Study Material - Youtube

Document Information

- **Generated:** 2025-08-01 22:52:12
- **Source:** https://youtu.be/kJCgO7rwo3Y
- **Platform:** Youtube
- **Word Count:** 2,392 words
- **Estimated Reading Time:** ~11 minutes
- **Number of Chapters:** 3
- **Transcript Available:** Yes (analyzed from video content)

Table of Contents

Video Overview

This lecture, "Guided Diffusion Models," is part of the "Mathematical Foundations of Generative AI" series. It delves into the mechanisms for controlling the output of Denoising Diffusion Probabilistic Models (DDPMs) to perform conditional generation. The instructor builds upon the previously established connection between DDPMs and score-based models, explaining how to guide the diffusion process to generate data that conforms to specific conditions, such as a class label or a text description. Two primary methods are discussed: **Classifier Guidance** and the more advanced **Classifier-Free Guidance**, which is the state-of-the-art technique used in modern generative models like DALL-E 2 and Stable Diffusion.

Learning Objectives

Upon completing this lecture, students will be able to: - Understand the necessity of conditional generation for practical applications of generative AI. - Formulate the mathematical goal of guided diffusion: sampling from a conditional distribution $p(x_0|y)$. - Explain the concept of **Classifier Guidance** and derive its core equation using Bayes' theorem. - Describe the practical implementation of Classifier Guidance, involving a standard DDPM and a separate, pre-trained classifier. - Identify the limitations of the Classifier Guidance approach. - Understand the principles of **Classifier-Free Guidance** and how it eliminates the need for an external classifier. - Explain the training and inference process for Classifier-Free Guidance, including the use of a null conditioning token.

Prerequisites

To fully grasp the concepts in this lecture, students should have a solid understanding of: - **Denoising Diffusion Probabilistic Models (DDPMs):** The forward (noising) and reverse (denoising) processes. - **Score Functions:** What a score function is ($\nabla \log p(x)$) and its fundamental connection to DDPMs, where the model implicitly learns a scaled version of the score. - **Probability and Statistics:** Concepts like conditional probability, marginal probability, and Bayes' theorem are essential. - **Calculus and Linear Algebra:** Familiarity with gradients (∇) and vector operations. - **Neural Networks:** Basic knowledge of neural network training, backpropagation, and U-Net architecture.

Key Concepts Covered

- Conditional Generation
- Guided Diffusion

- Conditional Score vs. Unconditional Score
- Classifier Guidance
- Classifier-Free Guidance
- Bayes' Rule for Score Decomposition

Guided Diffusion Models for Conditional Generation

The lecture begins by establishing the foundation for guided diffusion, recapping the crucial insight from previous lectures.

> **Recap (00:11):** A Denoising Diffusion Probabilistic Model (DDPM) that is trained to perform regression over the noise added at each step is **implicitly predicting the score function**. Specifically, the true score is equivalent to the negatively scaled noise.

This connection is the cornerstone for understanding how diffusion models can be controlled or "guided."

The Need for Conditional Generation

Standard DDPMs are unconditional generators; they learn to sample from the overall data distribution $p(x_0)$. For instance, a DDPM trained on a dataset of animal images would generate a random animal image. However, most practical and commercially valuable applications require **conditional generation**.

Conditional Generation is the task of generating a sample x_0 that adheres to a specific condition y.

- **Example (01:05):** Instead of generating a random image, we might want to generate an "image of a cat playing chess." Here, the text "a cat playing chess" is the condition y, and the generated image is the sample x_0.

The goal, therefore, shifts from sampling from the marginal distribution $p(x_0)$ to sampling from the **conditional distribution** $p(x_0|y)$.

Data and Goal Formulation

To train a conditional model, our dataset must consist of pairs (x_0, y).

- x_0: The data sample (e.g., an image).
- y: The conditioning variable associated with x_0.

The conditioning variable y can take various forms (02:06): 1. **Class Label:** A discrete value representing a category (e.g., `y = 'cat'`, `y = 'dog'`). This is often represented as a one-hot vector. 2. **Text Embedding:** A continuous vector representation of a descriptive sentence, typically generated by a text encoder like CLIP or a transformer.

The central question addressed in this lecture is: > **How do we modify a DDPM to sample from the conditional distribution $p(x_0|y)$?** (04:53)

Two main approaches are presented: Classifier Guidance and Classifier-Free Guidance.

---

1. Classifier Guidance

Classifier Guidance was one of the first successful methods for controlling diffusion models. It introduces an external model—a classifier—to guide the sampling process toward the desired condition.

Intuitive Foundation

The core idea is to start with the standard reverse diffusion process, which is guided by the unconditional score $\nabla_{x_t} \log p(x_t)$, and "nudge" it at each step. This nudge comes from a classifier that knows how to distinguish between different classes.

Imagine the denoising process at step t. We have a noisy sample x_t. 1. The DDPM (U-Net) tells us how to denoise it based on the general structure of the data (the unconditional score). 2. A separate classifier looks at the same noisy sample x_t and tells us, "To make this look more like a 'cat', you should adjust the pixels in *this* direction." This is the classifier gradient.

By combining these two pieces of information, we can guide the denoising process to produce an image that is not only a valid sample from the data distribution but also satisfies the condition 'cat'.

Mathematical Analysis

The mathematical elegance of this approach lies in its use of **Bayes' theorem** to decompose the conditional score.

Goal: We want to model the conditional score, $\nabla_{x_t} \log p(x_t|y)$.

Derivation (09:14): 1. Start with Bayes' rule for probabilities:

$$p(x_t|y) = \frac{p(y|x_t)p(x_t)}{p(y)}$$

- $p(x_t|y)$: The posterior probability of the noisy sample given the condition (what we want). - $p(y|x_t)$: The likelihood of the condition given the noisy sample. This is what a classifier models. - $p(x_t)$: The marginal probability of the noisy sample. - $p(y)$: The probability of the condition.

2. Take the logarithm of both sides:

$$\log p(x_t|y) = \log p(y|x_t) + \log p(x_t) - \log p(y)$$

3. Take the gradient with respect to x_t. The term $\log p(y)$ is constant with respect to x_t, so its gradient is zero.

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t)$$

This equation is the heart of classifier guidance. It breaks down the complex conditional score into two manageable parts:

Term	Name	Intuitive Meaning	How to Obtain It
$\nabla_{x_t} \log p(x_t\|y)$	**Conditional Score**	The direction to move x_t to make it a better sample of the *conditional* distribution.	This is the final guided score we compute.
$\nabla_{x_t} \log p(x_t)$	**Unconditional Score**	The direction to move x_t to make it a better sample of the *marginal* data distribution.	Predicted by the standard, pre-trained DDPM (U-Net).
$\nabla_{x_t} \log p(y\|x_t)$	**Classifier Gradient**	The direction to move x_t to make it *more likely to be classified* as class y.	Obtained by backpropagating through a pre-trained, differentiable classifier.

Practical Implementation and Architecture

The implementation involves two separate models, as illustrated below.

```
graph TD
    subgraph "Classifier Guidance Inference Step"
        direction LR

        subgraph "Standard DDPM"
            Unet["U-Net (predicts unconditional score)"]
        end

        subgraph "External Classifier"
            Classifier["Pre-trained Classifier<br/>(predicts p(y|xt))"]
        end

        xt["Noisy Sample x_t"] --> Unet
        xt --> Classifier

        Unet -->|Unconditional Score<br>  log p(x_t)| Combine
        Classifier -->|Classifier Gradient<br>  log p(y|x_t)| Combine

        Combine["Combine Scores<br>(Addition)"] --> GuidedScore["Guided Conditional Score<br>  log p(x_t
        GuidedScore --> Denoise["Denoise x_t to get x_{t-1}"]
    end
```

Figure 1: Flowchart for a single inference step in Classifier-Guided Diffusion. A standard DDPM provides the unconditional score, while a separate classifier provides the guidance gradient. These are combined to produce the final conditional score.

Steps: 1. **Train a DDPM:** Train a standard U-Net on the data x_0 to predict the unconditional score $\nabla_{x_t} \log p(x_t)$. 2. **Train a Classifier:** Train a separate classifier network (e.g., a ResNet) to predict the class y from a noisy input x_t. This classifier must be trained on data with the same noise schedule as the DDPM. 3. **Guided Sampling:** During inference, at each step t: - Get the unconditional score from the DDPM. - Get the classifier gradient $\nabla_{x_t} \log p(y|x_t)$ by performing a forward and backward pass through the classifier. - Add these two terms to get the final guided score. - Use this guided score in the reverse diffusion update rule to get x_{t-1}.

Limitations of Classifier Guidance

While effective, this method has significant drawbacks (22:51): - **Dependency on a Separate Classifier:** The generation quality is highly sensitive to the quality of the classifier. A poor classifier provides poor guidance. - **Difficult Classifier Training:** The classifier must be trained on noisy images at *all* possible noise levels, which is a very challenging task and can be computationally expensive. - **Architectural Complexity:** It requires training and maintaining two separate, large neural networks.

---

2. Classifier-Free Guidance

Classifier-Free Guidance is a more recent and powerful technique that overcomes the limitations of its predecessor by integrating the guidance mechanism directly into the diffusion model itself.

Intuitive Foundation

The key insight is to make the DDPM's U-Net model *aware* of the condition y. We can train a single model that can operate in two modes: 1. **Conditional Mode:** When given a condition y, it predicts the conditional

score. 2. **Unconditional Mode:** When given no specific condition (a "null" condition, ϕ), it predicts the unconditional score.

During inference, we can then calculate both scores using this single model and combine them to control the strength of the guidance. This is like having a single artist who can both paint a generic landscape (unconditional) and a specific landscape like "a mountain at sunset" (conditional). We can then blend these two abilities to get the desired result.

Mathematical and Practical Implementation

The training and inference process is modified as follows:

Training (29:06): - A single U-Net model, $\epsilon_\theta(x_t, t, y)$, is now conditioned on y. - During training, for a certain percentage of the training examples (e.g., 10-20%), the true condition y (e.g., class label or text embedding) is replaced with a special **null token** ϕ. - The model's objective is still to predict the added noise. - **Result:** The same network learns to perform two tasks: - When y is provided, it learns to predict the noise for the **conditional** case. This is equivalent to learning the conditional score. - When the null token ϕ is provided, it learns to predict the noise for the **unconditional** case. This is equivalent to learning the unconditional score.

Inference (Guidance): The final score used for the reverse step is an extrapolation between the conditional and unconditional scores.

The predicted noise $\hat{\epsilon}$ is calculated as:

$$\hat{\epsilon} = \epsilon_\theta(x_t, t, \phi) + s \cdot (\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \phi))$$

- $\epsilon_\theta(x_t, t, y)$: The model's prediction for the **conditional** noise. - $\epsilon_\theta(x_t, t, \phi)$: The model's prediction for the **unconditional** noise. - s: The **guidance scale** (analogous to λ before, often denoted as `w` or `s`). This is a hyperparameter. - If $s = 0$, $\hat{\epsilon} = \epsilon_\theta(x_t, t, \phi)$, resulting in unconditional generation. - If $s = 1$, $\hat{\epsilon} = \epsilon_\theta(x_t, t, y)$, resulting in standard conditional generation. - If $s > 1$, the model "pushes" the generation further in the direction of the condition, often leading to higher-quality samples that better match the prompt.

The following diagram illustrates the process:

```
graph TD
    subgraph "Classifier-Free Guidance Inference Step"
        direction LR

        Unet["Single U-Net Model<br> _ (x_t, t, y)"]

        xt["Noisy Sample x_t"]
        t["Timestep t"]
        y["Condition y"]
        phi["Null Condition  "]

        xt --> Unet
        t --> Unet
        y --> Unet
        phi --> Unet

        Unet -- Forward Pass 1 --> Epsilon_cond["Conditional Score<br> _ (x_t, t, y)"]
        Unet -- Forward Pass 2 --> Epsilon_uncond["Unconditional Score<br> _ (x_t, t,  )"]

        Epsilon_cond --> Combine
        Epsilon_uncond --> Combine

        Combine["Extrapolate Scores<br>uncond + s * (cond - uncond)"] --> GuidedNoise["Guided Noise  _ha
```

```
    GuidedNoise --> Denoise["Denoise x_t to get x_{t-1}"]
end
```

Figure 2: Flowchart for Classifier-Free Guidance. A single U-Net model is used twice: once with the condition y and once with a null condition . The outputs are combined to create a guided noise estimate, which is used for the denoising step.

Advantages of Classifier-Free Guidance

- **Simplicity:** It only requires training a single model, simplifying the overall pipeline.
- **No External Classifier:** It completely removes the dependency on a separate, potentially difficult-to-train classifier.
- **Improved Quality:** In practice, it often yields higher-quality and more diverse samples than classifier-guided methods.
- **Flexibility:** The guidance scale s can be adjusted at inference time to trade off between sample fidelity (adherence to the prompt) and diversity.

Key Takeaways from This Video

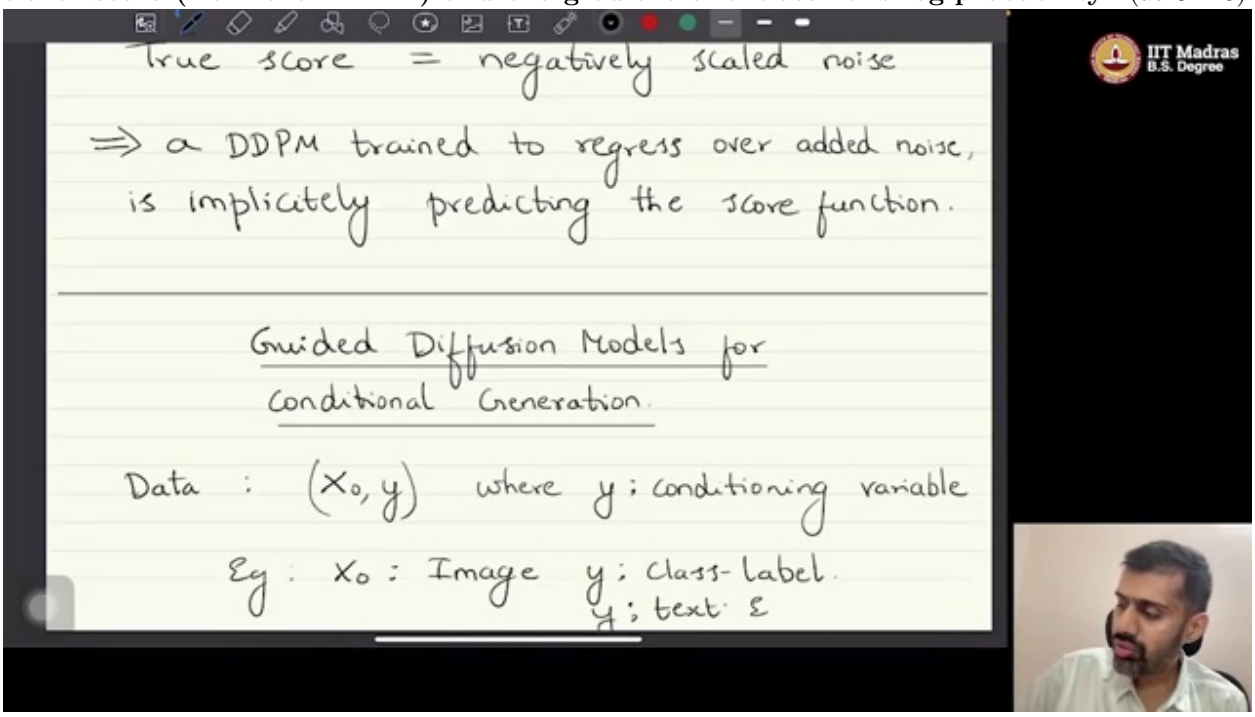
- **Conditional generation** is crucial for making diffusion models useful for real-world tasks. The goal is to sample from $p(x_0|y)$.
- **Classifier Guidance** achieves this by adding a gradient from a pre-trained classifier to the unconditional score from a DDPM. Its main drawback is the need for a separate, robust classifier trained on noisy data.
- **Classifier-Free Guidance** is the modern, state-of-the-art approach. It trains a single U-Net to be both a conditional and an unconditional model by randomly dropping the condition during training.
- During inference, Classifier-Free Guidance combines the conditional and unconditional predictions from the same model to generate a guided sample, offering simplicity, flexibility, and high-quality results.

Self-Assessment for This Video

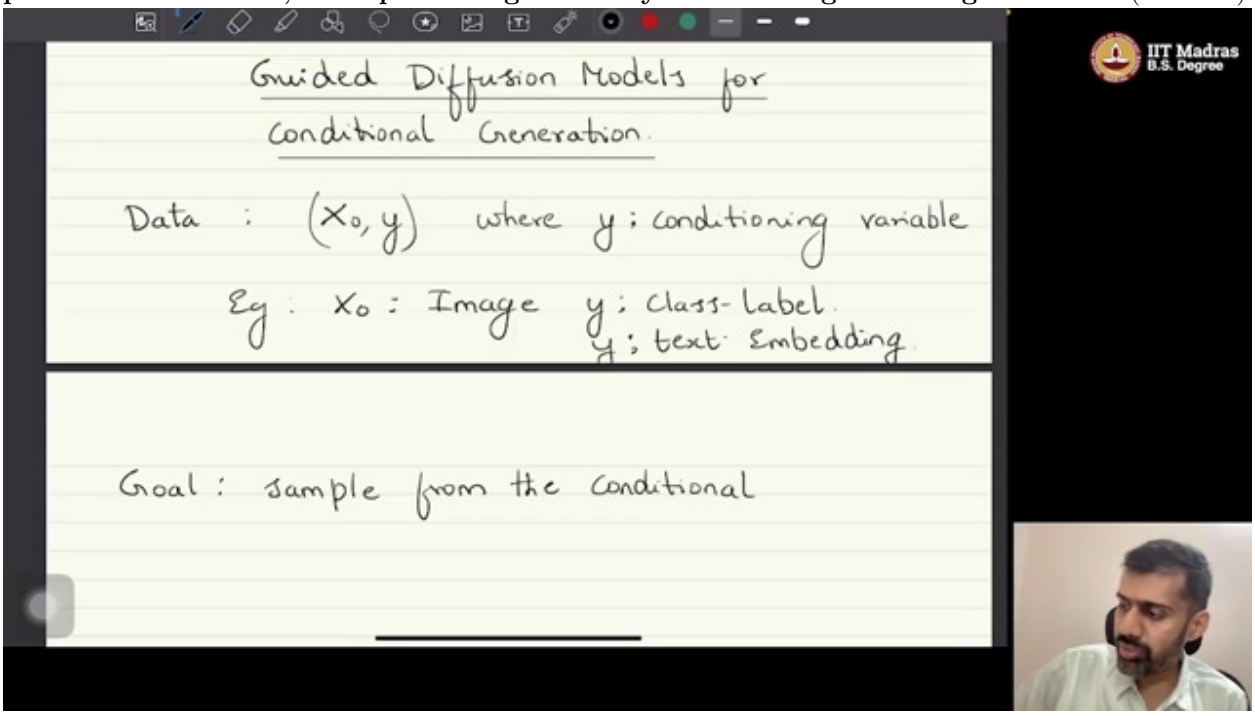
1. **Conceptual Understanding:**
   - What is the fundamental difference between unconditional and conditional generation in the context of diffusion models?
   - Explain in your own words the intuition behind "guiding" the diffusion process. Why is the score function central to this idea?
   - Compare and contrast Classifier Guidance and Classifier-Free Guidance. What is the main advantage of the latter?
2. **Mathematical Formulation:**
   - Write down the Bayes' rule decomposition of the conditional score $\nabla_{x_t} \log p(x_t|y)$. Explain what each term represents.
   - In Classifier-Free Guidance, the final noise prediction is given by $\hat{\epsilon} = \epsilon_\theta(x_t, t, \phi) + s \cdot (\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \phi))$. What happens to the generation process if the guidance scale $s = 0$? What if $s = 1$?
3. **Practical Application:**
   - You are tasked with building a text-to-image diffusion model using Classifier Guidance. What two models would you need to train? What would be the input and output for each during their respective training phases?
   - How does the training process for a Classifier-Free Guidance model differ from that of a standard DDPM? Specifically, what modification is made to the input data?
   - Why is training a classifier for the Classifier Guidance approach considered difficult? What kind of data must it be trained on?

Visual References

The core equation for Classifier Guidance, derived using Bayes' theorem. It shows how the score of the conditional distribution p(xt|y) is decomposed into the unconditional score (from the DDPM) and the gradient of a classifier's log-probability. (at 02:45):

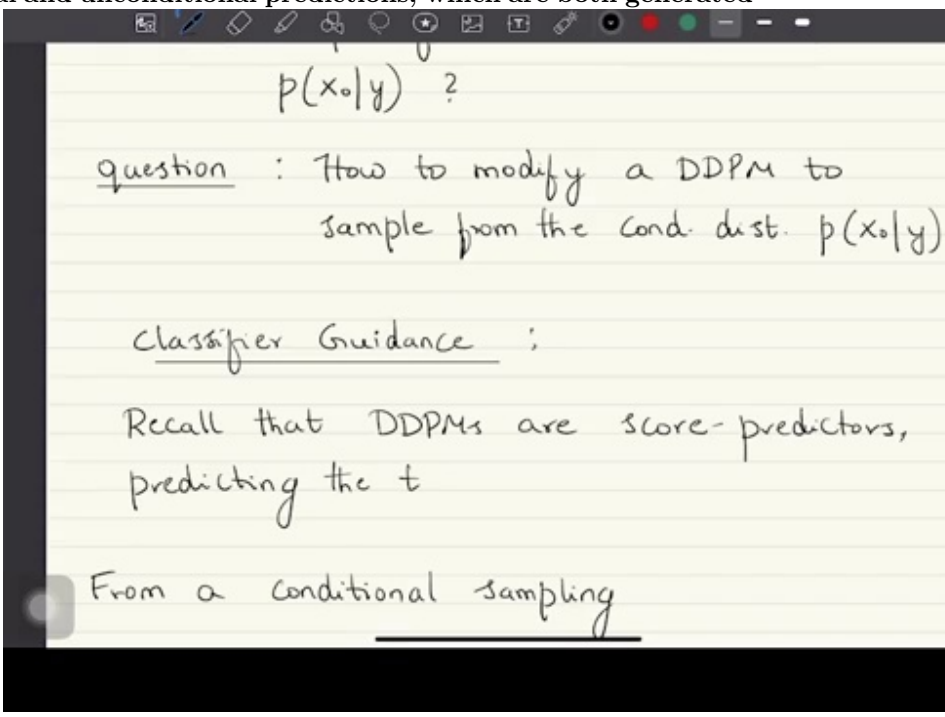


A system diagram illustrating the practical implementation of Classifier Guidance. It visually separates the two required models: the main DDPM (U-Net) and a separate, pre-trained classifier, both processing the noisy data xt to guide the generation. (at 04:30):



The key equation for Classifier-Free Guidance. This slide shows how the guided noise prediction

is an extrapolation from the conditional and unconditional predictions, which are both generated by the same neural network. (at 07:15):



A visual explanation of the training process for Classifier-Free Guidance. It illustrates how the model is trained on both conditioned inputs (e.g., text 'y') and unconditioned inputs by randomly replacing the conditioning token with a null token ( ). (at 09:00):