

# Diffusion Models: Theory and Applications

## Lecture 3: Mathematical Foundations of Generative Models

Shubham Chatterjee

Missouri University of Science and Technology, Department of Computer Science

June 10, 2025

We discovered the power of gradual generation...

**But how do we actually make this work mathematically?** 🧠

- ✓ **Conceptual insight:** Small steps are easier than big leaps
- ✓ **Intuitive understanding:** Reverse destruction step by step
- ? **Missing piece:** The mathematical framework to make it trainable

**Today: We build the mathematical toolkit for ALL generative models** 🧰

# The Natural Learning Approach: Maximum Likelihood


How should we train a generative model?

## The Standard Machine Learning Recipe

- 1 **Collect data:**  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  (e.g., millions of images)
- 2 **Choose model:**  $p_\theta(\mathbf{x})$  parameterized by  $\theta$  (neural network weights)
- 3 **Define objective:** Maximize likelihood of observed data
- 4 **Optimize:** Find  $\theta^* = \arg \max_\theta \prod_{i=1}^N p_\theta(\mathbf{x}_i)$

# The Natural Learning Approach: Maximum Likelihood

How should we train a generative model?

Taking the logarithm (standard practice) 

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i)$$

**Goal:** Make our model assign high probability to real data

Sounds straightforward... but there's a catch! 

**Most interesting generative models involve hidden variables...**

## The Latent Variable Story

**For image generation:**

- **x**: Observable image (what we see)
- **z**: Hidden factors that generated it (lighting, pose, style, content, ...)
- Model:  $p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})$

**Most interesting generative models involve hidden variables...**

## The Computational Crisis

To compute likelihood of observed image  $\mathbf{x}$ , we need:

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

**Must integrate over ALL possible hidden factors that could have generated this image!**

Every generative model faces the same fundamental problem...

## What we want:

- Learn a model that generates realistic data
- Train on observed samples
- Use maximum likelihood estimation
- Straightforward optimization

## What we get:

- Intractable likelihood computations
- High-dimensional integrals
- No closed-form solutions
- Optimization nightmares

Every generative model faces the same fundamental problem...

The Core Problem (From Maximum Likelihood)

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

This integral is computationally impossible for most interesting models!

The tools we learn today solve this universal challenge 



## 🕒 Understand the mathematical foundations that power modern generative AI

### Essential Tools We'll Master ✓

- **Marginal distributions:** Why some things can't be computed directly
- **Expected values:** How to approximate intractable quantities
- **Bayes' rule:** The foundation of learning from data
- **KL divergences:** How to measure "closeness" of distributions
- **Jensen's inequality:** The key to creating tractable bounds
- **Variational inference:** The strategy that makes it all work

**These tools work for VAEs, GANs, diffusion models, and beyond! ★**

Why some probabilities are impossible to compute...

## Definition (Marginal Distribution)

The marginal distribution of a subset of variables is obtained by integrating over all other variables:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

## Intuitive Understanding

- Think of  $\mathbf{z}$  as “hidden factors” that influence  $\mathbf{x}$
- $p(\mathbf{x})$  asks: “What’s the probability of  $\mathbf{x}$  regardless of what  $\mathbf{z}$  is?”
- We “marginalize out”  $\mathbf{z}$  by considering all possible values, weighted by probability
- Example: Image Generation.  $\mathbf{x}$  = A face image (observable);  $\mathbf{z}$  = Abstract factors like lighting, expression, age (hidden);  $p(\mathbf{x})$  = Probability of this face across all possible hidden factors

# Why Marginal Distributions Are Intractable

The curse of dimensionality strikes...

## The Computational Nightmare

For typical generative models:

- $\mathbf{z}$  is high-dimensional (64, 128, or 512 dimensions)
- Need to integrate over  $\mathbb{R}^{64}$ ,  $\mathbb{R}^{128}$ , or  $\mathbb{R}^{512}$
- No closed-form analytical solutions
- Numerical integration fails due to curse of dimensionality

## Monte Carlo Estimation Also Fails

- Try:  $p(\mathbf{x}) \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_k)$  where  $\mathbf{z}_k \sim p(\mathbf{z})$
- Problem: Most random  $\mathbf{z}$  values give  $p(\mathbf{x}|\mathbf{z}_k) \approx 0$
- Need astronomically many samples to get meaningful estimates
- Like finding a needle in a haystack by random sampling

### How do we approximate intractable expectations?

#### Definition (Expected Value)

The expected value is the average value of a function, weighted by probability:

$$\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

### How do we approximate intractable expectations?

#### Monte Carlo Approximation

When we can sample from  $p(\mathbf{x})$ , we can approximate:

$$\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad \text{where } \mathbf{x}_i \sim p(\mathbf{x})$$

#### Why This Matters

- Our bounds will involve expectations we can't compute analytically
- Monte Carlo lets us approximate these with samples
- **Key insight:** Draw samples and average—much more tractable!
- Connects intractable theory to practical computation

# In-Class Exercise 1: Understanding Marginalization

## Think About Image Generation

Imagine generating face images where:

- $x$ : The final face image (what we observe)
- $z$ : Hidden factors [lighting, age, expression, hair color, ...]

## Discussion Questions:

- 1 Why can't we directly compute  $p(x)$  for a specific face?
- 2 What would it mean to "integrate over all possible hidden factors"?
- 3 Why does random sampling fail to estimate this probability?

Think about it 

**Hint:** How many different combinations of lighting, age, expression could produce the same face?

**Discuss with your neighbor for 3 minutes!** 

# Exercise 1 Solution: The Marginalization Challenge ✓

## 1. Why we can't compute $p(\mathbf{x})$ directly ☹

- Need to consider ALL possible combinations of hidden factors
- Lighting: continuous range of angles, intensities
- Age: continuous variable
- Expression: infinite subtle variations
- Result: Integral over infinite-dimensional space!

# Exercise 1 Solution: The Marginalization Challenge ✓

## 2. “Integrating over hidden factors” means

- For every possible (*lighting*, *age*, *expression*, ...) combination
- Compute: probability of that combination  $\times$  probability it generates this face
- Sum/integrate over ALL such combinations
- Computationally impossible for high-dimensional spaces

## 3. Why random sampling fails

- Most random combinations produce faces very different from target
- Only tiny fraction of factor space generates anything close to our image
- Like trying to hit a microscopic target by throwing darts blindfolded



How we learn about hidden factors from observed data...

## Bayes' Rule

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

## Breaking Down Each Component

- $p(\mathbf{z}|\mathbf{x})$  (**Posterior**): What we want—hidden factors given observed data
- $p(\mathbf{x}|\mathbf{z})$  (**Likelihood**): How likely the data is, given hidden factors
- $p(\mathbf{z})$  (**Prior**): Our belief about hidden factors before seeing data
- $p(\mathbf{x})$  (**Evidence**): Total probability of data—the intractable part!

How we learn about hidden factors from observed data...

## Bayes' Rule

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

## The Circular Problem

- We want to learn about hidden factors: need  $p(\mathbf{z}|\mathbf{x})$
- But computing posterior requires  $p(\mathbf{x})$  in denominator
- $p(\mathbf{x})$  is the same intractable marginal we started with!
- **Circular dependency:** Can't compute what we need to learn

How do we measure how “different” two distributions are?

## Definition (KL Divergence)

$$D_{KL}(p\|q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})} \left[ \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right]$$

## Key Properties

- $D_{KL}(p\|q) = 0$  if and only if  $p = q$  (distributions identical)
- $D_{KL}(p\|q) > 0$  when  $p \neq q$  (always non-negative)
- **Asymmetric:**  $D_{KL}(p\|q) \neq D_{KL}(q\|p)$  in general
- **Interpretation:** Extra information needed when using  $q$  to approximate  $p$

**How do we measure how “different” two distributions are?**

## Definition (KL Divergence)

$$D_{KL}(p\|q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})} \left[ \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right]$$

## Why KL Divergence Matters ★

- Measures how well one distribution approximates another
- Will be crucial for comparing learned vs. true distributions
- Forms the backbone of our tractable objectives



# In-Class Exercise 2: KL Divergence Intuition

## Understanding KL Divergence

Consider two ways of modeling the “happiness level” in face images:

- **Model A:**  $p = \mathcal{N}(0.5, 0.1^2)$  (centered, narrow)
- **Model B:**  $q = \mathcal{N}(0.3, 0.3^2)$  (shifted, wide)

## Questions:

- 1 Without computing, which do you think is larger:  $D_{KL}(p||q)$  or  $D_{KL}(q||p)$ ?
- 2 What does each KL divergence measure in this context?
- 3 Which would be better for approximating rare “very happy” faces?

Think about it 

**Hint:** Consider what happens in regions where one distribution is much larger than the other

**Discuss your intuition!** 

## Exercise 2 Solution: KL Asymmetry Insights ✓

$D_{KL}(p||q)$ : “Model A using Model B” →

- Large penalty when  $p$  is big but  $q$  is small
- Model B (wide) will try to cover all of Model A (narrow)
- Results in **mode-covering** behavior
- Better for capturing full range, but may be diffuse

$D_{KL}(q||p)$ : “Model B using Model A” ←

- Large penalty when  $q$  is big but  $p$  is small
- Model A (narrow) will focus on peak of Model B (wide)
- Results in **mode-seeking** behavior
- Better for sharp approximations, but may miss parts

## Exercise 2 Solution: KL Asymmetry Insights ✓

For rare “very happy” faces 😊

$D_{KL}(p||q)$  (mode-covering) would be better because the wide model B can capture rare events that narrow model A might miss

**This asymmetry will be crucial for understanding different training objectives!**

# Essential Tool 5: Jensen's Inequality - The Bound Creator

The mathematical tool that transforms impossible into tractable...

## Definition (Convex Function)

A function  $f$  is convex if for all  $\mathbf{x}_1, \mathbf{x}_2$  and  $t \in [0, 1]$ :

$$f(t\mathbf{x}_1 + (1 - t)\mathbf{x}_2) \leq tf(\mathbf{x}_1) + (1 - t)f(\mathbf{x}_2)$$

*The function curves upward (like a bowl)*

## Theorem (Jensen's Inequality)

If  $f$  is convex:  $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

If  $f$  is concave:  $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$

## The Key Insight

The logarithm is **concave**, so:  $\log(\mathbb{E}[X]) \geq \mathbb{E}[\log(X)]$

**This inequality creates our tractable lower bounds!**



# Why Jensen's Inequality Creates Lower Bounds ↓

## Concave Function Behavior 🎯

For concave  $f$  (like  $\log$ ):

- Curves downward (upside-down bowl)
- Average of function values  $\leq$  function of average ( $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$ )
- Rearranging:  $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$

## Concrete Example 🧮

Let  $X \in \{1, 4\}$  with equal probability:

- $\log(\mathbb{E}[X]) = \log(2.5) \approx 0.916$
- $\mathbb{E}[\log(X)] = \frac{\log(1) + \log(4)}{2} \approx 0.693$
- Indeed:  $0.916 > 0.693$

# Why Jensen's Inequality Creates Lower Bounds: Geometric Intuition ↓

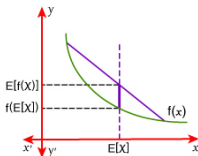
## Jensen's Inequality



States that if 'X' is an integrable random variable and  $f: \mathbb{R} \rightarrow \mathbb{R}$  is a convex or concave function, then

For Convex Function

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$



For Concave Function

$$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$$

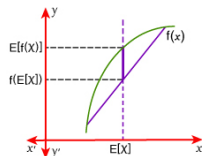


Figure: Jensen's Inequality: For a concave function like  $\log$ ,  $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$

**Jensen's inequality gives us:  $\log(\text{intractable}) \geq \text{tractable bound}$  📁**

# In-Class Exercise 3: Jensen's Inequality Practice

## Apply Jensen's Inequality

Given a random variable  $X$  that takes values  $\{2, 8\}$  with equal probability.

Calculate:

- 1  $\mathbb{E}[X] = ?$
- 2  $\log(\mathbb{E}[X]) = ?$
- 3  $\mathbb{E}[\log(X)] = ?$
- 4 Verify Jensen's inequality: which side is larger?

## Quick Poll

**Before calculating:** Which do you think will be larger?

- A)  $\log(\mathbb{E}[X])$
- B)  $\mathbb{E}[\log(X)]$
- C) They're equal

Work it out step by step! 

## Exercise 3 Solution: Jensen's Inequality in Action ✓

**Given:**  $X \in \{2, 8\}$  with equal probability

①  $\mathbb{E}[X] = \frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 8 = 1 + 4 = 5$

②  $\log(\mathbb{E}[X]) = \log(5) \approx 1.609$

③  $\mathbb{E}[\log(X)] = \frac{1}{2} \log(2) + \frac{1}{2} \log(8) = \frac{1}{2}(0.693) + \frac{1}{2}(2.079) \approx 1.386$

④  $\log(\mathbb{E}[X]) = 1.609 > 1.386 = \mathbb{E}[\log(X)]$

**Jensen's inequality confirmed!** The function of the average is larger than the average of the function.

Now we combine all these tools to solve our intractable integral...

## The Strategy

**Goal:** Bound the intractable  $\log p(\mathbf{x})$

**Tool:** Jensen's inequality with a carefully chosen expectation

**Key insight:** If you can't compute it exactly, bound it cleverly!

# The Variational Inference Strategy

## Step 1: The Variational Trick

For any distribution  $q(\mathbf{z})$ , we can write:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} d\mathbf{z}$$

## Why This “Pointless” Step Matters

- We’re multiplying by  $\frac{q(\mathbf{z})}{q(\mathbf{z})} = 1$ —seems useless!
- But we’re setting up to change the **measure** of integration
- Instead of uniform weighting, we’ll weight by  $q(\mathbf{z})$
- This lets us focus on “important” regions of  $\mathbf{z}$ -space

# From Integral to Expectation

## Step 2: Expectation Form

$$p(\mathbf{x}) = \int q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} d\mathbf{z} = \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

## Pattern Recognition

This has the form:  $\int \underbrace{q(\mathbf{z})}_{\text{probability}} \cdot \underbrace{\frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}}_{\text{function of } \mathbf{z}} d\mathbf{z}$

Which is exactly:  $\mathbb{E}_{q(\mathbf{z})}[\text{function}]$

# The Problem We're Still Facing ⚠

We've made progress, but we're not out of the woods yet...

After Step 2, we have:

$$\log p(\mathbf{x}) = \log \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

Unfortunately, this is **still intractable!** ❌

We've converted the integral to an expectation, but we haven't solved the fundamental optimization problem.

- **Complex computation:** Sample many  $\mathbf{z}$  values, compute ratios, average, then take log
- **Killer issue:** Expression  $\log(\text{average of stuff})$  is not easily differentiable
- **No gradients:** Can't straightforwardly backpropagate through this

**We need a completely different strategy! ⚡**



# The Strategic Insight: If You Can't Optimize It, Bound It

## The breakthrough realization of variational inference...

### The Key Question

*We can't compute or optimize  $\log p(\mathbf{x})$  directly, but what if we could find something that's always  $\leq \log p(\mathbf{x})$ , and when we make this lower bound as large as possible, we're also pushing  $\log p(\mathbf{x})$  up?*

For gradient-based optimization, we need an objective that is: 

- 1 **Computable:** We can evaluate it numerically
- 2 **Differentiable:** We can compute gradients with respect to our parameters
- 3 **Relevant:** Optimizing it actually helps with our real goal

### The Strategy

Instead of trying to maximize the intractable  $\log p(\mathbf{x})$ , we'll find a **lower bound** that satisfies all three criteria.

# Enter Jensen's Inequality: The Perfect Tool ✎

Jensen's inequality is exactly what we need...

- For any concave function  $f$  (like the logarithm):

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$$

- Applied to our problem:

$$\log \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right] \geq \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

# Enter Jensen's Inequality: The Perfect Tool

$$\log \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right] \geq \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

## Why This Transforms Everything

**Left side (intractable):**  $\log(\text{expectation})$  — creates differentiability issues

**Right side (tractable):**  $\text{expectation}(\log)$  — we can handle this!

The right side can be:

- **Computed via sampling:**  $\mathbb{E}[g(\mathbf{z})] \approx \frac{1}{K} \sum_{k=1}^K g(\mathbf{z}_k)$
- **Differentiated easily:** Move gradients inside expectations
- **Optimized with standard methods:** Gradient descent works perfectly

# The Evidence Lower BOund (ELBO) 📁

Our tractable objective function is born...

## The ELBO

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

**We have:**  $\log p(\mathbf{x}) \geq \mathcal{L}$

## Why This is Revolutionary ★

- **Tractable:** We can compute and optimize this bound
- **General:** Works for any choice of  $q(\mathbf{z})$
- **Tight:** Good approximation  $q$  makes bound tight

## The Strategy 🎯

**Instead of:** Maximize intractable  $\log p(\mathbf{x})$

**Do this:** Maximize tractable lower bound  $\mathcal{L}$

**Result:** Indirectly optimize what we actually care about!

# Choosing the Variational Distribution

## The Key Decision

The ELBO depends critically on our choice of  $q(\mathbf{z})$ :

- **Good choice:** Tight bound, efficient learning
- **Poor choice:** Loose bound, slow or failed learning
- **Optimal choice:**  $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x})$  (but this is intractable!)

# Choosing the Variational Distribution

## Different Model Strategies

- **VAEs**: Learn  $q_{\phi}(\mathbf{z}|\mathbf{x})$  with encoder network
- **Mean Field**: Assume independence, learn factorized  $q$
- **Normalizing Flows**: Use invertible transformations
- **Diffusion**: Use fixed, designed forward process

## The Brilliant Diffusion Insight

Instead of learning  $q$ , diffusion models **fix** it by design:

$q(\mathbf{x}_{1:T}|\mathbf{x}_0)$  = predetermined noise schedule This eliminates the approximation problem entirely!

**How this mathematical framework enables real training...**

## Typical Training Loop

- Sample batch of data  $\{\mathbf{x}_i\}$
- Sample latent codes  $\{\mathbf{z}_i\}$  from  $q(\mathbf{z}|\mathbf{x}_i)$
- Compute ELBO estimate:  $\frac{1}{N} \sum_i \log \frac{p(\mathbf{x}_i, \mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x}_i)}$
- Backpropagate and update network parameters
- Repeat until convergence

**Elegant theory meets practical implementation! **



# Homework Problem 1: ELBO Derivation and Analysis

## Your Mission: Complete ELBO Derivation

Walk through the complete ELBO derivation step by step and analyze its properties.

### Part A: Step-by-Step Derivation (6 points)

Starting from the intractable marginal likelihood  $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ :

- 1 Introduce variational distribution  $q(\mathbf{z})$  using the “multiply by 1” trick
- 2 Convert to expectation form
- 3 Apply Jensen’s inequality to create the lower bound
- 4 Show that the gap equals  $D_{KL}(q(\mathbf{z}) \| p(\mathbf{z} | \mathbf{x}))$

**Show all algebraic steps clearly.**





# Homework Problem 1: ELBO Derivation and Analysis

## Part B: ELBO Decomposition (4 points)

- 1 Decompose the ELBO into reconstruction and regularization terms
- 2 Explain the intuitive meaning of each term
- 3 Describe the tension between these terms and why it leads to good representations

# Homework Problem 1 (continued)

## Part C: Jensen's Inequality Deep Dive (4 points)

- 1 Prove Jensen's inequality for the concave logarithm function
- 2 Explain geometrically why the inequality creates a lower bound
- 3 Give a concrete numerical example showing the gap between  $\log(\mathbb{E}[X])$  and  $\mathbb{E}[\log(X)]$

## Part D: Practical Implications (3 points)

- 1 Why is the ELBO easier to optimize than the original likelihood?
- 2 How would you estimate the ELBO in practice using Monte Carlo sampling?
- 3 What happens to optimization if you choose a poor variational distribution  $q(\mathbf{z})$ ?



# Homework Problem 2: KL Divergence and Gaussian Distributions

## Your Mission: Master KL Divergences

Explore KL divergences between Gaussian distributions and their role in variational inference.

### Part A: Gaussian KL Derivation (5 points)

Derive the KL divergence formula for two multivariate Gaussians:

$$D_{KL}(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \parallel \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2))$$

**Start from the definition and show all steps.**

### Part B: Simple Case Analysis (3 points)

For the special case  $p = \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$  and  $q = \mathcal{N}(\mathbf{0}, \mathbf{I})$ :

- 1 Derive the simplified KL formula
- 2 Analyze how the KL changes as  $\|\boldsymbol{\mu}\|$  increases
- 3 Explain what happens when  $\sigma^2 \rightarrow 0$  and when  $\sigma^2 \rightarrow \infty$

### Part C: Asymmetry Exploration (4 points)

Compare  $D_{KL}(p\|q)$  vs.  $D_{KL}(q\|p)$  for specific Gaussian examples:

- 1 Choose two different Gaussian distributions
- 2 Compute both KL divergences numerically
- 3 Explain the difference in terms of mode-seeking vs. mode-covering behavior
- 4 Discuss implications for variational approximation quality

### Part D: Connection to Optimization (3 points)

- 1 How does minimizing  $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$  affect the regularization term in ELBO?
- 2 Why do we typically use KL divergences instead of other distance measures?
- 3 What computational advantages do Gaussian distributions provide?



# Homework Problem 3: ELBO Decomposition and Two-Force Analysis

## Your Mission: Decompose and Understand the ELBO

Take apart the ELBO to reveal its fundamental structure and understand the competing forces.

### Part A: Mathematical Decomposition (5 points)

Starting with the ELBO:  $\mathcal{L} = \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$

- 1 Substitute the chain rule  $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$
- 2 Use logarithm properties to split:  $\log \frac{AB}{C} = \log A + \log B - \log C$
- 3 Use linearity of expectation to separate terms
- 4 Rearrange to show:  $\mathcal{L} = \text{Reconstruction} - \text{Regularization}$

**Show all algebraic steps clearly.**



# Homework Problem 3: ELBO Decomposition and Two-Force Analysis

## Part B: Conceptual Understanding (4 points)

- 1 Explain what the reconstruction term  $\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})]$  measures
- 2 Explain what the regularization term  $D_{KL}(q(\mathbf{z})||p(\mathbf{z}))$  measures
- 3 Describe the tension between these two forces
- 4 Why does this tension lead to meaningful representations?

## Homework Problem 3 (continued)

### Part C: Force Analysis (4 points)

Analyze what happens when forces are unbalanced:







- 1 What happens if reconstruction force dominates (regularization weight  $\rightarrow 0$ )?
- 2 What happens if regularization force dominates (reconstruction weight  $\rightarrow 0$ )?
- 3 Give a concrete example in image generation context
- 4 How would you balance these forces in practice?

### Part D: Connection to Other Models (2 points)

- 1 How does this two-force structure appear in autoencoders?
- 2 How might this relate to the bias-variance tradeoff in machine learning?

# Summary: The Universal Toolkit

## Essential Tools We've Mastered

-  **Marginal distributions:** Why direct computation fails
-  **Expected values:** How to approximate intractable quantities
-  **Bayes' rule:** The foundation of learning from data
-  **KL divergences:** Measuring distribution differences
-  **Jensen's inequality:** Creating tractable bounds
-  **ELBO framework:** The universal solution strategy

**We now have the mathematical superpowers to understand any generative model!** 






# Next Session Preview:

## The ELBO for Diffusion Models

We now have the mathematical superpowers...

Time to apply them to diffusion!

We'll see how to:

- Apply variational inference to sequence generation 
- Derive the three forces of diffusion learning 
- Transform intractable optimization into simple noise prediction 
- Bridge elegant theory with practical training!