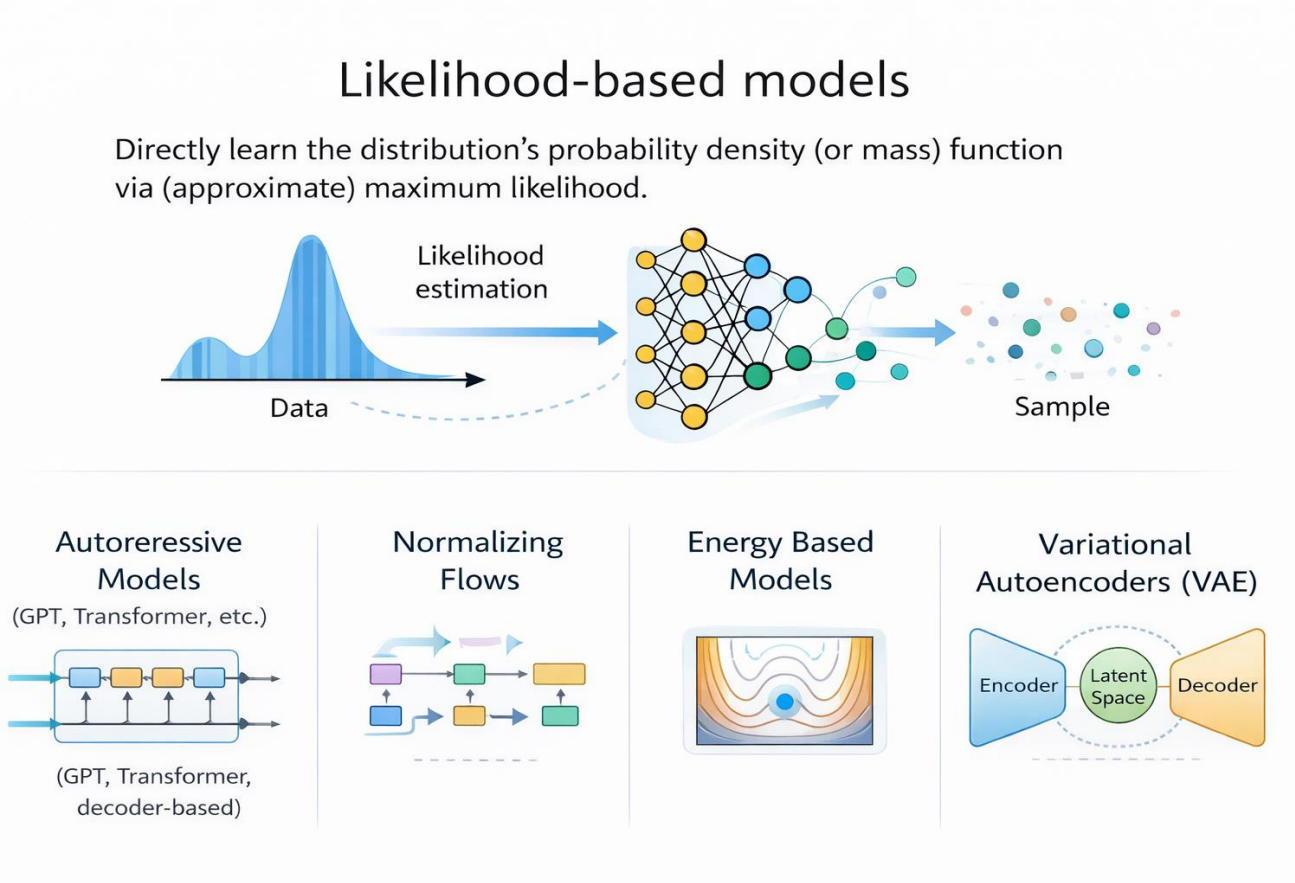
Types of generative models

- Likelihood-based models
- Implicit generative models



Likelihood based models

- Directly learn the distribution's probability density (or mass) function via (approximate) maximum likelihood.
  - Autoregressive models (GPT, Transformer (decoder-based), etc.)
  - Normalizing flows
  - Energy based models
  - Variational Autoencoders (VAE)

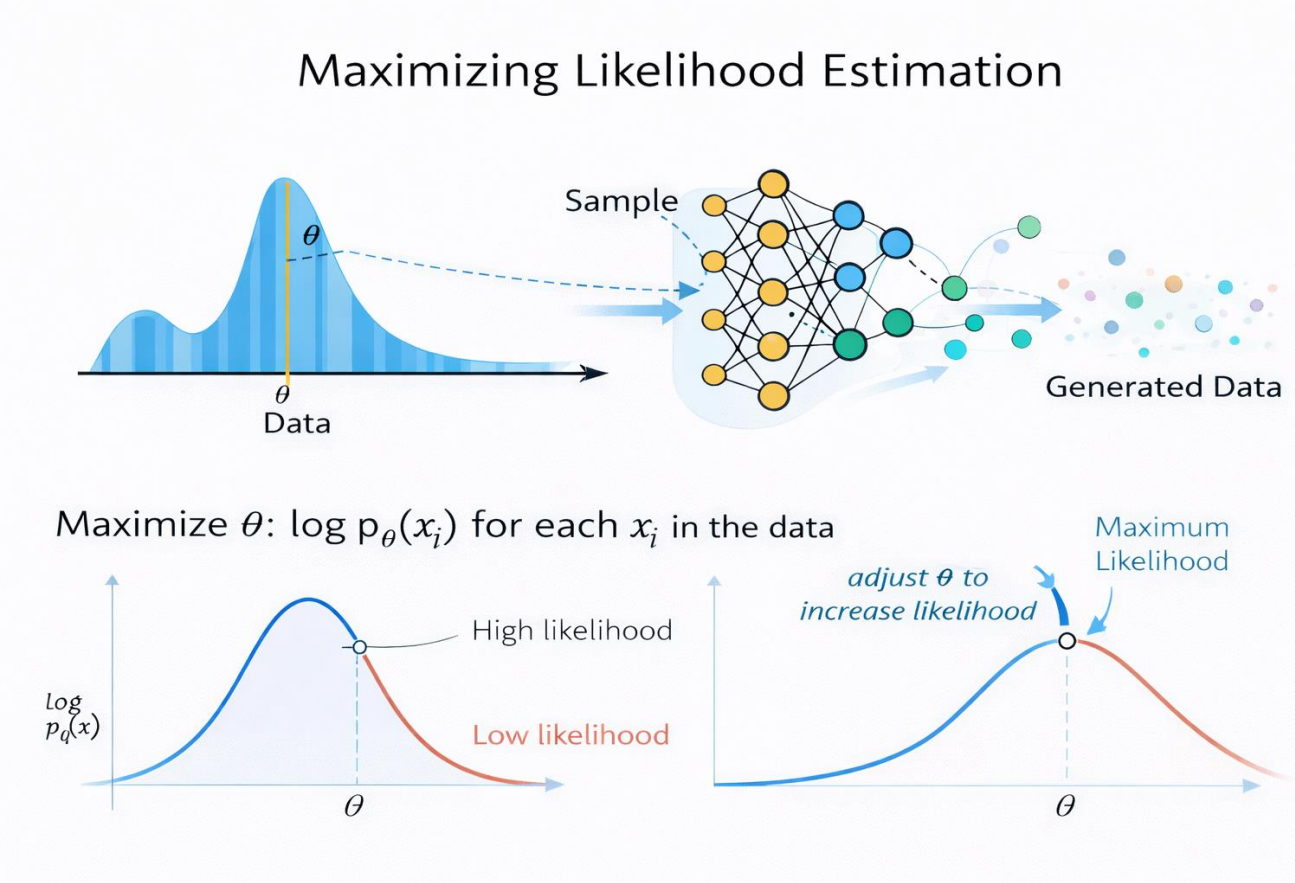


Likelihood based models

- These models learn the probability distribution of data
- They can answer:

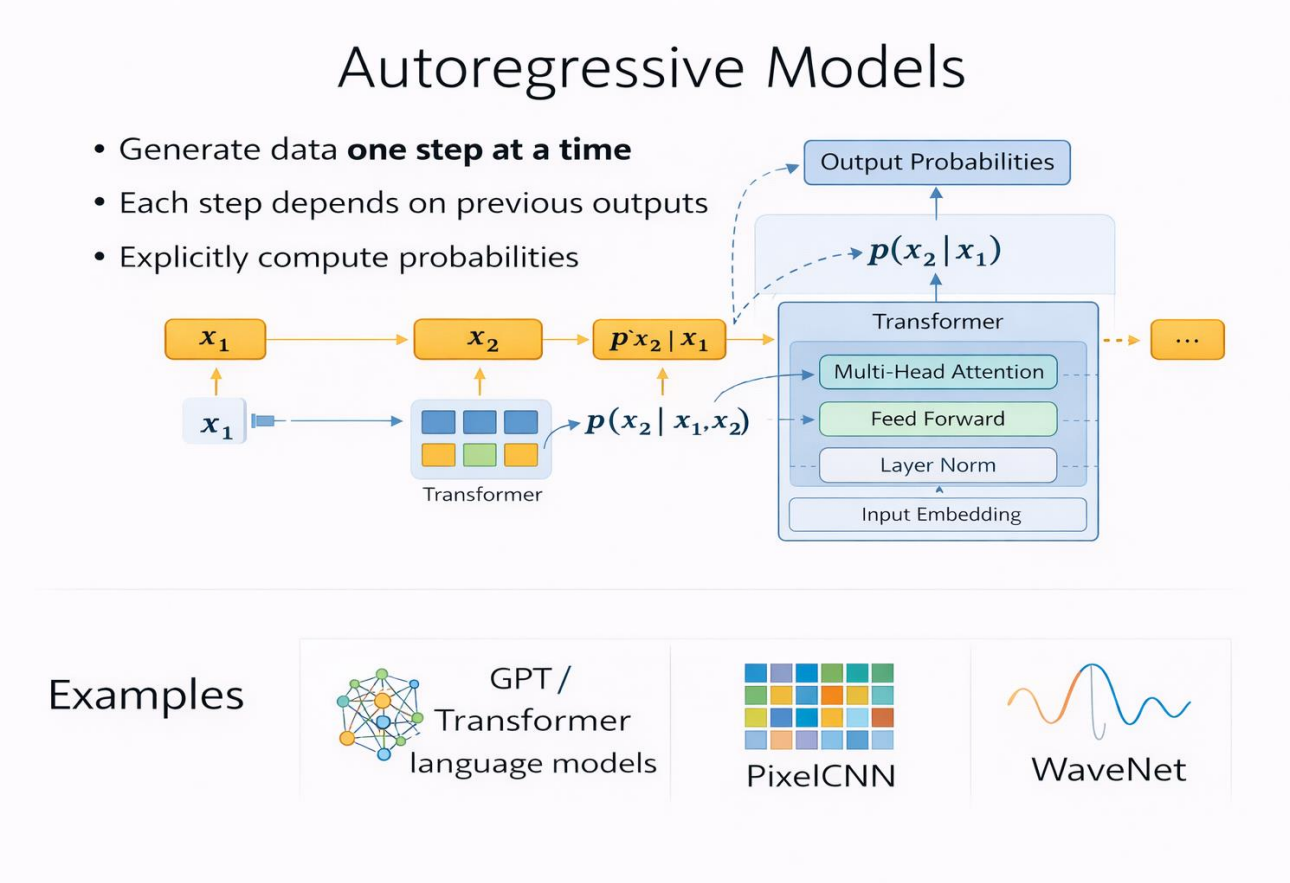
"How likely is this data?"

- Trained by maximizing likelihood



Autoregressive models

- Generate data **one step at a time**
- Each step depends on previous outputs
- Explicitly compute probabilities
- **Examples**
  - GPT / Transformer language models
  - PixelCNN
  - WaveNet



Pre-requisties

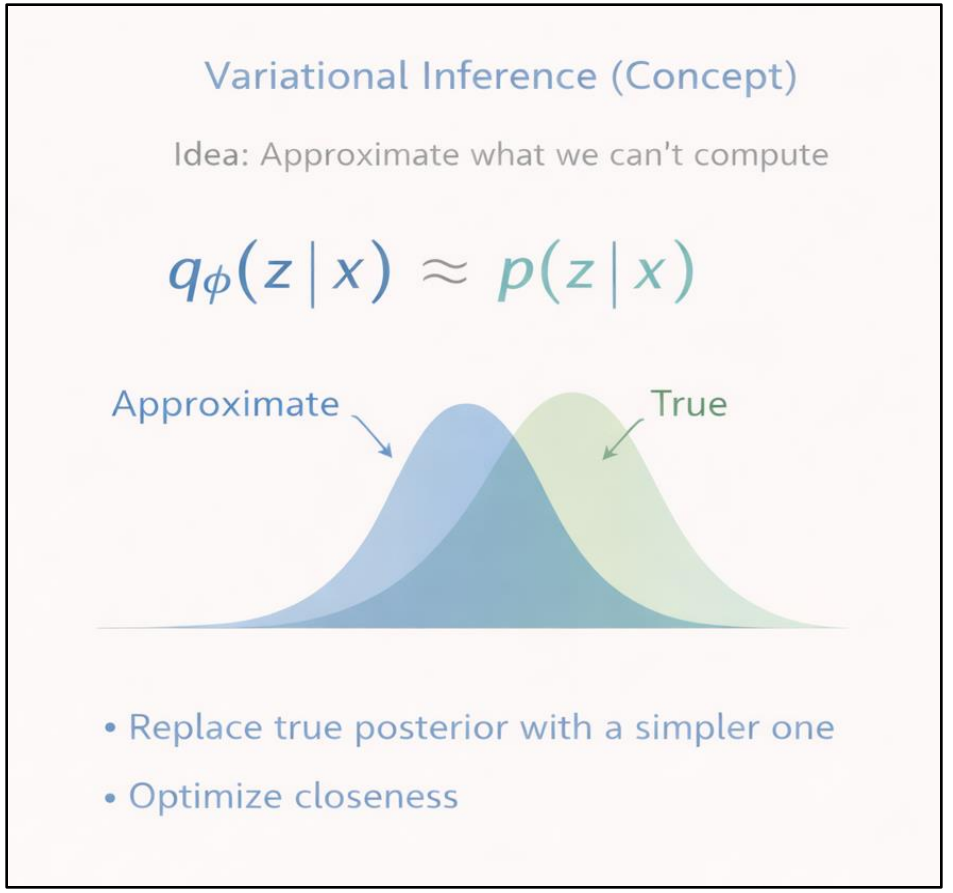
- **Prior**: belief before seeing data
- **Posterior**: belief after seeing data

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

- Variational Inference
  - Approximate what we can't compute

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

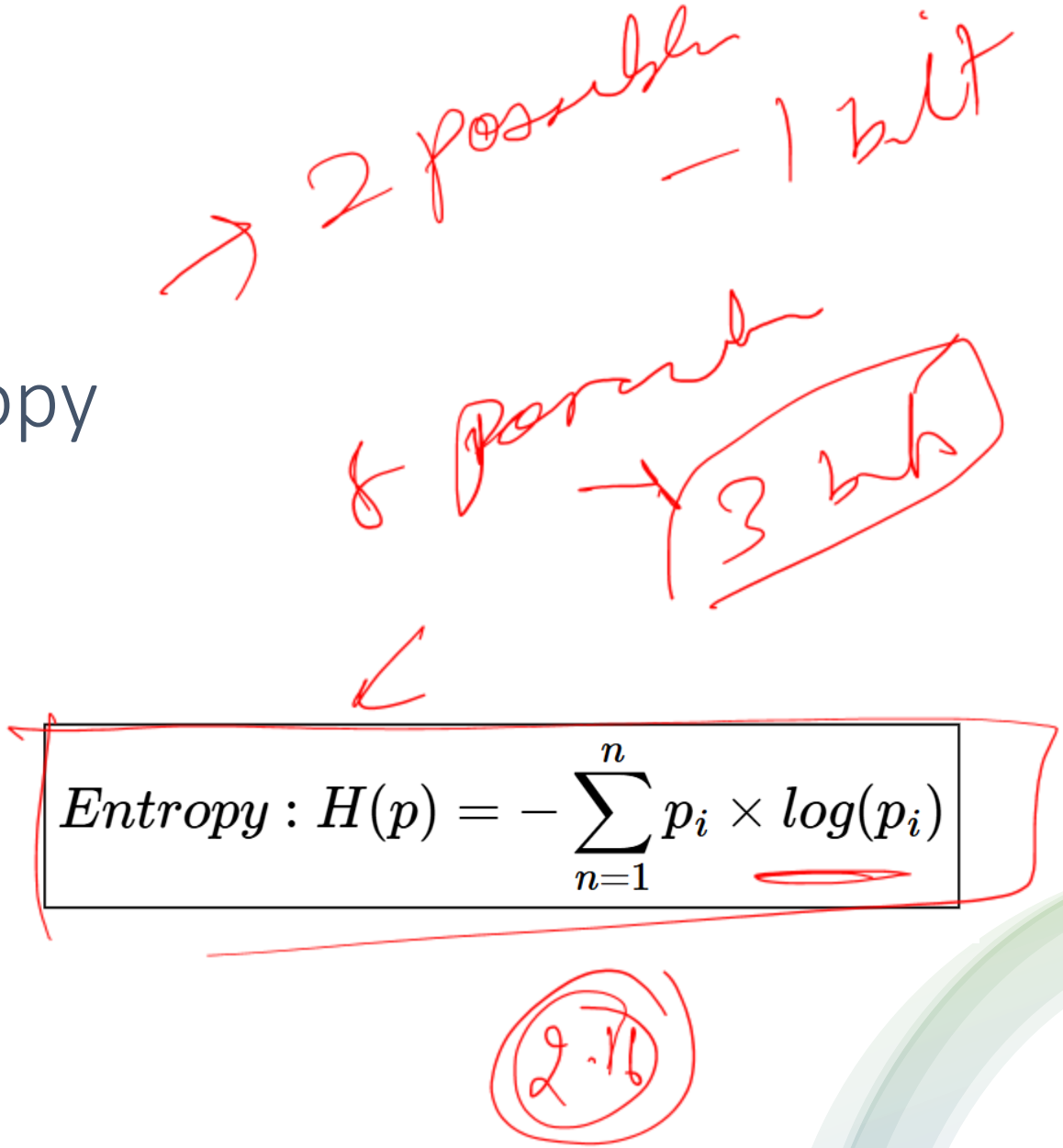
- Explanation
  - Replace true posterior with a simpler one
  - Optimize closeness



Entropy

- According to Shannon, **Entropy is the minimum no of useful bits required to transfer information from a sender to a receiver.**

Entropy (expressed in 'bits') is a measure of how unpredictable the probability distribution is. So more the individual events vary, the more is its entropy.



Entropy 3 bits

In case of equally likely

$$\log\left(\frac{1}{p}\right) = -\log p$$

If 8 possibilities then

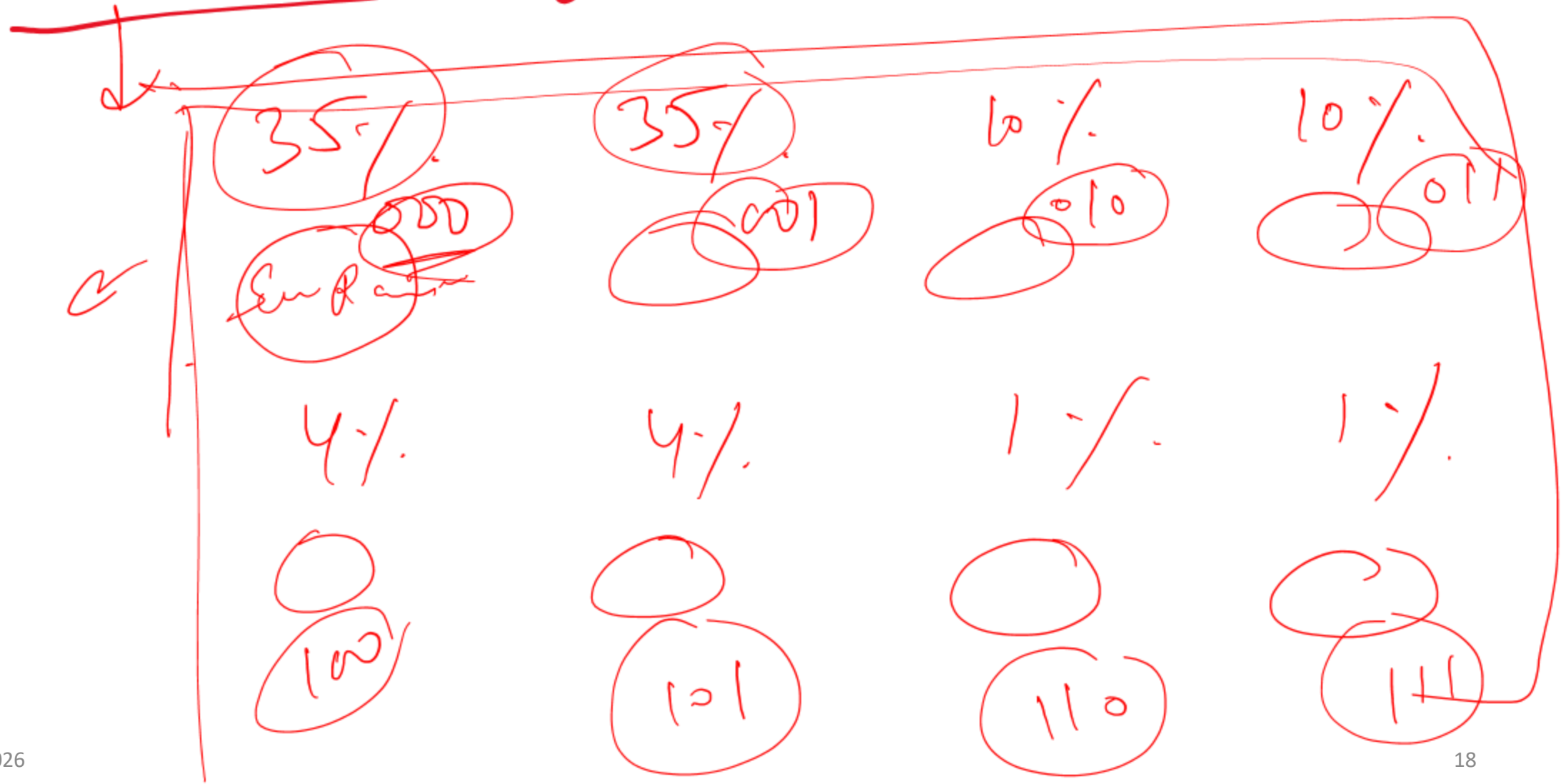
$\log(8) = 3$ bits

$$\text{Cross Entropy} : H(p,q) = -\sum_{n=1}^{n} p_i \times log(q_i)$$

Cross Entropy

- **Cross entropy is the average message length that is used to transmit the message.**

~~Gini~~ entropy (3 bits)



Measuring “Closeness”: KL Divergence

- **The amount by which the cross-entropy exceeds the entropy is called Relative Entropy or commonly known as Kullback-Leibler Divergence or KL Divergence.**

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{\mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

- Used to quantify the difference between one probability distribution from a reference probability distribution

References

- **Vaswani et al., 2017 — *Attention Is All You Need***
  https://arxiv.org/abs/1706.03762
- **Goodfellow et al., 2014 — *Generative Adversarial Networks***
  https://arxiv.org/abs/1406.2661
- **Kingma & Welling, 2013 — *Auto-Encoding Variational Bayes***
  https://arxiv.org/abs/1312.6114
- **Jumper et al., 2021 — *Highly Accurate Protein Structure Prediction with AlphaFold***
  https://www.nature.com/articles/s41586-021-03819-2
- **Shen et al., 2019 — *Deep Image Reconstruction from Human Brain Activity***
  https://www.nature.com/articles/s41593-019-0389-0