# Diffusion Models: Theory and Applications

## Lecture 1: The Challenge of Generating Reality

Shubham Chatterjee

Missouri University of Science and Technology, Department of Computer Science

June 9, 2025

Some slides borrowed and adapted from CS 429d taught by Minhyuk Sung.

# Welcome to the Diffusion Revolution! 🚀

## What we'll explore in this course:

- 🎨 How machines can create stunning images, videos, audio & more!
- 🧠 The mathematical beauty behind "gradual generation"
- 🛠️ Hands-on implementation and training of diffusion models
- 📈 Why diffusion models are a driving force in modern AI

**Today's Journey:** From "Wow!" to "How?"

# The Mind-Bending Question 🤔

## How do we teach machines to dream?

**Imagine standing before a vast collection of photographs...**

Portraits, landscapes, street scenes—each capturing a moment of reality.

**The Challenge:** How can we generate *new*, realistic photographs that have never been taken before?

📷 ➡️ ❓ ➡️ 🪄

# The Art of the Impossible: AI That Creates 🤯

**We're in an era where AI is not just analyzing, but CREATING...**


Figure: OpenAI Sora AI Generating Video from Text

▶️ Click here to watch Sora Example Video (opens in browser)

# Beyond Visuals: AI Creating Audio 🔊💃


Figure: Text-to-Audio (e.g., Meta Audio Box)

▶️ Play Audio Example

# High-Fidelity Images and Scientific Frontiers


Figure: Image Generation (e.g., Flux)

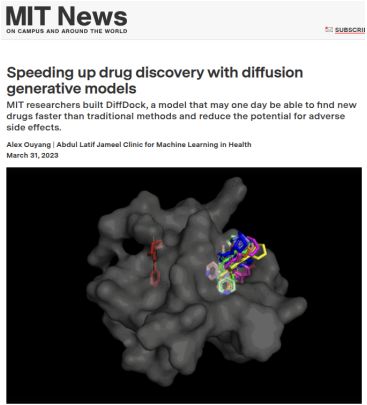



Figure: Scientific Applications [PAPER]

# What Powers This Magic? 🪄

The incredible examples you just saw are powered by **Generative AI**.
Many of the latest breakthroughs leverage **Diffusion Models**.

**This course will unpack the science behind models capable of:**

- 🎥 Video Generation (e.g., Sora, Luma)
- 🎵 Audio and 💃 Motion Synthesis (e.g., Meta Audio Box)
- 🎨 High-Fidelity Image Generation (e.g., Flux, Midjourney, Stable Diffusion)
- 🏡 3D Object and Scene Generation
- 💊 Drug Discovery and 🌧️ Weather Forecasting

**All powered by the math we'll learn together!**

# Quick Exercise: Think Like a Generator

## Partner Discussion (2 minutes)

With your neighbor, brainstorm:

1. What makes a photograph "realistic"?
2. If you had to teach someone to draw faces, how would you break it down?
3. What's the difference between copying and creating?

No wrong answers—just think out loud! 🗣️

# Quick Exercise: Think Like a Generator

## Partner Discussion (2 minutes)

With your neighbor, brainstorm:

1. What makes a photograph "realistic"?
2. If you had to teach someone to draw faces, how would you break it down?
3. What's the difference between copying and creating?

No wrong answers—just think out loud! 🗣️

## Key Insight

Human artists learn by observing patterns, then generating variations. Can machines do the same?

# It's Not Just About Photos! 🌎

**The generation challenge appears everywhere:**

- 🔊 Synthesizing speech
- 🎨 Creating artwork
- 🧬 Modeling molecular structures
- 🎵 Composing music
- ✍️ Writing stories

**The Universal Question**

How do we capture and reproduce the patterns that make data "real" or "plausible"?

**Today:** We start with images, but the principles apply everywhere

# The Key Insight: Think Statistically! 📊

**The Paradigm Shift**

Instead of viewing data as isolated observations, imagine each point is a **sample** from some underlying probability distribution.

**The Data Distribution $p(x)$:**

- High probability = "typical" or "realistic" data
- Low probability = unlikely or unrealistic configurations
- Problem: We only have samples, not the distribution!

🎲 ➡️ Data

📈 Distribution

❓ How to learn?

# The Dream Solution 💭

**If we knew** $p(x)$, generation would be trivial!

## Perfect World Scenario

1. Sample $x \sim p(x)$
2. Get a realistic data point
3. Repeat as needed

## Reality Check ⚠️

We don't know $p(x)$—we only have samples $x_1, x_2, \ldots, x_n$ from it!

**The entire field of generative modeling** is about solving this challenge.

# Why Images Are Astronomically Hard

## A Humbling Calculation

Consider a modest RGB image: $256 \times 256$ pixels
Each pixel: $256^3$ possible colors
Total possible images: $(256^3)^{256\times256\times3} = 2^{24\times196608} \approx 2^{4.7\times10^6}$

**This number exceeds the atoms in the observable universe!**

## The Core Challenge

Among this vast space, only a **tiny fraction** corresponds to realistic photographs.

# The Manifold Hypothesis: Our Saving Grace

**The Key Insight**

Realistic images don't fill the entire high-dimensional space uniformly. Instead, they lie on or near a much **lower-dimensional manifold**.

**Translation:**

- Image space: 196,608 dimensions
- Actual "degrees of freedom": hundreds or thousands
- Think: lighting, pose, identity, expression...

**Manifold = structured subspace**


Not everywhere


Specific paths

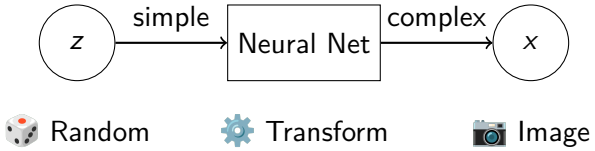
**This suggests a powerful approach...**

# Enter Neural Networks!

## The Big Idea

Instead of modeling the full distribution $p(x)$ directly:

1. Work in a simpler latent space: $z \sim p(z) = \mathcal{N}(0, I)$
2. Learn a mapping: $D_\theta(z) \to x$
3. Use neural networks as flexible function approximators

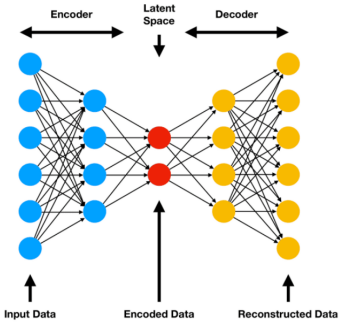



**Generation becomes:** Sample $z$, compute $x = D_\theta(z)$

# The Autoencoder Starting Point ♻️

**Natural Architecture**

- **Encoder:** $E_\phi(x) \to z$
- **Decoder:** $D_\theta(z) \to x$
- **Training:** Minimize reconstruction error




**But wait... 🤔**

For generation: sample $z \sim p(z)$, compute $D_\theta(z)$

**Problem:** How do we ensure random $z$ gives realistic outputs?

# Exercise: The Fundamental Challenge

## Think-Pair-Share (3 minutes)

**The Problem:** An autoencoder trained on faces might encode a smiling face as $z = [0.8, -0.3, 1.2, ...]$

**Questions:**

1. What happens if we sample $z = [0.1, 0.1, 0.1, ...]$ randomly?
2. Why might the decoder produce garbage?
3. What properties should a "good" latent space have?

# Exercise: The Fundamental Challenge

## Think-Pair-Share (3 minutes)

**The Problem:** An autoencoder trained on faces might encode a smiling face as $z = [0.8, -0.3, 1.2, ...]$

**Questions:**

1. What happens if we sample $z = [0.1, 0.1, 0.1, ...]$ randomly?
2. Why might the decoder produce garbage?
3. What properties should a "good" latent space have?

## Key Insight

The latent space organization matters! Random samples might land in "unrealistic" regions.

**To solve the latent space challenge, several paradigms emerged...**

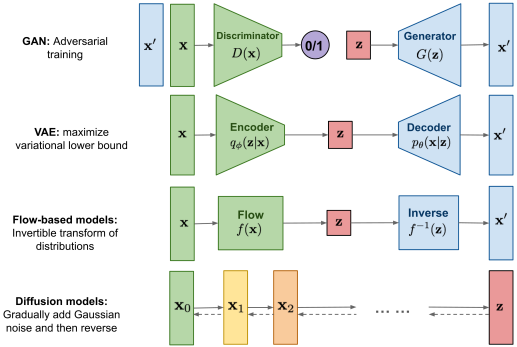



Figure: Overview of Generative Model Families

Image: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

# Why GANs Struggled 😵

## The Adversarial Approach

Train two networks simultaneously:

- **Generator:** Creates fake images
- **Discriminator:** Detects fake images
- **Training:** Minimax game optimization

**Major Challenges:**

- 😵 **Training Instability**
- 🔁 **Mode Collapse**
- ⚖️ **Balance Issues**
- ⚙️ **Hyperparameter Sensitivity**

**Result:** Great when they work, but hard to make them work reliably

# Why VAEs Had Limitations 🤔

## The Variational Approach

Probabilistic framework with encoder-decoder:

- Learn approximate posterior $q_\phi(z|x)$
- Optimize Evidence Lower Bound (ELBO)
- Principled Bayesian foundations

**Key Limitations: (Chatterjee Notes: 3.11, 3.6.1)**

- 🌫️ **Blurry Samples**
- 🤷 **Posterior Collapse**
- 🏗️ **Limited Expressiveness**
- ⚙️ **Reconstruction-Regularization Trade-off**

**Result:** Stable training but often less sharp sample quality

## The Revolutionary Insight

**Instead of one big jump from noise to data...**

**What if we learned to take many small steps?**

**The Diffusion Philosophy:**

1. Start with pure noise
2. Gradually remove noise
3. Each step is small and learnable
4. Final result: realistic data

**Artist Analogy** 🎨

Like a digital restoration artist
working backwards from
a corrupted masterpiece
to reveal the original

**Key Insight:** Divide and conquer—each denoising step is easier than direct generation

# Visualizing the Diffusion Process

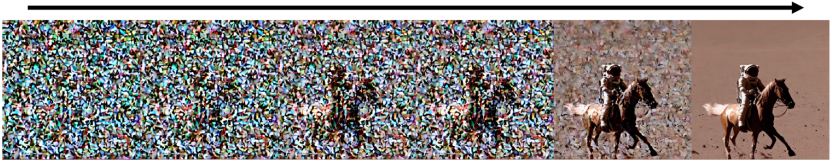


Figure: The generation process of a diffusion model is a denoising process.

We'll unpack the math later!

# Visualizing the Diffusion Process

**Forward Process** ➡️

- Fixed, mathematical
- Gradually add noise
- $x_0 \to x_1 \to \cdots \to x_T$
- Eventually: pure noise

**Reverse Process** ⬅️

- Learned with neural networks
- Gradually remove noise
- $x_T \to x_{T-1} \to \cdots \to x_0$
- Result: realistic data

**The Magic**

We only need to learn the reverse process! The forward process is given to us.

## Four Key Advantages That Changed Everything

1. **Divide and Conquer** 🧩 - Break hard problems into easy ones
2. **Strong Inductive Biases** 🧠 - Natural curriculum learning
3. **Stable Training** 🏡 - No more adversarial nightmares
4. **High Sample Quality** 💎 - State-of-the-art results

**Let's dive deep into each advantage...**

## The Power of Small Steps

### The Challenge with Direct Generation

**Traditional approach:** Learn to map noise $\rightarrow$ realistic image in one giant leap

- Huge function space to explore
- Easy to get stuck in bad local minima
- No guidance on "how to get there"

## The Power of Small Steps

### The Diffusion Solution

**Diffusion approach:** Learn $T$ small denoising steps

- Step 1: Remove just a little noise
- Step 2: Remove a little more noise
- ...
- Step $T$: Final polishing

**Analogy:** Learning to climb Everest vs. learning to climb 1000 small hills ⛰️

## Mathematical Intuition

**Direct Generation:**

- Learn: $f : \mathcal{N}(0, I) \to$ Images
- Massive function approximation
- No intermediate supervision
- Hard optimization landscape

**Example Failure:**

- Random noise $\to$ ??? $\to$ Cat
- No guidance on intermediate steps

**Diffusion Generation:**

- Learn: $f_t : x_t \to x_{t-1}$ for each $t$
- Each step has clear target
- Abundant supervision at every level
- Smoother optimization

**Example Success:**

- Noisy cat $\to$ less noisy cat $\to$ clean cat
- Clear learning signal at each step

**Each denoising step is a well-posed regression problem!**

# Advantage 2: Strong Inductive Biases

## What Are Inductive Biases?

Assumptions built into the learning algorithm that guide it toward good solutions

- **Example:** CNNs assume spatial locality
- **Example:** RNNs assume sequential structure
- **Diffusion:** Assumes hierarchical refinement

## The Natural Curriculum

Diffusion models automatically learn in a sensible order:

- **Early steps ($t$ large):** Remove lots of noise $\rightarrow$ learn global structure
- **Middle steps:** Refine shapes, positioning, major features
- **Late steps ($t$ small):** Add fine details, textures, polish

**Like an artist:** Rough sketch $\rightarrow$ shapes $\rightarrow$ details $\rightarrow$ finishing touches

# Why This Order Matters 🎓

**Early Denoising Steps:**

- Heavy noise → moderate noise
- Learn: "Is this a face? A car? A landscape?"
- Focus on global semantics
- Coarse spatial arrangements

**Benefits:**

- Must learn meaningful representations
- Can't rely on memorizing details
- Forces understanding of structure

**Late Denoising Steps:**

- Light noise → clean image
- Learn: "What texture? Which details?"
- Focus on local refinements
- High-frequency information

**Benefits:**

- Build on solid foundation
- Add details that make sense
- Avoid contradictory information

This curriculum emerges automatically from the noise schedule—no manual design needed! 🪄

# Advantage 3: Stable Training 🏠

## Remember GAN Training? 😵

- Two networks competing in a minimax game
- Generator tries to fool discriminator
- Discriminator tries to catch generator
- Result: Unstable dynamics, mode collapse, vanishing gradients

## Diffusion Training is Different 😌

- **Single network** learning a regression task
- **Clear target:** Predict the noise that was added
- **Abundant data:** Every training image at every noise level
- **Well-posed problem:** Always a correct answer

# Why Stable Training Matters ⚖️

**No Adversarial Dynamics:**

- No careful balancing of two networks
- No discriminator overpowering generator
- No generator mode collapse
- Standard gradient descent works

**Predictable Training:**

- Loss goes down steadily
- Easy to monitor progress
- Hyperparameters less sensitive

**Reproducible Results:**

- Same hyperparameters work across datasets
- Less random trial-and-error
- Easier to scale up
- Better for research and production

**Simple Loss Function:**

- $\mathcal{L} = \mathbb{E}[||\epsilon - \epsilon_\theta(x_t, t)||^2]$
- Just mean squared error!
- No complex adversarial objectives

# Advantage 4: High Sample Quality 💎

## Iterative Refinement Power

Unlike single-step generation, diffusion models can:

- **Self-correct:** Each step can fix mistakes from previous steps
- **Progressive improvement:** Quality increases throughout sampling
- **Flexible quality:** More steps = higher quality (when needed)

## Quality Comparison (Typical Results)

- **VAEs:** Stable but often blurry, limited detail
- **GANs:** Sharp when working, but mode collapse issues
- **Diffusion:** Sharp, diverse, AND stable training

**Diffusion models often achieve the best FID scores on standard benchmarks**

# The Power of Iterative Generation 🔁

**Single-Step Generation:**

- Network must be perfect immediately
- No opportunity to fix mistakes
- All-or-nothing approach
- Limited by network capacity

**Pressure:** Get everything right in one shot!

**Iterative Generation:**

- Each step only needs small improvement
- Can correct previous mistakes
- Gradual refinement approach
- Compound improvements over time

**Freedom:** Many chances to get it right!

🎯

**Real-World Analogy**

**Single-step:** Writing a perfect essay in one draft
**Iterative:** Brainstorm → outline → draft → revise → polish

**Which approach typically produces better results?** 🤔

# Bringing It All Together 🧩

## The Virtuous Cycle

1. **Divide & Conquer** makes each step learnable
2. **Natural Curriculum** ensures learning happens in the right order
3. **Stable Training** means we can actually optimize this effectively
4. **Iterative Refinement** produces high-quality results

## The Result

Diffusion models solved the generative modeling trilemma:

- ✅ High sample quality
- ✅ Training stability
- ✅ Mode coverage/diversity

**This is why diffusion models are powering the current AI revolution!** 🚀

Zhang et al., Adding Conditional Control to Text-to-Image Diffusion Models, ICCV 2023.

## Conditional Generation

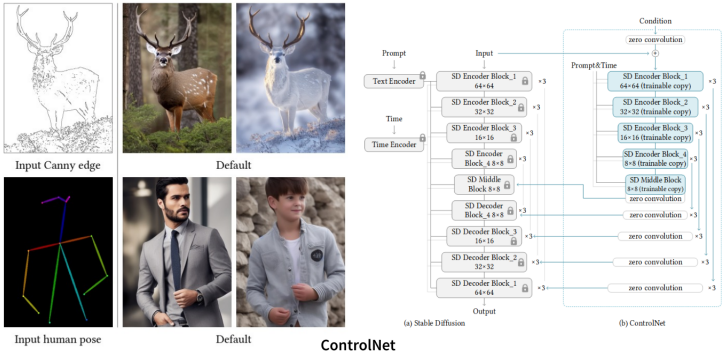



Figure: Controlling image structure and pose with ControlNet [ControlNet Paper]

https://www.shruggingface.com/blog/self-portraits-with-stable-diffusion-and-lora

## Stylization

LoRA

Figure: Parameter-efficient fine-tuning of diffusion models using LoRA [LoRA Paper]

Ruiz et al., DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, CVPR 2023.

## Personalization

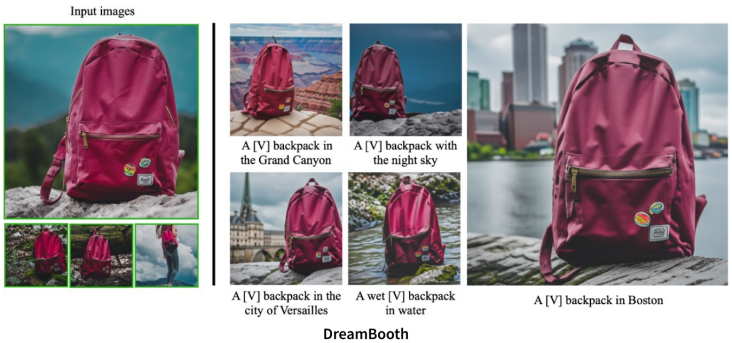



Figure: Personalized image generation with DreamBooth [DreamBooth Paper]

Tang et al., LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation, ECCV 2024.

## 3D Generation


Figure: 3D shape and object synthesis using diffusion-based models [Paper]

# Diffusion's Reach: A Quick Tour of Applications

Zheng et al., Dream-in-4D: A Unified Approach for Text- and Image-guided 4D Scene Generation, CVPR 2024.

## 4D Generation

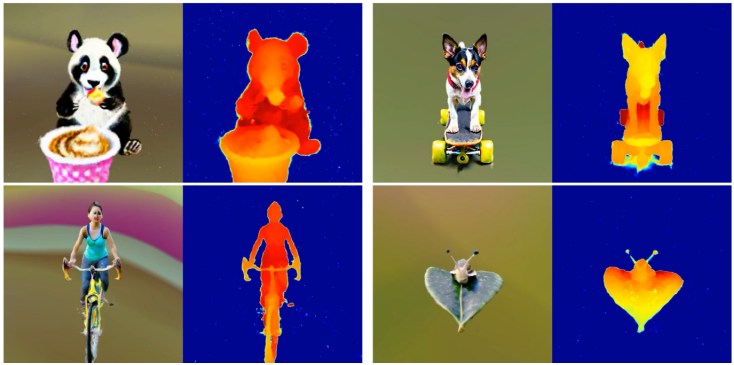


Figure: 4D generation: Modeling dynamic scenes over time with spatio-temporal diffusion [Paper]

# Diffusion's Reach: A Quick Tour of Applications

https://research.nvidia.com/publication/2024-08_kilometer-scale-convection-allowing-model-emulation-using-generative-diffusion

## Weather Forecasting

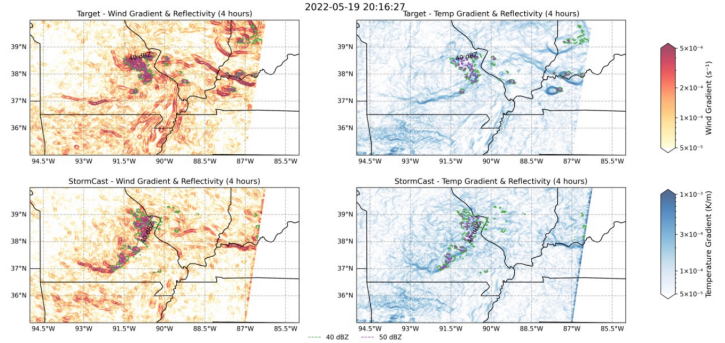



Figure: Diffusion models for high-resolution weather prediction [Paper]

# Final Exercise: Connecting the Dots

## Group Discussion (4 minutes)

Form groups of 3-4. Discuss and prepare to share:

**Scenario:** You're explaining diffusion models to a friend who knows about photo editing but nothing about AI.

1. How would you explain the "add noise then remove noise" concept?
2. What real-world analogy would you use?
3. Why is learning to denoise easier than learning to generate directly?
4. What questions would your friend likely ask?

Think creatively—the best explanations are often the simplest!

# Ready for the Math? 📈

**Now that we understand WHY diffusion works...**

## Coming Up Next

**The mathematical machinery that makes it all possible:**

- How exactly do we add noise? (Forward process)
- What does the noise schedule look like?
- How can we "jump" to any timestep directly?
- What makes this mathematically tractable?

**Spoiler:** The math is more elegant than you might expect! 🪄

# Course Roadmap & Mini-Project Kick-off 👷

## Our Journey Ahead (Next Lectures):

- **Lecture 2:** The Mathematics of Noise (Forward Process details)
- **Lecture 3:** Learning to Reverse (ELBO, Loss Functions)
- **Lecture 4 & Beyond:** Training, Sampling, Conditional Models, Advanced Topics...

## Mini-Project Quick Info

- Team-based (2 students)
- Apply course concepts!
- Details in Syllabus
- Team formation in first lab
- Presentations on Day 5

See syllabus for full mini-project details.

# Today's Key Takeaways ✅

1. **Generative modeling is about learning distributions** from samples
2. **Images live on low-dimensional manifolds**, not in full pixel space
3. **Direct generation is hard**, but gradual refinement is manageable
4. **Diffusion models divide and conquer** the generation problem
5. **Learning to denoise is easier** than learning to generate directly

**The stage is set for the mathematical deep dive!**

🚀 Ready to see how the magic actually works? 🪄

# Questions & Next Steps 🤔

**Any questions about today's concepts?**

## For Next Time

- Review: What is a probability distribution?
- Think about: How might you add noise to an image mathematically?
- Bring: Curiosity about Gaussian distributions! 💹

**Next lecture:** We'll see how controlled chaos becomes the foundation for incredible generation capabilities!

🙈 ➡️ 🌊 ➡️ 🎨

# Next Session Preview:

# The Forward Diffusion Process

**We'll see how diffusion models:**

- Systematically destroy data with mathematical precision
- Design the perfect noise injection schedule
- Use Markov chains for elegant tractability
- Enable instant "forward jumps" through Gaussian magic

**From perfect images to pure noise...**
**in a completely reversible way!**