Diffusion Models: Theory and Applications

5-Day Intensive Summer Course

PES University

Department of Computer Science and Engineering

Summer 2025

Instructor
Dr. Shubham Chatterjee
Department of Computer Science
Missouri University of Science and Technology, USA

Contents

List of Tables

1 Course Overview

This intensive course provides a comprehensive introduction to diffusion probabilistic models, from theoretical foundations to practical implementations. Students will learn the mathematics of forward and reverse diffusion processes, implementation techniques, and how to apply these powerful generative models to various applications.

2 Prerequisites

- Strong foundation in machine learning and deep learning concepts
- Proficiency in Python programming and PyTorch
- Basic understanding of probabilistic models and stochastic processes
- Familiarity with computer vision concepts

3 Learning Objectives

By the end of this course, students will be able to:

1. Explain the mathematical foundations of diffusion models
2. Implement basic diffusion processes for generative modeling
3. Train and optimize diffusion models for various applications
4. Apply conditional generation techniques to guide the generation process
5. Develop efficient sampling strategies for diffusion models
6. Create a working implementation for a real-world application

4 Course Structure

The course is structured as a 5-day intensive program:

- **Days 1-4**: Focused on teaching theoretical concepts and hands-on implementation sessions
- **Day 5**: Reserved for mini-project presentations and evaluations

5 Mini-Project

A team-based mini-project will be announced on the first day of the course. This project is designed to assess students' understanding of the course material.

- Students will form teams of two and register their team IDs in the provided Excel sheet
- No separate time will be allocated during teaching hours for project work
- Teams are expected to work on the project independently outside of class hours
- Projects will be evaluated on Day 5

6 Topics Covered

Day 1: Foundations & Forward Diffusion

- Introduction to generative models and their challenges
- Denoising diffusion probabilistic models (DDPMs)
- Mathematical formulation of the forward diffusion process
- Implementation of forward diffusion pipeline

Day 2: Reverse Diffusion & Training Objective

- Reverse diffusion process and noise prediction parameterization
- Derivation of the simplified training objective
- U-Net architecture for noise prediction
- Training loop implementation

Day 3: Training & Sampling

- Practical training aspects and network architectures
- DDPM and DDIM sampling algorithms
- Model training on real datasets
- Implementation of different sampling strategies

Day 4: Conditional Generation & Advanced Topics

- Conditional generation techniques
- Classifier guidance and classifier-free guidance
- Latent diffusion models
- Advanced applications and techniques

Day 5: Project Presentations

- Mini-project presentations by student teams
- Technical discussions and knowledge sharing
- Course wrap-up and future directions

7 Assessment

The course will be evaluated based on a team-based mini-project, which constitutes **100%** of the final grade. Teams of two students will develop, implement, and present a working diffusion model application, demonstrating their technical understanding, problem-solving skills, creativity, and ability to apply course concepts to real-world scenarios.

The assessment will be based on the following four dimensions given below. The detailed rubric is shown in Table 2.

- **Technical Implementation (40%)**: Correctness, efficiency, and completeness of the diffusion model implementation.
- **Oral Presentation (20%)**: Clarity, structure, and professionalism of the project presentation.
- **Quality of Results (20%)**: Quality of generated outputs, experimental results, and evaluation metrics.
- **Understanding of Theoretical Concepts (20%)**: Depth of understanding demonstrated through explanation and responses to questions.

8 Mini-Project Topics

The mini-project is a key component of this course, designed to give students hands-on experience in developing, training, and applying diffusion models to practical problems. Teams of two students will select a project topic from the list below or propose a custom topic (subject to instructor approval). Each project must demonstrate the successful implementation of a diffusion process, training methodology, and sampling procedure. Creativity and originality are encouraged.

Suggested Mini-Project Topics

- **Image Denoising using Diffusion Models**: Train a diffusion model to remove noise from corrupted images such as MNIST or CIFAR-10.
- **Image Inpainting with Diffusion**: Develop a model that can fill missing regions in an image using diffusion-based methods.
- **Class-Conditional Image Generation**: Train a class-conditional diffusion model capable of generating images of specific categories.
- **Text-to-Image Generation with Simple Prompts**: Implement a basic text-to-image system where small vocabulary prompts guide image generation.
- **Super-Resolution via Diffusion**: Build a model that upsamples low-resolution images to higher resolution using diffusion techniques.
- **Latent Diffusion Modeling**: Perform diffusion in a compressed latent space rather than directly in pixel space to improve efficiency.
- **Time Series Data Generation**: Extend diffusion models to generate synthetic one-dimensional signals such as stock prices or ECG patterns.
- **Style Transfer through Diffusion Interpolation**: Use diffusion models to gradually morph one image into another by interpolating in the noise space.
- **Noise Scheduling Experiments**: Experiment with different noise schedules (linear, cosine, learned) and evaluate their impact on model performance.
- **Guided Handwriting Generation**: Build a diffusion model conditioned on characters to generate synthetic handwritten digits or letters.

Custom Projects

Students may propose their own mini-projects, provided they meet the following requirements:

- The project must involve the core concepts of diffusion modeling, training, and sampling.
- The project must be sufficiently challenging and require original implementation work.
- A short proposal (title, description, expected outcomes, and dataset to be used) must be submitted for approval by the instructor within the first two days of the course.

General Guidelines

- All projects must include a working code implementation and a live demonstration during the final presentation.
- Students are encouraged to explore creative applications of diffusion models beyond the examples provided.
- Clear documentation and well-organized code will be considered in the final assessment.

9 Required Reading

"Probabilistic Machine Learning: Advanced Topics" (`https://probml.github.io/pml-book/book2.html`)–Chapters as specified in the detailed schedule

10 Detailed Lecture Sessions

This section provides comprehensive details about what will be covered in each lecture session throughout the four teaching days, based on a direct chronological mapping of ‘lectureN.tex‘ to the Nth session slot.

10.1 Day 1: Foundations & Forward Diffusion Process

Lecture Session 1 (9:00 AM - 10:30 AM; 1.5 hours)

Topic: The Challenge of Generating Reality
Content Details:

- Course overview and what will be explored (e.g., creating images/media, math behind gradual generation, hands-on implementation).
- The fundamental question: How to teach machines to create/dream, specifically generating new, realistic photographs.
- Showcase of Generative AI capabilities: OpenAI Sora for video, Meta Audio Box for audio, Flux for images, and scientific applications like drug discovery.
- Introduction to Generative AI and Diffusion Models as the power behind these breakthroughs.
- The challenge of defining ”realistic” and the difference between copying and creating.
- The universality of the generation challenge beyond photos (speech, artwork, music, stories).
- Statistical Perspective: Data as samples from an underlying probability distribution $p(x)$; the problem of only having samples, not the true distribution.
- The ”Dream Solution”: If $p(x)$ were known, generation would be trivial sampling; reality is we only have samples $x_1, ..., x_n$.
- The astronomical difficulty of generating images due to the vastness of the pixel space (e.g., $(256^3)^{256\times256\times3}$ possibilities).
- The Manifold Hypothesis: Realistic data lies on or near a lower-dimensional manifold within the high-dimensional space.
- Neural Network Approach: Using a simpler latent space z and learning a mapping $D_\theta(z) \to x$.
- Autoencoders as a starting point: Encoder $E_\phi(x) \to z$, Decoder $D_\theta(z) \to x$; problem of ensuring random z yields realistic outputs.
- Overview of Generative Model Families: Paths to generation like GANs and VAEs.

- Challenges with GANs (training instability, mode collapse) and VAEs (blurry samples, posterior collapse).
- Introduction to Diffusion Models: Gradual, iterative refinement by starting with noise and progressively denoising.
- Visualization of the diffusion process: Forward process (adding noise $x_0 \to x_T$) and Reverse process (learning to denoise $x_T \to x_0$).
- Key advantages of Diffusion Models: Divide and Conquer, Strong Inductive Biases (hierarchical refinement), Stable Training (simple regression), High Sample Quality.
- Broad applications of diffusion: ControlNet, LoRA, DreamBooth, 3D/4D generation, weather forecasting.
- Mini-Project Announcement & Team Formation Guidelines (Details in Syllabus).

Lecture Session 2 (10:45 AM - 12:00 PM; 1.25 hours)

Topic: The Forward Diffusion Process: Learning to Destroy Data Systematically
Content Details:

- Recap: Diffusion models offer stable training and high-quality samples compared to GANs/VAEs.
- Today's goal: Understanding the forward process – systematically destroying data (adding noise).
- The core insight: Fix the forward (noising) process and learn only the reverse (denoising) process.
- Analogy: Controlled demolition, where each step is predictable.
- Mathematical Foundation: Markov Chains, where $q(x_t|x_{t-1}, ..., x_0) = q(x_t|x_{t-1})$.
- Forward Process Definition: $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$.
  - β_t: Noise schedule (variance of added noise).
  - $\sqrt{1-\beta_t}$: Signal preservation factor.
- Noise Schedules β_t: How much noise to add at each step (e.g., linear, cosine schedules); gradual increase ensures learnability.
- The Reparameterization Trick:
  - Challenge: Sampling operations are not differentiable.
  - Solution: Transform stochastic sampling to deterministic computation: $z = \mu + \sigma\epsilon$ for $\epsilon \sim \mathcal{N}(0, 1)$.
  - Applied to diffusion step: $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}$, where $\epsilon_{t-1} \sim \mathcal{N}(0, I)$.

  - This transformation preserves statistics, enables gradients, and is computationally efficient.

- Variance Evolution: The design ensures variance stays constant (e.g., at 1.0 if x_0 is normalized and ϵ_t has unit variance) if x_{t-1} has variance 1.

- Recursive Structure and Notation:
  - Expanding x_t in terms of x_0 and multiple noise terms.
  - Introducing $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ (cumulative signal preservation).

- The Forward Jump Formula:
  - Due to Gaussian arithmetic (sum of Gaussians is Gaussian).
  - $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$.
  - Allows direct computation of x_t from x_0 for any t, crucial for efficient training.
  - Signal-to-Noise Ratio (SNR): $\text{SNR}_t = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$, decreases as t increases.

- Training Data Generation Algorithm: Sample clean image x_0, sample $t \sim \text{Uniform}\{1, ..., T\}$, sample $\epsilon \sim \mathcal{N}(0, I)$, compute x_t using forward jump, return (x_t, t, ϵ).

10.2 Day 2: Reverse Diffusion Process & Training Objective

Lecture Session 1 (9:00 AM - 10:30 AM; 1.5 hours)

Topic: Mathematical Foundations of Generative Models

Content Details:

- The Maximum Likelihood approach for training generative models: $\theta^* = \arg\max_\theta \sum \log p_\theta(x_i)$.

- The Hidden Variable Problem: For models $p_\theta(x, z)$, computing $p_\theta(x) = \int p_\theta(x, z)dz$ is often intractable due to high-dimensional integration over latent variables z.

- Essential Mathematical Toolkit Overview: Marginal distributions, expected values, Bayes' rule, KL divergences, Jensen's inequality, Variational Inference.

- Marginal Distributions: $p(x) = \int p(x, z)dz$, why they are intractable (curse of dimensionality, Monte Carlo failure).

- Expected Values: $\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx$, Monte Carlo approximation $\frac{1}{N} \sum f(x_i)$ when sampling is possible.

- Bayes' Rule: $p(z|x) = \frac{p(x|z)p(z)}{p(x)}$; its components (posterior, likelihood, prior, evidence) and the circular problem due to intractable $p(x)$.

- KL Divergence: $D_{KL}(p||q) = \mathbb{E}_{p(x)}[\log \frac{p(x)}{q(x)}]$; properties (non-negative, zero if $p = q$, asymmetric).

- Jensen's Inequality: For a concave function f (like log), $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$; key for creating tractable lower bounds.
- The Variational Inference Strategy:
  - Goal: Bound the intractable $\log p(x)$.
  - Trick: Introduce an arbitrary distribution $q(z)$ to write $p(x) = \int q(z) \frac{p(x,z)}{q(z)} dz = \mathbb{E}_{q(z)}[\frac{p(x,z)}{q(z)}]$.
  - Problem: $\log p(x) = \log \mathbb{E}_{q(z)}[\frac{p(x,z)}{q(z)}]$ is still intractable and hard to differentiate.
  - Solution: Apply Jensen's inequality: $\log \mathbb{E}_{q(z)}[\cdot] \geq \mathbb{E}_{q(z)}[\log \cdot]$.
- The Evidence Lower Bound (ELBO): $\mathcal{L} = \mathbb{E}_{q(z)}[\log \frac{p(x,z)}{q(z)}]$, such that $\log p(x) \geq \mathcal{L}$.
  - Maximizing $\mathcal{L}$ indirectly optimizes $\log p(x)$.
  - The gap $\log p(x) - \mathcal{L} = D_{KL}(q(z)||p(z|x))$.
  - ELBO decomposition into reconstruction and regularization terms (e.g., for VAEs: $\mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)))$.
- Choosing $q(z)$: VAEs learn $q_\phi(z|x)$, Diffusion models fix q by design (the forward process).
- General training loop using ELBO and Monte Carlo estimation.

Lecture Session 2 (10:45 AM - 12:00 PM; 1.25 hours)

Topic: The ELBO for Diffusion Models - Learning to Reverse Chaos

Content Details:

- Adapting ELBO for sequential latents in diffusion models: x_0 observed, $x_{1:T}$ hidden sequence. $\mathcal{L} = \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$.
- Exploiting Markovian Structure: Factorizing $p_\theta(x_{0:T}) = p(x_T) \prod p_\theta(x_{t-1}|x_t)$ and $q(x_{1:T}|x_0) = \prod q(x_t|x_{t-1})$.
- Telescoping the Logarithm and ELBO Decomposition: Algebraic manipulation (involving Bayes' rule for $q(x_{t-1}|x_t, x_0)$) leads to:
  - $\mathcal{L} = \mathcal{L}_0 - \mathcal{L}_T - \sum_{t=2}^{T} \mathcal{L}_{t-1}$.
  - $\mathcal{L}_0 = \mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)]$ (Reconstruction term).
  - $\mathcal{L}_T = D_{KL}(q(x_T|x_0)\|p(x_T))$ (Prior matching term, non-learnable, approx. zero by design).
  - $\mathcal{L}_{t-1} = \mathbb{E}_{q(x_t|x_0)}[D_{KL}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t))]$ (Denoising matching terms).
- The Denoising Matching Term $\mathcal{L}_{t-1}$: The heart of learning.

  - The true reverse posterior $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\sigma}_t^2 I)$ is tractable.
  - $\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$.
  - $\tilde{\sigma}_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ (fixed, not learned).

- Parameterizing the learned reverse step $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \tilde{\sigma}_t^2 I)$.
- The KL divergence $D_{KL}(q||p_\theta)$ simplifies to minimizing the MSE between means: $||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2$.
- The Reparameterization Breakthrough for μ_θ:
  - Express $\tilde{\mu}_t$ in terms of x_t and the noise ϵ_t (using $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t)$): $\tilde{\mu}_t(x_t, \epsilon_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t)$.
  - So, $\mu_\theta(x_t, t)$ is parameterized by having a network $\epsilon_\theta(x_t, t)$ predict the noise ϵ_t.
- The Simplified Loss Objective: The MSE loss on means becomes an MSE loss on noise: $L_{simple} = \mathbb{E}_{t,x_0,\epsilon}[||\epsilon - \epsilon_\theta(x_t, t)||^2]$.
- The Complete Training Algorithm based on the simplified objective.

10.3 Day 3: Training & Sampling Diffusion Models

Lecture Session 1 (9:00 AM - 10:30 AM; 1.5 hours)

Topic: Sampling from Trained Diffusion Models - From Noise to Data

Content Details:

- Recap of training: Learned noise predictor $\epsilon_\theta(x_t, t)$ and its use in reverse process mean $\mu_\theta(x_t, t)$.
- The U-Net Architecture for $\epsilon_\theta(x_t, t)$:
  - Input: Noisy image x_t and timestep t.
  - Output: Predicted noise $\hat{\epsilon}$.
  - Structure: Encoder-decoder with skip connections, suitable for image-to-image tasks like denoising.
- DDPM Sampling Algorithm (Ancestral Sampling):
  - Iteratively sample x_{t-1} from $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \tilde{\sigma}_t^2 I)$, starting from $x_T \sim \mathcal{N}(0, I)$.
  - $\mu_\theta(x_t, t)$ computed using the learned $\epsilon_\theta(x_t, t)$.
  - Stochastic step: $x_{t-1} = \mu_\theta(x_t, t) + \tilde{\sigma}_t z$ where $z \sim \mathcal{N}(0, I)$ (except for $t = 1$ where often $z = 0$).
- DDPM Trade-offs: High quality and diversity, but slow due to many steps.

- DDIM (Denoising Diffusion Implicit Models):
  - Motivation: Faster, deterministic sampling.
  - Core Idea: There are many (non-Markovian) reverse processes $q(x_{t-1}|x_t, x_0)$ that share the same marginals $q(x_t|x_0)$ as DDPM.
  - This leads to a family of solutions for the reverse step, parameterized by a variance term σ_t^2, which can be a free parameter.
  - The Deterministic Case ($\sigma_t = 0$ or $\eta = 0$ in DDIM parameterization):
    1. Estimate the noise: $\hat{\epsilon} = \epsilon_\theta(x_t, t)$.
    2. Predict the "clean image" $\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}$.
    3. Construct the next step $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$. This reconstructs the trajectory.
- DDIM Sampling Algorithm:
  - Uses a subset of timesteps $\{\tau_1, ..., \tau_S\}$ where $S \ll T$.
  - Iteratively applies the deterministic update rule.
- Comparison: DDPM (stochastic, slow, high diversity) vs. DDIM (often deterministic, fast, good quality, potentially less diversity if $\eta = 0$). DDIM with 20-50 steps is a common default.

Lecture Session 2 (10:45 AM - 12:00 PM; 1.25 hours)

Topic: Conditional Generation - From Random to Controllable

Content Details:

- Motivation for Conditional Generation: Moving beyond random samples to controlled outputs ($p(x|y)$ based on a condition y).
- Recap of U-Net Architecture for Diffusion Models and its suitability for conditioning:
  - Detailed U-Net Structure: Encoder (contracting path), Decoder (expanding path), skip connections.
  - Adaptations for Diffusion: Time embedding injection (e.g., sinusoidal), attention mechanisms (self-attention for global context, cross-attention for conditioning).
  - Conditioning Points: Injecting conditional information (e.g., class embeddings, text embeddings via cross-attention) at various points/scales in the U-Net.
- Approach 1: Class-Conditional Diffusion Models
  - Training: Learning $\epsilon_\theta(x_t, y, t)$ by incorporating class label y.
  - Architecture Modifications: Class embedding (e.g., learnable lookup table), fusion with time embedding (e.g., addition or concatenation), injection into U-Net blocks (e.g., modulating feature maps).

  - Training and Sampling algorithms adapted for class conditioning.
  - Pros (simple, fast at inference) and Cons (limited to predefined classes, less flexible control).

- Approach 2: Classifier Guidance
  - Core Idea: Steer a pre-trained unconditional model $\epsilon_\theta(x_t, t)$ during sampling using a separate, pre-trained classifier $p_\phi(y|x_t)$.
  - Mathematical Basis (Bayes' rule for scores): $\nabla_{x_t} \log p(x_t|y) \approx \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p_\phi(y|x_t)$.
  - Modified Noise Prediction: $\tilde{\epsilon}(x_t, y, t) = \epsilon_\theta(x_t, t) - \omega\sqrt{1-\bar{\alpha}_t}\nabla_{x_t} \log p_\phi(y|x_t)$, where ω is the guidance scale.
  - Sampling Algorithm: Incorporates classifier gradient computation at each step.
  - Challenge: Requires a noise-aware classifier trained on noisy images across all timesteps.
  - Guidance Scale ω: Controls the trade-off between conditioning strength and sample quality/diversity.
  - Pros (modular, flexible for multiple types of guidance) and Cons (slower due to extra classifier forward/backward pass, complexity of training robust classifier).

- Approach 3: Classifier-Free Guidance (CFG)
  - Core Insight: Implicitly learn the "classifier gradient" by training a single diffusion model $\epsilon_\theta(x_t, y, t)$ that can also handle an unconditional case (e.g., $y = \emptyset$, a null token).
  - Training: During training, randomly replace y with $\emptyset$ with some probability p_{uncond} (conditioning dropout).
  - CFG Sampling Equation: $\tilde{\epsilon}(x_t, y, t) = \epsilon_\theta(x_t, \emptyset, t) + \omega(\epsilon_\theta(x_t, y, t) - \epsilon_\theta(x_t, \emptyset, t))$. This extrapolates in the direction of the condition.
  - Sampling Algorithm: Requires two forward passes of ϵ_θ per step (one conditional, one unconditional).
  - CFG has become dominant for text-to-image models (e.g., Stable Diffusion).

- Comparison of the three approaches and when to use them.

10.4 Day 4: Conditional Generation & Advanced Topics

Lecture Session 1 (9:00 AM - 10:30 AM; 1.5 hours)

Topic: Score-Based Generative Models - Learning the Geometry of Data
Content Details:

- Analogy for Score Functions: Lost hiker in a mountainous landscape; compass (score function) points uphill (towards higher probability) to peaks (data modes).

- The Score Function Definition: $\mathbf{s}(x) = \nabla_x \log p(x)$.
  - Why $\log p(x)$: Transforms products to sums.
  - ∇_x: Direction of steepest increase.
  - $\mathbf{s}(x)$: Vector field pointing to higher likelihood.
  - Example: Score of 1D Gaussian $s(x) = -(x-\mu)/\sigma^2$.
- Score-based perspective vs. traditional $p(x) = \frac{1}{Z}\tilde{p}(x)$: Score function $\nabla_x \log p(x) = \nabla_x \log \tilde{p}(x)$, makes intractable partition function Z vanish.
- Geometric Intuition: Probability density as a landscape; score vectors point uphill.
- Energy-Based Perspective: $p(x) = \frac{1}{Z}e^{-E(x)}$, so $\mathbf{s}(x) = -\nabla_x E(x)$; score as a force towards lower energy (more natural configurations).
- Examples: Score functions for Multivariate Gaussian and Mixture of Gaussians (weighted combination of individual scores).
- Sampling with Score Functions: Langevin Dynamics
  - Algorithm: $x_{t+1} = x_t + \eta\mathbf{s}(x_t) + \sqrt{2\eta}\epsilon_t$.
  - Combines systematic guidance (drift term $\eta\mathbf{s}(x_t)$) and random exploration (diffusion term $\sqrt{2\eta}\epsilon_t$).
  - Theoretical guarantee: Converges to $p(x)$ as $\eta \to 0, T \to \infty$.
- Learning Challenge: How to learn $\mathbf{s}_\theta(x) \approx \nabla_x \log p(x)$ when $p(x)$ is unknown.
  - Naive loss $\mathcal{L}_{naive} = \mathbb{E}_{p(x)}[||\mathbf{s}_\theta(x) - \nabla_x \log p(x)||^2]$ is intractable.
- Score Matching:
  - Mathematical transformation using integration by parts to eliminate the unknown true score $\nabla_x \log p(x)$.
  - Derivation steps: expand squared norm, use $\nabla_x \log p(x) \cdot p(x) = \nabla_x p(x)$, apply integration by parts component-wise (surface integral vanishes), sum components to get trace of Jacobian.
  - The explicit Score Matching objective: $\mathcal{L}_{SM} = \mathbb{E}_{p(x)}[\text{tr}(\nabla_x \mathbf{s}_\theta(x)) + \frac{1}{2}||\mathbf{s}_\theta(x)||^2]$ (ignoring constants).
  - This objective depends only on the learned score function $\mathbf{s}_\theta(x)$ and data samples.
- Computational Cost of Score Matching: The trace term $\text{tr}(\nabla_x \mathbf{s}_\theta(x)) = \sum_i \frac{\partial s_{\theta,i}}{\partial x_i}$ requires d second-order derivatives (full Jacobian), which is expensive for high-dimensional data like images.

Lecture Session 2 (10:45 AM - 12:00 PM; 1.25 hours)

Topic: Research Discussion: Generative Models for Information Retrieval, NLP, and RAG Systems

Content Details:

- Overview of the research session: Identifying high-impact research opportunities by exploring cutting-edge applications of generative models in IR, NLP, and RAG systems.
- The Convergence Opportunity: Mature generative model technologies (Diffusion, VAEs, GANs, Transformers, LLMs) meeting emerging needs (intelligent IR, contextual document generation, personalized systems, knowledge-grounded generation).
- Research Theme 1: Generative Information Retrieval (e.g., using diffusion for query expansion, VAEs for document synthesis, GANs for retrieval enhancement, neural document generation from knowledge graphs).
- Research Theme 2: Neural Document Generation (using VAEs for structure/style, diffusion for refinement/multimodal content) for applications like personalized education, technical documentation.
- Research Theme 3: Advanced RAG (Retrieval Augmented Generation) Architectures (e.g., Diffusion RAG for iterative queries, VAE-enhanced context, GAN-based source validation; proposal for Probabilistic RAG with Diffusion Models for uncertainty quantification).
- Research Theme 4: Multimodal Knowledge Integration (cross-modal VAEs, multimodal diffusion, GAN-based modality translation) for tasks like visual QA from documents.
- Research Theme 5: Personalized Knowledge Systems (user modeling with VAEs for preferences/expertise, content adaptation with diffusion for personalized explanations/complexity).
- Research Theme 6: Factual Accuracy and Hallucination Control in generative models (uncertainty-aware VAEs, fact-grounded diffusion, adversarial fact checking, retrieval-constrained generation).
- Research Theme 7: Efficient and Scalable Generative Systems (few-step diffusion, quantized VAEs, distilled models, system optimizations like edge deployment).
- Research Theme 8: AI for Indian Communities (opportunities in agriculture, education, healthcare, multilingual systems, cultural preservation, leveraging generative AI).
  - Application 1: AI-Powered Agricultural Assistance (e.g., multimodal advisor).
  - Application 2: Intelligent Teaching Assistant for Indian Schools (e.g., culturally-aware content generation).
  - Application 3: Rural Healthcare AI Assistant (e.g., symptom-to-advice VAE, medical image enhancement with diffusion).

- Research Theme 9: Agentic AI and Multi-Agent Systems (agents that plan, execute, adapt; novel training paradigms like synthetic experience generation with diffusion, VAE-based curriculum learning, constitutional AI for value alignment).
- Discussion of cross-disciplinary research opportunities, student involvement, technology transfer, future trends, and an activity for a collaborative mini-proposal workshop.

Table 1: Course Schedule with Aligned Lab Content

Time	**Day 1: Foundations & Forward**	**Day 2: Reverse & Training**	**Day 3: Training & Sampling**	**Day 4: Conditional & Advanced**	**Day 5: Project Day**
9:00–10:30	Lecture 1: The Challenge of Generating Reality	Lecture 3: Mathematical Foundations of Generative Models	Lecture 5: Sampling from Trained Diffusion Models - From Noise to Data	Lecture 7: Score-Based Generative Models - Learning the Geometry of Data	Mini-Project Presentations (Groups 1-3)
10:30–10:45	**Break**				
10:45–12:00	Lecture 2: The Forward Diffusion Process: Learning to Destroy Data Systematically	Lecture 4: The ELBO for Diffusion Models - Learning to Reverse Chaos	Lecture 6: Conditional Generation - From Random to Controllable	Lecture 8: Research Discussion: Generative Models for Information Retrieval, NLP, and RAG Systems	Mini-Project Presentations (Groups 4-6)
12:00–1:00	**Lunch Break**				
1:00–3:00	**Hands-on Session 1:** Naive Generators & Forward Diffusion Setup *(Lab 1 Parts 1, 2, Start Part 3; Mini-project announcement)*	**Hands-on Session 3:** U-Net for Diffusion & Reparameterization Trick *(Lab 1 Part 4.1; Lab 2 Part 2)*	**Hands-on Session 5:** Diffusion ELBO Insights & Training Loop Setup *(Lab 1 Part 4.2; Lab 4 Parts 3, 4.2)*	**Hands-on Session 7:** Advanced DDIM (η) & Classifier-Free Guidance Intro *(Lab 5 Part 4.2; Lab 6 Part 5 Initial)*	Mini-Project Presentations (Groups 7-9)
3:00–3:15	**Break**				

Time	**Day 1: Foundations & Forward**	**Day 2: Reverse & Training**	**Day 3: Training & Sampling**	**Day 4: Conditional & Advanced**	**Day 5: Project Day**
3:15–5:00	**Hands-on Session 2:** Advanced Noise Schedules & Forward Jumps *(Lab 1 Finish Part 3; Lab 2 Parts 3, 4)*	**Hands-on Session 4:** Math Foundations: KL, Jensen's & VAE ELBO *(Lab 3 Parts 2, 3, 4, 5)*	**Hands-on Session 6:** DDPM Sampling: Stochastic Generation *(Lab 5 Part 3)*	**Hands-on Session 8:** Conditional Generation: CFG & Other Techniques *(Lab 6 Parts 3, 4, 5; Project Work)*	Mini-Project Presentations (Groups 10-12) & Course Wrap-up

Table 2: Detailed Mini-Project Rubric

Criterion	**Excellent (90–100%)**	**Good (75–89%)**	**Needs Improvement (Below 75%)**
Technical Implementation (40%)	Fully functional, efficient, well-documented code; demonstrates mastery of diffusion modeling techniques; implements advanced features beyond minimum requirements.	Mostly functional code with minor issues; basic diffusion model correctly implemented; may lack advanced features or polish.	Incomplete or incorrect implementation; major functional errors; minimal use of course techniques.
Oral Presentation (20%)	Clear, logical, and professional presentation; excellent use of visuals and demonstrations; smooth delivery and strong engagement with audience.	Presentation is generally clear and structured; minor issues in flow, visuals, or delivery; responds adequately to questions.	Disorganized, unclear, or rushed presentation; poor use of visuals; difficulty explaining concepts or answering questions.
Quality of Results (20%)	Generated outputs are high quality, thoroughly evaluated, and insightful; demonstrates careful experimental design and analysis.	Generated outputs are reasonable but may have limited evaluation or analysis; some minor issues in experimental setup.	Outputs are poor quality, inconsistent, or missing; little to no evaluation or meaningful analysis.
Understanding of Theoretical Concepts (20%)	Demonstrates deep understanding of forward/reverse diffusion processes, training techniques, and conditional generation; answers questions with strong theoretical insight.	Demonstrates basic understanding of core concepts; able to explain main ideas but may struggle with deeper questions.	Demonstrates superficial understanding; unable to adequately explain diffusion concepts or answer questions.

Mini-Project Proposal Form

Course: Diffusion Models: Theory and Applications Summer 2025

Team Information:

- **Team Name:** ____________________
- **Team Members:** ____________________

Project Title:

____

Brief Project Description: (2–3 sentences summarizing what the project aims to achieve)

Dataset to be Used: (e.g., MNIST, CIFAR-10, synthetic data, etc.)

____

Main Diffusion Concepts Involved:

- Forward Process: ____________________
- Reverse Process / Sampling: ____________________
- (Optional) Conditioning Method: ____________________

Expected Outputs: (what results or outputs will be demonstrated)

Potential Challenges: (what you foresee as the hardest parts of the project)

Approval Status: (To be completed by Instructor)

Instructor Comments:

Approval: ☐ Approved ☐ Needs Revision