Diffusion Models: Theory and Applications

Lecture 4: The ELBO for Diffusion Models - Learning to Reverse Chaos

Shubham Chatterjee

Missouri University of Science and Technology, Department of Computer Science

June 10, 2025

Last Time: We Built the Mathematical Toolkit

We mastered the universal tools for generative modeling...

Now let's apply them to diffusion models!

- ✔ **ELBO framework:** Tractable bounds for intractable likelihoods
- ✔ **Variational inference:** The strategy that makes learning possible
- ✔ **Jensen's inequality:** Creating lower bounds from impossible integrals
- **Today:** Apply these tools to learn the reverse diffusion process!

Today's Mission: From Chaos to Order

How do we learn to reverse destruction?

The Challenge:

- Reverse process is unknown
- Must learn from forward examples
- Need mathematical framework
- Want stable, efficient training

The Solution: ELBO

- Borrowed from VAEs
- Variational inference framework
- Tractable lower bound
- Interpretable loss terms

The ELBO = Our Mathematical Bridge from Theory to Practice

From Single to Sequential Latents: A Crucial Shift

Extending our ELBO framework...

Single Latent Variable (Lecture 3)

$$\mathbf{z} \to \mathbf{x} \quad \text{(single hidden variable)}$$

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

From Single to Sequential Latents: A Crucial Shift

Sequential Latents (Today)

$$\mathbf{x}_T \to \mathbf{x}_{T-1} \to \cdots \to \mathbf{x}_1 \to \mathbf{x}_0$$

- $\mathbf{x}_0$: **Observed data** (what we see)
- $\mathbf{x}_{1:T}$: **Hidden sequence** (latent hierarchy)
- **ELBO:** $\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$

Why This Notation Change?

- Emphasizes **sequential structure** of hidden variables
- Makes **hierarchical relationships** explicit
- Prepares us for **Markovian factorizations** we'll exploit

The Fundamental Challenge: Intractable Likelihood 💀

What we want to maximize:

Our goal

$$\log p_\theta(\mathbf{x}_0)$$

The likelihood of our training data under our generative model

The Fundamental Challenge: Intractable Likelihood

The problem

This requires marginalizing over ALL possible noise trajectories:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$$

- High-dimensional integral over all possible sequences
- Computationally intractable
- Can't optimize directly!

We need a clever workaround...

The Variational Inference Solution

The Big Idea from VAEs:

If you can't compute the exact thing...
Find a tractable lower bound and optimize that instead!

The Evidence Lower Bound (ELBO)

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \triangleq \mathcal{L}$$

What this means: ◎

- $p_\theta(\mathbf{x}_{0:T})$: Our learned generative model
- $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$: The fixed forward process
- Expectation taken over all forward trajectories

Exploiting the Markovian Structure

Now comes the mathematical magic...

Our processes factorize beautifully

Generative model: $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

Forward process: $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$

Why this factorization is powerful

- Markov property: Each step depends only on previous step
- Transforms complex joint distributions into simple products
- Makes mathematical manipulation tractable
- Enables step-by-step analysis

Telescoping the Logarithm

We have the ELBO, but it's not obviously trainable yet...

The Challenge ⚠

Our current ELBO mixes everything together - we can't tell:

- What each part of the model should learn
- How to design loss functions for different components
- Which terms matter most for good generation

Telescoping the Logarithm

Substituting our factorizations into the ELBO:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \tag{1}$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) - \sum_{t=1}^{T} \log q(\mathbf{x}_t|\mathbf{x}_{t-1}) \right] \tag{2}$$

The strategic goal

We need to rearrange these terms to reveal **interpretable learning objectives:**

- **Reconstruction term:** How well we recover $\mathbf{x}_0$ from $\mathbf{x}_1$
- **Prior matching term:** How well our endpoint matches noise
- **Denoising terms:** How well we match optimal reverse steps

Each term will become a separate, trainable loss component!

Homework Problem 1: Complete ELBO Derivation

Your Mission: Derive the Three Forces

Walk through the complete algebraic manipulation to decompose the ELBO into interpretable terms.

Part A: Telescoping and Separation (4 points)

Starting from:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) - \sum_{t=1}^{T} \log q(\mathbf{x}_t|\mathbf{x}_{t-1})\right]$$

1. Identify why $\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$ should be treated specially
2. Separate this term from the summation
3. Rewrite the remaining terms with proper index ranges

Homework Problem 1 (continued)

Part B: Strategic Reindexing (4 points)

You should now have:

$$\mathcal{L} = \mathbb{E}[\log p(\mathbf{x}_T)] + \mathbb{E}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] \tag{3}$$

$$+\,\mathbb{E}\left[\sum_{t=2}^{T} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) - \sum_{t=1}^{T} \log q(\mathbf{x}_t|\mathbf{x}_{t-1})\right] \tag{4}$$

1. Split the forward sum: separate $\log q(\mathbf{x}_T|\mathbf{x}_{T-1})$ from the rest
2. Reindex the remaining forward sum to align with the reverse sum
3. Show that both sums now run from $t = 2$ to T

Part C: The Bayes' Rule Transformation (5 points)

You should now have mismatched terms like:

$$\log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log q(\mathbf{x}_{t-1}|\mathbf{x}_{t-2})$$

1. Explain why this comparison doesn't make sense (what are we comparing?)
2. Use Bayes' rule to write:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

3. Substitute this to create proper comparisons between reverse processes
4. Show how logarithm properties help rearrange terms

Part D: Final KL Divergence Form (4 points)

Transform your result into:

$$\begin{aligned}
\mathcal{L} = \; & \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] && (5) \\
& - D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \| p(\mathbf{x}_T)) && (6) \\
& - \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))] && (7)
\end{aligned}$$

1. Show how differences of logarithms become KL divergences
2. Verify that the expectation subscripts are correct
3. Label each term: reconstruction, prior matching, denoising

Part E: Conceptual Understanding (3 points)

Interpret your final result:

1. Explain in words what each of the three terms measures
2. Why is the reconstruction term treated as a likelihood rather than a KL divergence?
3. Which term requires the most computation during training and why?

The ELBO Decomposition Result ★

After extensive algebraic manipulation...the beautiful result

The ELBO decomposes into three interpretable terms:

$$\mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\mathcal{L}_0\text{:Reconstruction}} \tag{8}$$

$$- \underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))}_{\mathcal{L}_T\text{:Prior matching}} \tag{9}$$

$$- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))\right]}_{\mathcal{L}_{1:T-1}\text{:Denoising matching}} \tag{10}$$

Three interpretable forces shape the learning process! ★

The Three Forces of Diffusion Learning

Force 1: Reconstruction ($\mathcal{L}_0$)

$$\mathcal{L}_0 = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$$

What it does: Ensures we can recover original data from slight noise
Intuition: "Given a photo with tiny grain, restore it perfectly"
Implementation: Often simplified to MSE: $\|\mathbf{x}_0 - \boldsymbol{\mu}_\theta(\mathbf{x}_1, 1)\|^2$

Force 2: Prior Matching ($\mathcal{L}_T$)

$$\mathcal{L}_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))$$

What it does: Ensures forward process endpoint matches prior
Beautiful insight: This is approximately zero by design!
Practical implication: No parameters to optimize - free by construction!

Force 3: The Heart of Diffusion Learning

Denoising Matching ($\mathcal{L}_{1:T-1}$)

$$\mathcal{L}_{t-1} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))\right]$$

What this measures

How well our learned reverse step matches the true optimal reverse step!

The key insight

- **True target:** $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ - optimal denoising given clean target
- **Our prediction:** $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ - learned denoising step
- **Training goal:** Make our network predict what the optimal step would be

This is where the actual learning happens!

The Tractable Reverse Distribution

The remarkable property that makes everything trainable...

The true reverse step is Gaussian!

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\sigma}_t^2 \mathbf{I})$$

The optimal mean (after Gaussian arithmetic)

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \tag{11}$$

$$\tilde{\sigma}_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{12}$$

The Tractable Reverse Distribution

Why this is amazing

- **Weighted interpolation:** Combines clean data and noisy observation
- **Adaptive weighting:** The weights depend on the noise level. As t increases (more corruption), the formula relies more heavily on the clean target $\mathbf{x}_0$ and less on the noisy observation $\mathbf{x}_t$.
- **Fixed variance:** No learning required for variance!
- **Perfect target:** Tells us exactly what optimal denoising looks like

Understanding the Optimal Mean

Let's decode the beautiful interpolation formula:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}}_{\text{Weight for } \mathbf{x}_0} \mathbf{x}_0 + \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\text{Weight for } \mathbf{x}_t} \mathbf{x}_t$$

The adaptive balancing act

- **Early timesteps** (small t): High weight on noisy observation $\mathbf{x}_t$
- **Late timesteps** (large t): High weight on clean target $\mathbf{x}_0$
- **Always sums to 1:** Perfect weighted average!

Intuition: When corruption is severe, trust the clean target more!

The Reparameterization Breakthrough

Instead of predicting the denoised image directly...
What if we predict the noise itself?

Recall the forward jump formula

$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t$
Solving for $\mathbf{x}_0$: $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_t\right)$

Substituting into optimal mean

$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \epsilon_t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right)$

Profound insight: Optimal denoising = noise prediction + simple arithmetic!

Why Noise Prediction is Brilliant

Training becomes: Learn $\epsilon_\theta(\mathbf{x}_t, t)$ to predict noise

Advantages of noise prediction:

- **Scale invariance:** Noise has same scale at all timesteps
- **Simpler target:** Often easier than predicting images
- **Better optimization:** Avoids scaling issues
- **Unified architecture:** Same network for all timesteps

Image prediction problems:

- Different scales at different timesteps
- Complex target structure
- Scaling can hurt gradients
- Requires timestep-specific tuning

Result: Transform intractable reverse learning into manageable noise prediction!

In-Class Exercise: The Noise Prediction Loss

Understanding the Final Loss

If we train $\epsilon_\theta(\mathbf{x}_t, t)$ to predict noise, what does our loss function look like?

Given information:

- Target: True noise ϵ used to create $\mathbf{x}_t$
- Prediction: $\epsilon_\theta(\mathbf{x}_t, t)$
- Training data: $(\mathbf{x}_t, t, \epsilon)$ triples

Questions ?

1. What's the simplest loss function you can write?
2. How does this relate to the ELBO we derived?
3. Why is this much simpler than the full KL divergence terms?

Think: What makes a good noise predictor?

Exercise Solution: The Elegant Simplification ✔

1. The Simple Loss

$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right]$ **Just MSE between true and predicted noise!**

2. Connection to ELBO

- Full ELBO has weighted KL divergences
- Noise prediction emerges from reparameterization
- Many theoretical details get absorbed into simple MSE
- **Remarkable:** Complex math $\rightarrow$ simple practice!

3. Why This Simplification Works

- Gaussian distributions make KL divergences $=$ MSE (up to constants)
- Weighting terms often omitted in practice without hurting performance
- **Insight:** Good noise prediction $\Rightarrow$ good denoising $\Rightarrow$ good generation

The Complete Training Algorithm

From ELBO theory to practical training:

Input: Dataset $\{\mathbf{x}_0^{(i)}\}$, timesteps T, noise schedule $\{\beta_t\}$
Output: Trained noise predictor ϵ_θ

while not converged **do**

1. Sample batch of clean images $\mathbf{x}_0$
2. Sample timesteps $t \sim \text{Uniform}\{1, \ldots, T\}$
3. Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
4. Compute $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
5. Predict $\epsilon_\theta(\mathbf{x}_t, t)$
6. Compute loss $\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2$
7. Backpropagate and update θ

end while

Elegant! Complex ELBO theory $\rightarrow$ simple, practical algorithm

Homework Problem 2: ELBO Term Analysis

Your Mission: Deep Dive into ELBO Components

Analyze how each ELBO term contributes to the learning process and final model performance.

Part A: Reconstruction Term Analysis (4 points)

Given: $\mathcal{L}_0 = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$

1. Explain why this term is treated differently from other denoising terms
2. How would you implement this in practice? (Hint: Think about the noise level at $t = 1$)
3. What happens to sample quality if you remove this term entirely?

Part B: Prior Matching Analysis (3 points)

Given: $\mathcal{L}_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))$

1. Prove why this term is approximately zero for well-designed noise schedules

Part C: Denoising Term Deep Dive (5 points)

Given: $\mathcal{L}_{t-1} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$

1. Explain the intuition: what is this term trying to achieve?
2. Why is conditioning on both $\mathbf{x}_t$ AND $\mathbf{x}_0$ crucial?
3. Derive the connection between this KL divergence and MSE loss on noise prediction

Part D: Practical Implementation (3 points)

1. Which terms actually require neural network computation during training?
2. How does the weighting of different timesteps affect training dynamics?
3. Compare computational cost: full ELBO vs. simplified noise prediction loss

Homework Problem 3: Noise Prediction Equivalence

Your Mission: Prove the Noise Prediction Connection

Show mathematically how the complex ELBO reduces to simple noise prediction.

Part A: Reparameterization Derivation (5 points)

Starting from: $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t$

And: $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right)$

Derive: $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \boldsymbol{\epsilon}) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$

Show all algebraic steps clearly.

Homework Problem 3 (continued)

Part B: Loss Function Connection (4 points)

1. Express the KL divergence $D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$ in terms of the means and variances
2. Show how this reduces to MSE when both distributions are Gaussian with the same variance
3. Explain why optimizing noise prediction $\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2$ is equivalent

Part C: Practical Insights (3 points)

1. Why is noise prediction often easier for neural networks than image prediction?
2. How does the choice of parameterization affect training stability?
3. What are the implications for model architecture design?

The Profound Achievement: From Theory to Practice

What we've accomplished today...

Theoretical Breakthrough:

- Derived tractable ELBO for diffusion
- Three interpretable learning forces
- Connected to optimal reverse process
- Rigorous mathematical foundation

Mathematical Tools:

- Variational inference
- Markov property exploitation
- Strategic term rearrangement
- Bayes' rule application

Practical Impact:

- Simple noise prediction training
- Stable, scalable optimization
- Elegant implementation
- State-of-the-art results

The Transformation:

- Intractable optimization $\rightarrow$ MSE loss
- Complex dependencies $\rightarrow$ Simple algorithm
- Theoretical elegance $\rightarrow$ Practical power

Summary: The ELBO Mastery

What we've learned today

- **ELBO Framework:** Tractable lower bound for intractable likelihood
- **Strategic Rearrangement:** Transform complex sums into interpretable terms
- **Three Forces:** Reconstruction, prior matching, and denoising
- **Noise Prediction:** Elegant reparameterization breakthrough
- **Simple Training:** Complex theory $\rightarrow$ MSE on noise prediction

The elegant conclusion ✓

- Forward process: Systematic destruction ✓
- ELBO derivation: Mathematical framework ✓
- Noise prediction: Practical training ✓
- Next: Advanced techniques and applications!

We now understand the mathematical heart of diffusion models! ♥

Next Session Preview:

Advanced Diffusion Techniques

Key topics we'll explore:

- How do we speed up sampling?
- What about conditional generation?
- How do we improve sample quality?
- What are the latest architectural innovations?

From mathematical foundations to cutting-edge research!

The journey from theory to state-of-the-art continues...