

# Diffusion Models

How AI Learns to Create Images from Noise

---

Presented by: *Anindya, Tamal and Din*

February 21, 2026

CSE 200: Technical Writing and Presentation  
**Bangladesh University of Engineering and Technology**

# Outline

Introduction

Core Idea

Forward Diffusion Process

Reverse Diffusion Process

Training

Insights

Applications

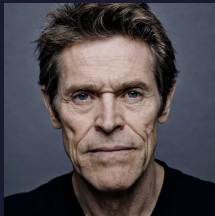
Limitations & Challenges

Conclusion

# What Problem Are We Solving?



Real face

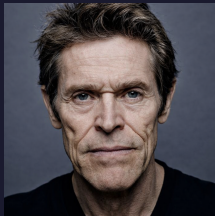


AI-generated face (*Sora*)

# What Problem Are We Solving?



Real face



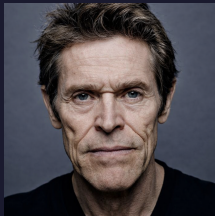
AI-generated face (*Sora*)

- Can a machine learn to **create new, realistic samples** that are *indistinguishable* from real data?

# What Problem Are We Solving?



Real face



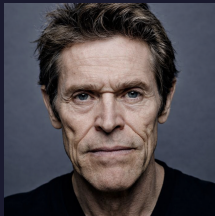
AI-generated face (Sora)

- Can a machine learn to **create new, realistic samples** that are *indistinguishable* from real data?
- In other words can it learn the underlying probability distribution of the data?

# What Problem Are We Solving?



Real face



AI-generated face (Sora)

- Can a machine learn to **create new, realistic samples** that are *indistinguishable* from real data?
- In other words can it learn the underlying probability distribution of the data?

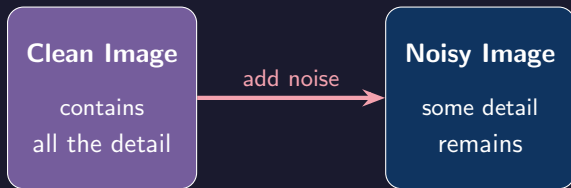
**Diffusion models** provide a *stable, high-quality* solution.

# The Core Idea: Learning by Destroying

**Clean Image**

contains  
all the detail

# The Core Idea: Learning by Destroying

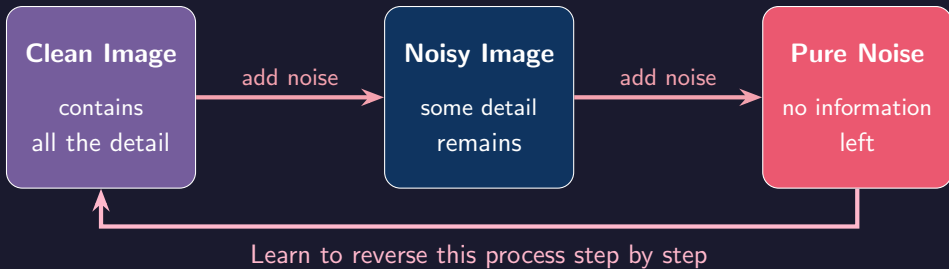




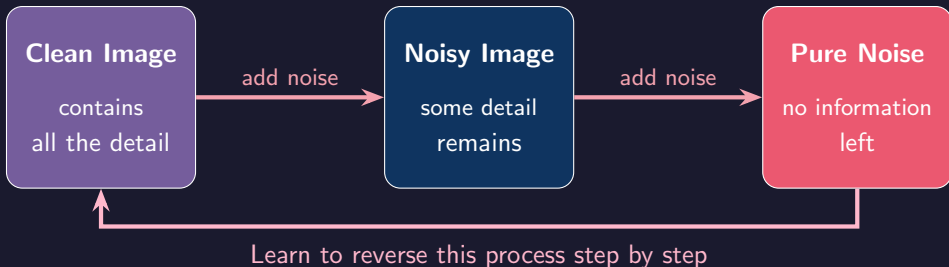
# The Core Idea: Learning by Destroying



# The Core Idea: Learning by Destroying



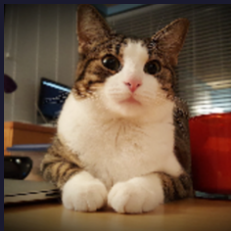
# The Core Idea: Learning by Destroying



## We Essentially Have Two Diffusion Processes

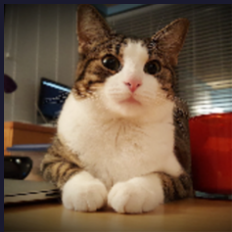
- The **Forward Diffusion Process** that gradually adds noise.
- The **Reverse Diffusion Process** that removes noise step by step.

# The Forward Diffusion Process: Adding Noise



Clean Image

# The Forward Diffusion Process: Adding Noise

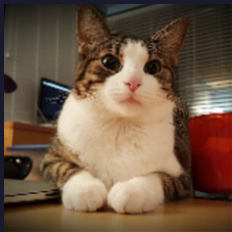


Clean Image



Slightly Noisy

# The Forward Diffusion Process: Adding Noise



Clean Image

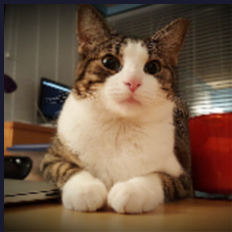


Slightly Noisy



Heavily Noisy

# The Forward Diffusion Process: Adding Noise



Clean Image



Slightly Noisy

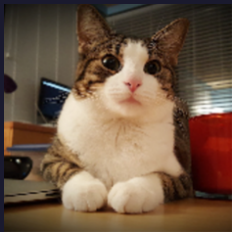


Heavily Noisy



Pure Noise

# The Forward Diffusion Process: Adding Noise



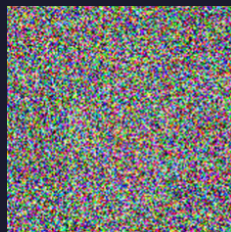
Clean Image



Slightly Noisy



Heavily Noisy



Pure Noise

## What Happens

Take a real image and add a *tiny* bit of random noise. Repeat this hundreds of times until only noise remains.



# The Reverse Diffusion Process: Removing Noise



Pure Noise

# The Reverse Diffusion Process: Removing Noise



Pure Noise



Slightly Less Noisy

# The Reverse Diffusion Process: Removing Noise



Pure Noise



Slightly Less Noisy



Almost Denoised

# The Reverse Diffusion Process: Removing Noise



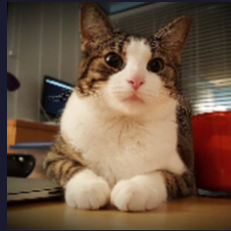
Pure Noise



Slightly Less Noisy



Almost Denoised



Fully Denoised

# The Reverse Diffusion Process: Removing Noise



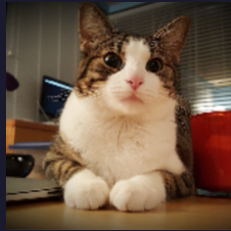
Pure Noise



Slightly Less Noisy



Almost Denoised



Fully Denoised

## How Generation Works

- Start from **pure noise**
- Predict and remove a little noise at each step
- Repeat until a realistic image appears

To produce realistic images, the model must **learn** to **reverse** the noise process.

To produce realistic images, the model must **learn** to **reverse** the noise process.

**But how does the model learn this reverse process?**

## How Do We Train It?

1. Take a real image from the training set



## How Do We Train It?

1. Take a real image from the training set



2. Pick a random step and add the corresponding noise

## How Do We Train It?

1. Take a real image from the training set



2. Pick a random step and add the corresponding noise



3. Ask the neural network: *“what noise was added?”*

## How Do We Train It?

1. Take a real image from the training set



2. Pick a random step and add the corresponding noise



3. Ask the neural network: *“what noise was added?”*



4. Compare the prediction to the **actual noise**, minimize the error

# Key Insight & Why It Works

## Key Insight

- We control the **noise process**
- Added noise is mathematically defined
- So the training signal is precise
- This makes the objective a well-defined *regression problem*

# Key Insight & Why It Works

## Key Insight

- We control the **noise process**
- Added noise is mathematically defined
- So the training signal is precise
- This makes the objective a well-defined *regression problem*

## Why It Works

- The task is simplified into **small, predictable steps**
- Each step only removes a tiny amount of noise
- Repeating simple steps reconstructs complex structure

# Why Diffusion Models Work So Well

## Advantages

- Stable training

# Why Diffusion Models Work So Well

## Advantages

- Stable training
- High Quality Samples

# Why Diffusion Models Work So Well

## Advantages

- **Stable training**
- **High Quality Samples**
- **Good Diversity** (less mode collapse than GANs)



# Why Diffusion Models Work So Well

## Advantages

- **Stable training**
- **High Quality Samples**
- **Good Diversity** (less mode collapse than GANs)
- **Flexible Conditioning** (text-to-image, video-generation, audio synthesis)

# Why Diffusion Models Work So Well

## Advantages

- **Stable training**
- **High Quality Samples**
- **Good Diversity** (less mode collapse than GANs)
- **Flexible Conditioning** (text-to-image, video-generation, audio synthesis)
- **Strong Theoretical Foundation** (probabilistic modeling, score matching)

# Why Diffusion Models Work So Well

## Advantages

- **Stable training**
- **High Quality Samples**
- **Good Diversity** (less mode collapse than GANs)
- **Flexible Conditioning** (text-to-image, video-generation, audio synthesis)
- **Strong Theoretical Foundation** (probabilistic modeling, score matching)

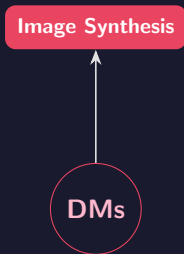
In practice, diffusion models combine reliability with strong output quality.

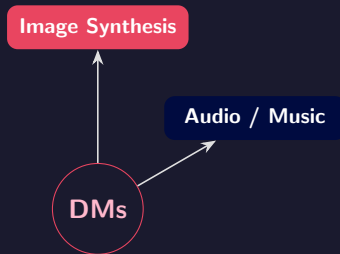
Diffusion models generate realistic data by learning the *underlying probability distribution* through reversing a *gradual, structured noise process*.

## Summary

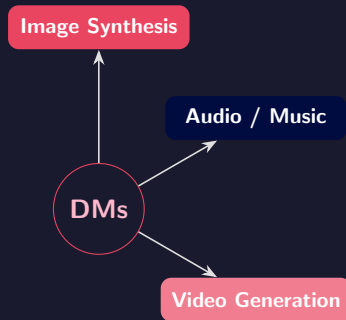
Diffusion models generate realistic data by learning the *underlying probability distribution* through reversing a *gradual, structured noise process*.

**But where is Diffusion Models actually applied?**



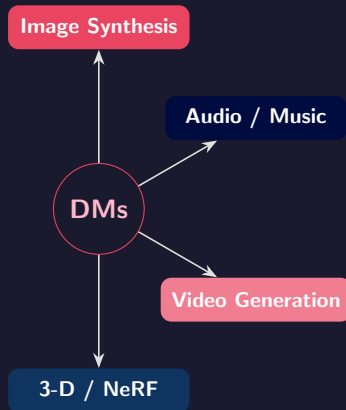


# Applications

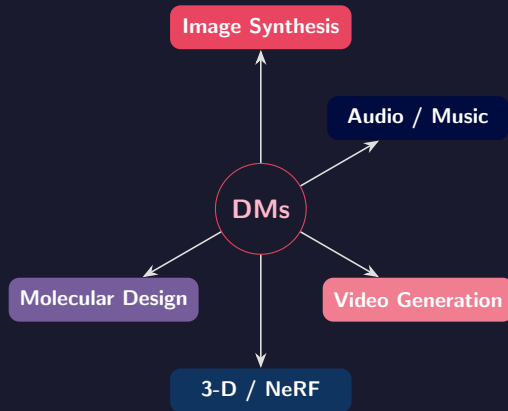




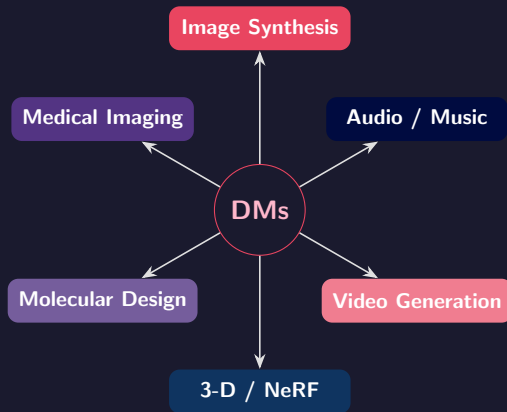
# Applications



# Applications



# Applications



## Some Real-World Tools Powered by Diffusion



Midjourney

## Some Real-World Tools Powered by Diffusion



Midjourney



DALL·E

## Some Real-World Tools Powered by Diffusion



Midjourney



DALL-E



OpenAI Sora

## Current Limitations

- **Slow generation:** requires many denoising steps (hundreds to thousands)

## Current Limitations

- **Slow generation:** requires many denoising steps (hundreds to thousands)
- **High compute cost:** training requires significant GPU resources



## Current Limitations

- **Slow generation:** requires many denoising steps (hundreds to thousands)
- **High compute cost:** training requires significant GPU resources
- **Large model size:** not easy to run on consumer hardware

# Open Challenges

- Can we make generation **faster**? (Active research area)

# Open Challenges

- Can we make generation **faster**? (Active research area)
- How do we ensure generated content is **safe and ethical**?

# Open Challenges

- Can we make generation **faster**? (Active research area)
- How do we ensure generated content is **safe and ethical**?
- Can we extend to **longer videos** and **interactive content**?

The background of the slide features three concentric circles. The innermost circle is a dark, muted purple. The middle circle is a slightly lighter shade of purple. The outermost circle is a dark blue-grey color. The text is centered within the innermost circle.

# Thank You!

Thanks for Listening!

## References

- [1] J. Ho et al. **“Denoising diffusion probabilistic models”**. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020.
- [2] Y. Song and S. Ermon. **“Generative modeling by estimating gradients of the data distribution”**. In: *Advances in neural information processing systems* 32 (2019).
- [3] J. Song et al. **“Denoising diffusion implicit models”**. In: *arXiv preprint arXiv:2010.02502* (2020).
- [4] A. Q. Nichol and P. Dhariwal. **“Improved denoising diffusion probabilistic models”**. In: *International conference on machine learning*. PMLR. 2021, pp. 8162–8171.