Diffusion Models

How AI Learns to Create Images from Noise

Presented by: *Anindya, Tamal and Din*

February 19, 2026

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

Roadmap

- Motivation
- Forward Process
- Reverse Process
- Training & Score Matching
- Applications
- Conclusion

• 10 min talk

Speaker 1 (Anindya Biswas)	3 min
Speaker 2 (Tasnim Tamal)	3 min
Speaker 3 (Din Mohammad)	3 min
Buffer / Transition	1 min

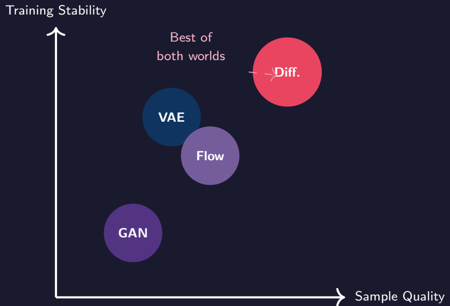
Why Diffusion Models?

In One Line

A diffusion model learns to **turn random noise into meaningful data** (like realistic images).

- **GANs**: high quality, but training is often unstable
- **VAEs**: stable and fast, but outputs can look blurry
- **Flows**: mathematically clean, but architecture is restricted
- **Diffusion**: stable training + very strong output quality

Used in tools like DALL·E 3, Stable Diffusion, and Sora

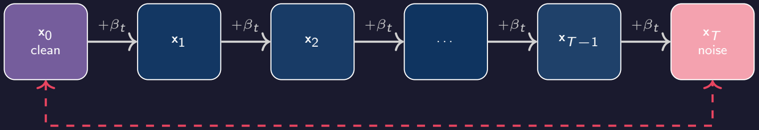


The Forward (Noising) Process

Key Idea

Gradually corrupt data $\mathbf{x}_0$ into pure Gaussian noise $\mathbf{x}_T$ through a fixed Markov chain.

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\Big(\mathbf{x}_t;\ \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t \mathbf{I}\Big)$$



$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Plainly: start from a real sample and add a little noise repeatedly until almost all information is gone.

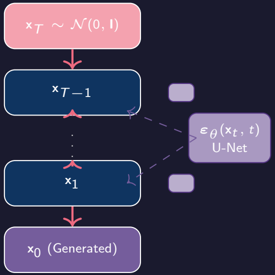
The Reverse (Denoising) Process

Goal: Learn $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to reverse the noise.

A **neural network** ε_θ predicts the noise present at each step:

$$\mathbf{x}_{t-1} \approx \mathbf{x}_t - \text{predicted noise}$$

- Backbone: usually a **U-Net**
- Input includes both noisy sample and timestep t
- Repeat many times to gradually recover clean data



Training Objective

Training Idea (Simple Version)

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\varepsilon}\left[\left\|\,\varepsilon \;-\; \varepsilon_\theta\!\left(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon,\; t\right)\right\|^2\right]$$

Algorithm — Training

1. Sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
2. Sample timestep $t \sim \mathcal{U}\{1, T\}$
3. Sample $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$
4. Compute noisy sample $\mathbf{x}_t$
5. **Minimise** $\|\varepsilon - \varepsilon_\theta(\mathbf{x}_t, t)\|^2$

What this loss means

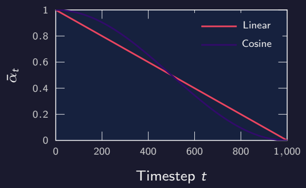
- We add known noise to a clean sample
- The model guesses that exact noise
- If the guess is close, the model is learning correctly
- After training, we remove noise step by step

Key insight: learning to denoise is enough for generation.

Noise Schedule & Sampling Speed

Noise Schedules

- *Linear*: increase noise at a constant rate
- *Cosine*: keeps more structure early on
- *Learned*: schedule is optimized by data

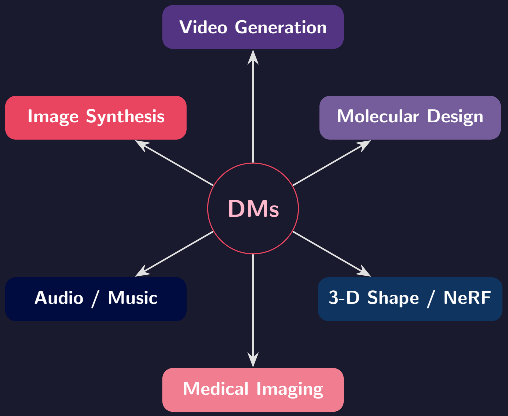


Accelerated Sampling

- DDPM: $T = 1000$ steps (slow)
- **DDIM** (Song et al., 2020): deterministic, $\sim$50 steps, same quality
- **DPM-Solver**: $\sim$20 steps
- **Consistency Models**: 1–2 steps!

Speed-quality trade-off is an active research frontier.

Applications & Variants



Conditional Generation

- **Classifier guidance**: steer outputs toward a target label
- **Classifier-free**: no separate classifier required
- **Text conditioning**: prompt-driven image generation

Notable: Stable Diffusion (latent space), DALL·E 3 (GPT captions), Sora (video)

Takeaways & Open Challenges

Key Takeaways

- Diffusion = iterative **denoising**
- Training objective is **simple MSE**
- Grounded in **score matching** theory
- Sets **state-of-the-art** across modalities

Open Challenges

- Slow sampling (active research)
- High compute cost
- Evaluation metrics (FID, CLIP-S)
- Alignment & safety of generated content

Core Formula to Remember

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\varepsilon}\left[\left\|\varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon,\ t\right)\right\|^2\right]$$

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.