Lecture 7:

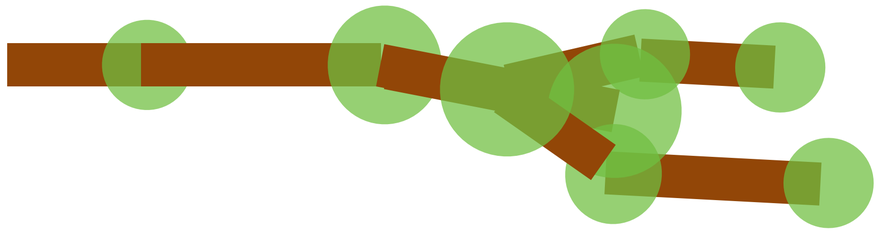
Discussion of Designing for AutoDiff + Intro to Generative AI

Visual Computing Systems
Stanford CS348K, Spring 2026

Discussion of Design for AutoDiff

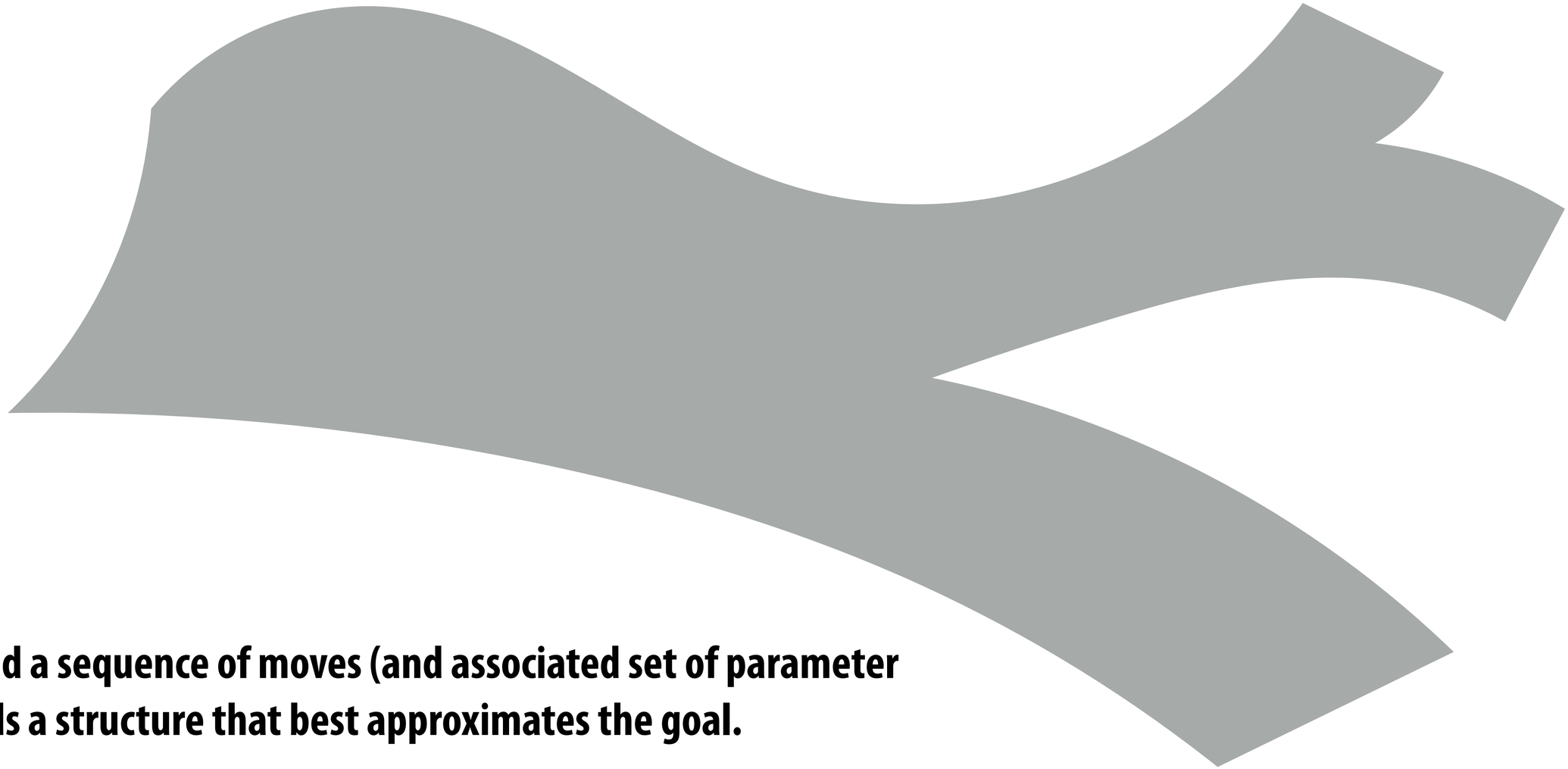
Procedural modeling as a sequence of moves

- Grammar's rules describe allowable moves
- A sequence of moves turns a starting geometry object into another one
- Example: tree builder
  - Geometric objects = { branch(length, angle), leaf(radius) }
  - Rules = { modifyParams of existing, addLeaf, removeLeaf, splitBranch, removeBranch, addAnywhere}
  - Constraints = { branch can't rotate too much, leaf r radius must with within a certain size, etc… }



Optimization to fit a target

Example, make me a structure that looks like “this image”, fills this region, etc.



So we need to find a sequence of moves (and associated set of parameter values) that yields a structure that best approximates the goal.

Evaluating a technical idea

- What is the claimed idea?

- What is the proposed solution?
  - What are the best known alternatives to the proproposed solution?

- In the evaluation:
  - What are questions a reasonable person might ask about the claimed idea to determine if it is in fact a good one?
  - What are the experiments to perform to answer those questions?

Evaluating a technical idea

- In the evaluation:
  - What are questions a reasonable person might ask about the claimed idea to determine if it is in fact a good one?
  - What are the experiments to perform to answer those questions?

- Possible questions:
  - Do I believe the results are "good" … meet stated goals, or meet the reader's assessment of what the goals should be
  - If the results are "good", is the stated idea responsible for the good results? (Or did they arise because of something else)
  - Could the authors have obtained the good results (or perhaps even better results) using a well-known alternative method?
  - What are the "limits" of this idea: Under what other conditions should I expect to also receive good results? (Or conversely, under what other conditions would I expect to receive bad results?)

A significant fraction of your project grade will be:

1. How thoughtfully (i.e. specifically) did the students define their problem to solve

2. How thoughtfully did the students evaluate their idea or technical implementation

(This is different from what many of you are used to.)


bitly
Scan me

Micro-intro to diffusion-based generation (Prep for next time)

Let's look at some examples…

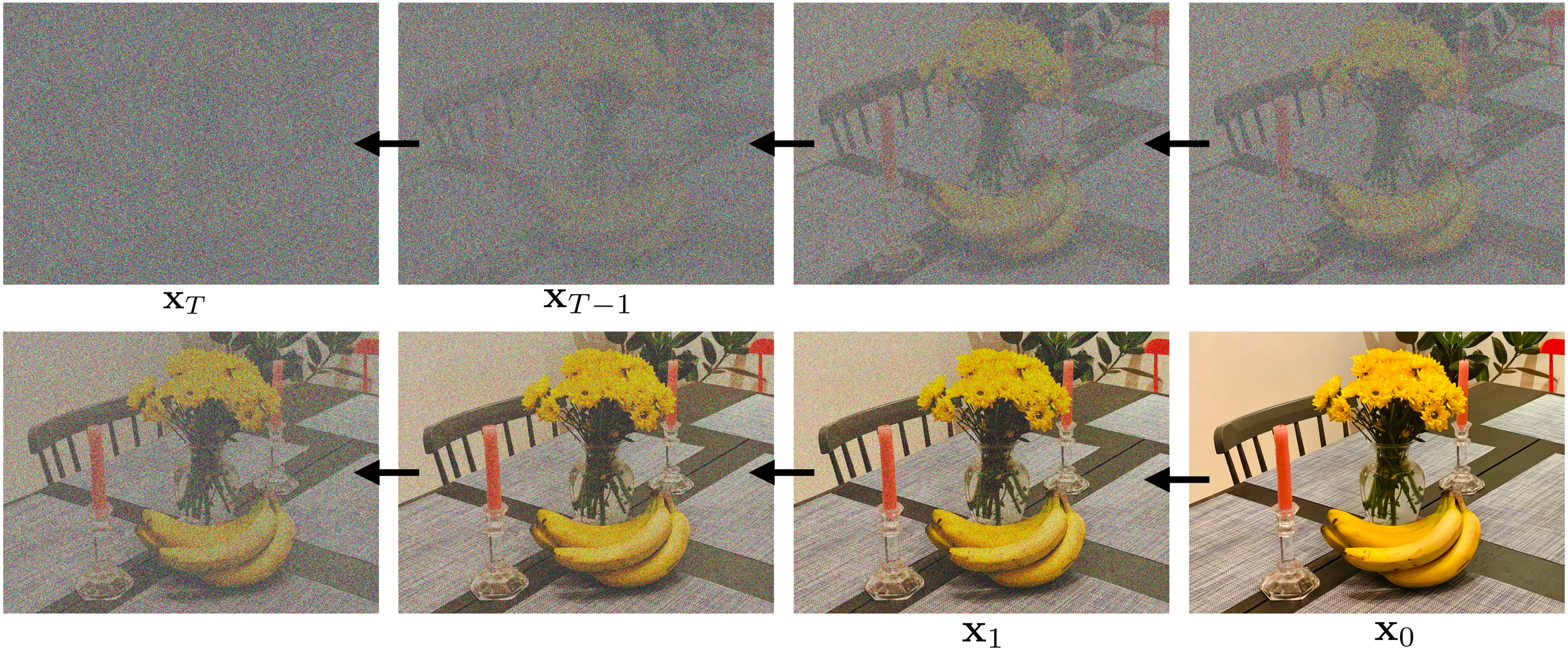
Suppose you are given a dataset of images x_i

- You'll like to draw a sample according to the underlying data distribution p(x)

Diffusion-based image synthesis

Iterative Markov-chain Monte-carlo (MCMC) process to generate a sample x (an image) from distribution p(x) of observed images

Forward diffusion: iteratively add noise $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$



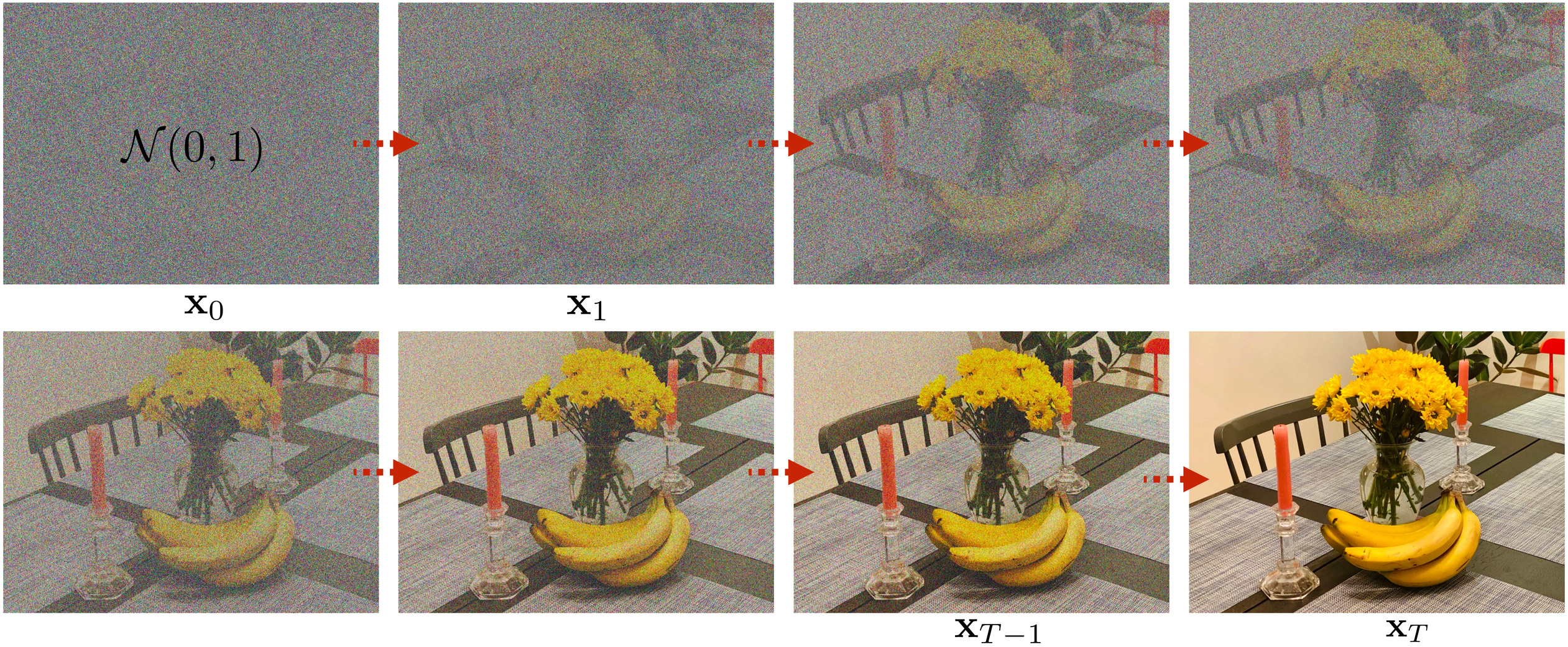
$\mathbf{x}_T$ $\mathbf{x}_{T-1}$ $\mathbf{x}_1$ $\mathbf{x}_0$

Diffusion-based image synthesis

Reverse: iteratively remove noise from random sample to obtain image from p(x)

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\ \mathbf{z}_i, \quad i = 0, 1, \cdots, T$$

("score function")



Guided diffusion

- Assume we know $p(\mathbf{y} \mid \mathbf{x})$ for random variables x and y (e.g., a classifier)
  - Example: x is an image, y is a string describing the image
  - Given an image (x), infer a caption (y)

$$p(\mathbf{x} \mid \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})\mathrm{d}\mathbf{x}$$ **(Bayes Rule)**

Bayes for score function

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})$$

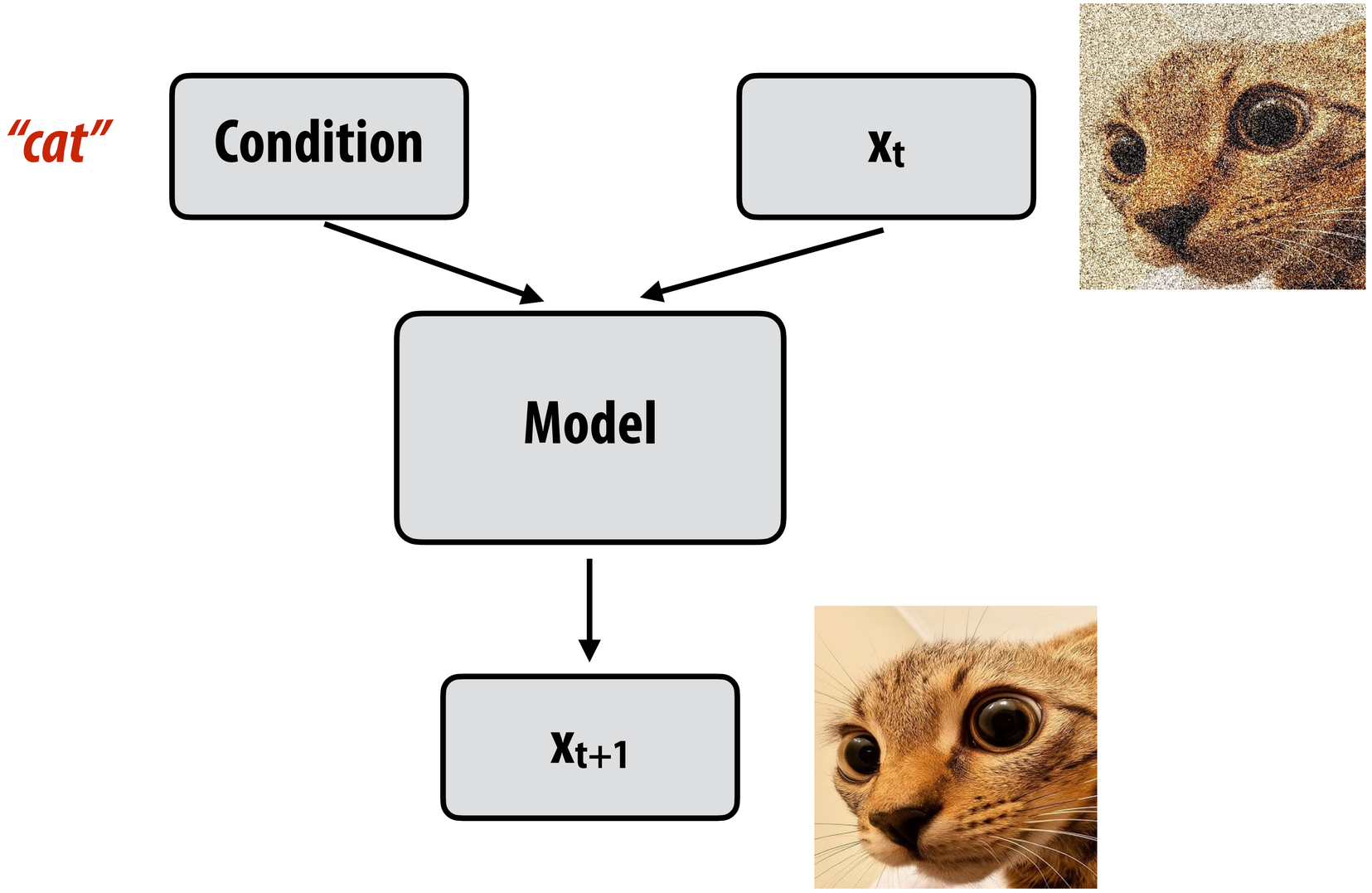
(Unguided score function)

Modify image x so that image is more likely [to come from the training set]

(Prompt guidance)

Modify image x to make the prompt a more likely description of the image

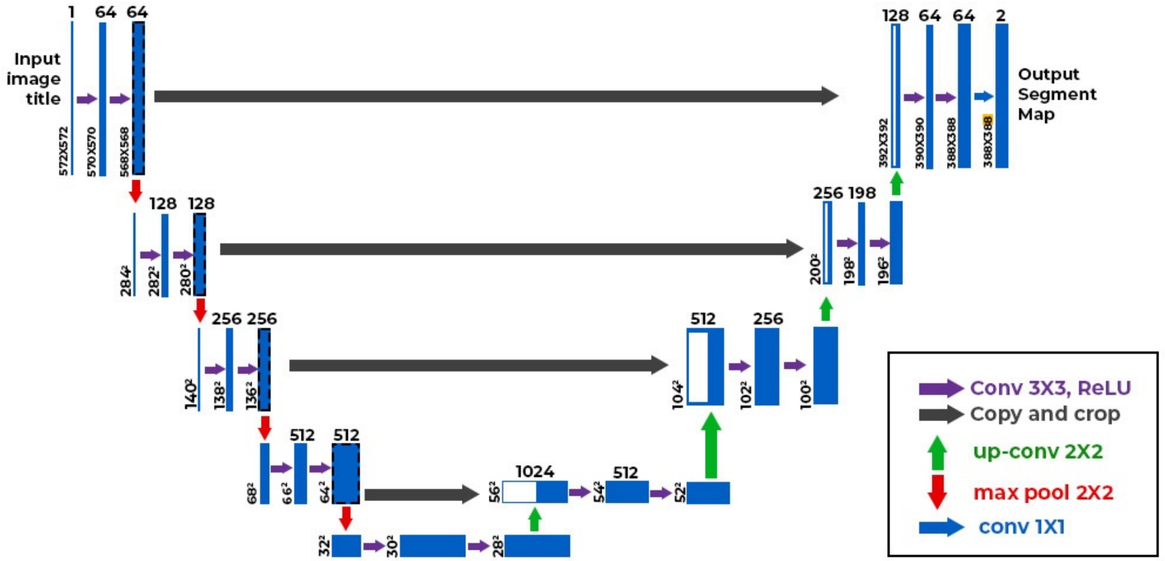
Conditioned generation



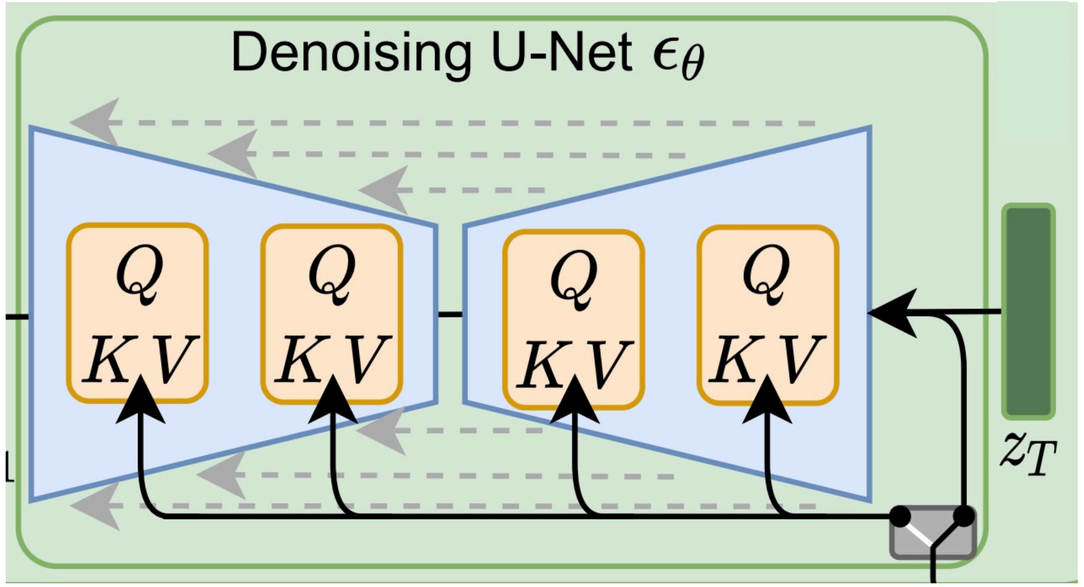
Performance/efficiency optimizations

A basic U-NET architecture

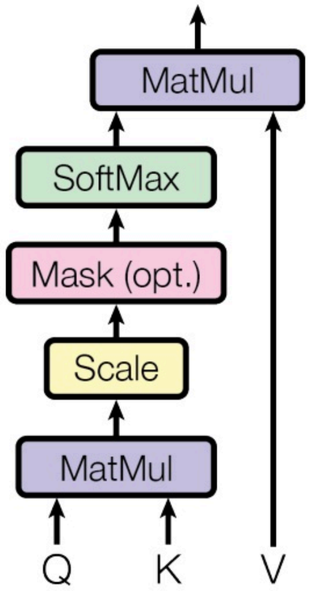
Classic U-Net for image segmentation
(Basic blocks are convolutional layers)



U-Net used in modern diffusion models
(Basic blocks are transformer modules with cross-attention layers)



Attention module



Let $\mathbf{S} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N \times N}$

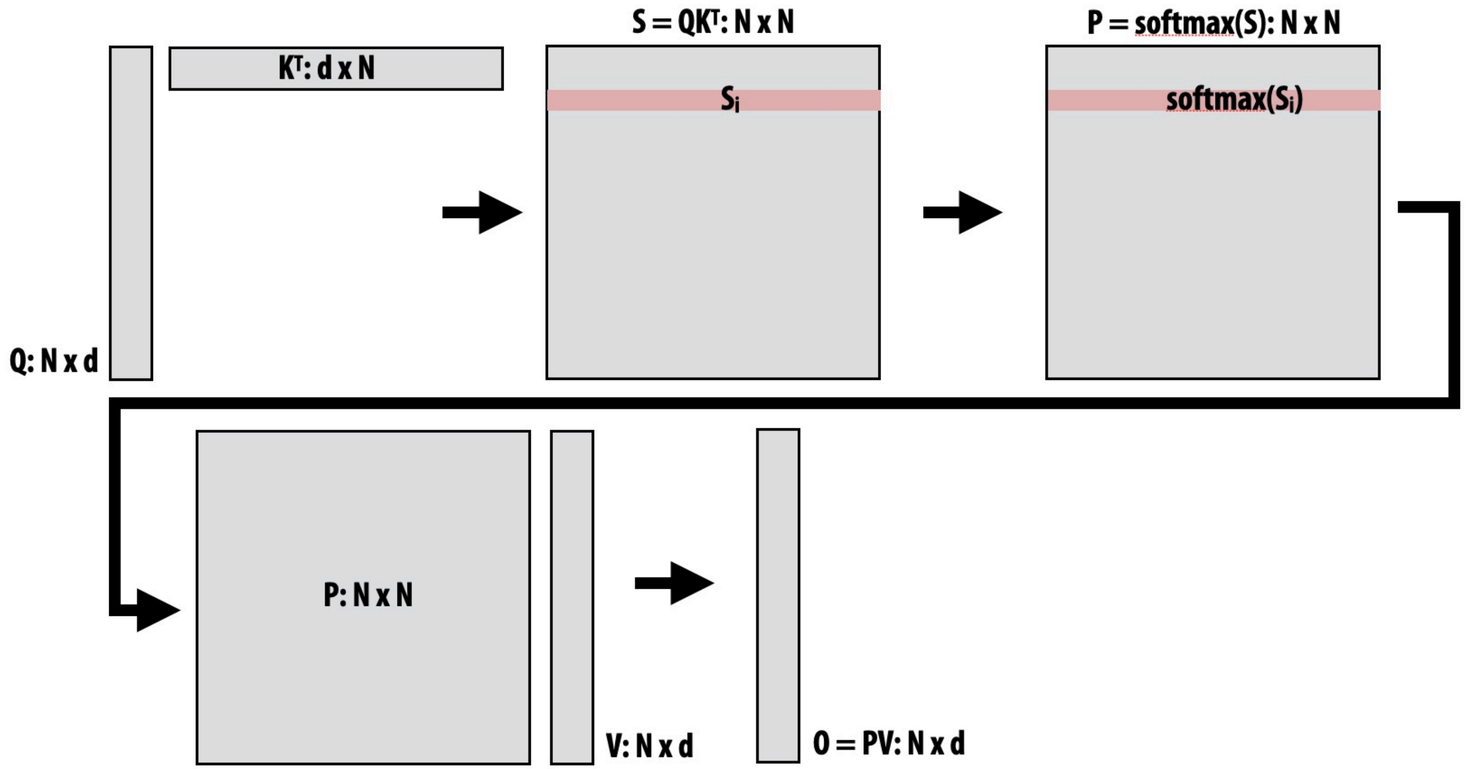
Let $\mathbf{P} = \mathrm{softmax}(\mathbf{S}) \in \mathbb{R}^{N \times N}$

Let $\mathbf{O} = \mathbf{P}\mathbf{V} \in \mathbb{R}^{N \times d}$

Where

$\mathrm{softmax}(\mathbf{S})$

Is softmax over the rows of S



For a row x:

$$\mathrm{softmax}(\mathbf{x}) = \frac{f(\mathbf{x})}{l(\mathbf{x})}$$

Where:

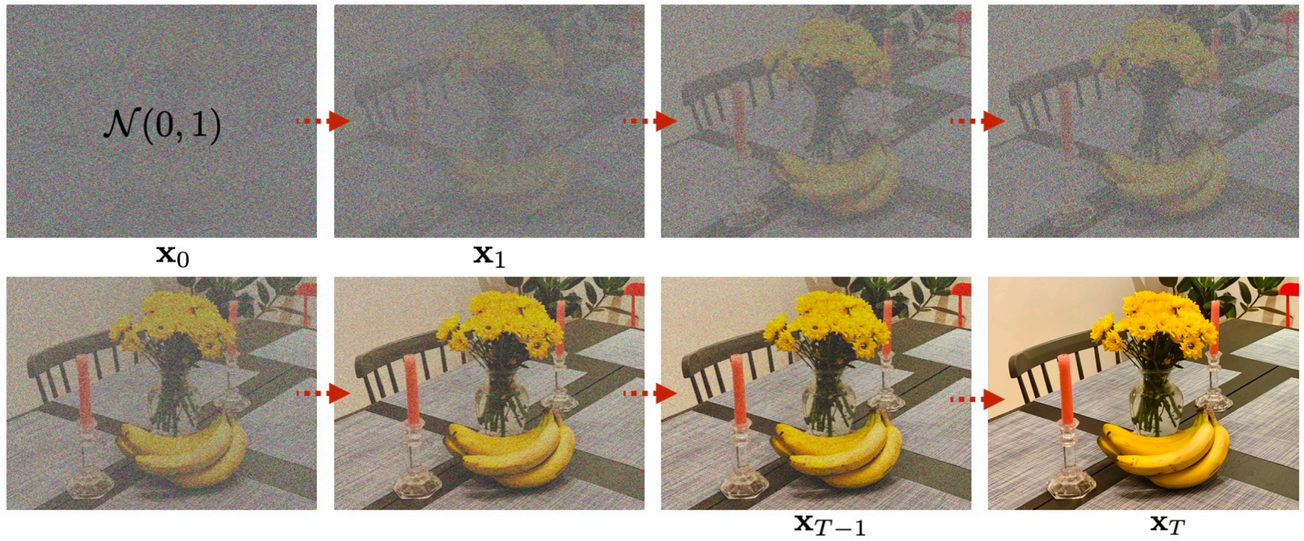
$$f(\mathbf{x}) = \begin{bmatrix} e^{\mathbf{x}_1 - m(\mathbf{x})} & e^{\mathbf{x}_1 - m(\mathbf{x})} & \dots & e^{\mathbf{x}_B - m(\mathbf{x})} \end{bmatrix}$$

$$m(\mathbf{x}) = \max_i(\mathbf{x}_i)$$

$$l(\mathbf{x}) = \sum_i f(\mathbf{x})_i = \sum_i e^{\mathbf{x}_i - m(\mathbf{x})}$$

Performance challenges of inference

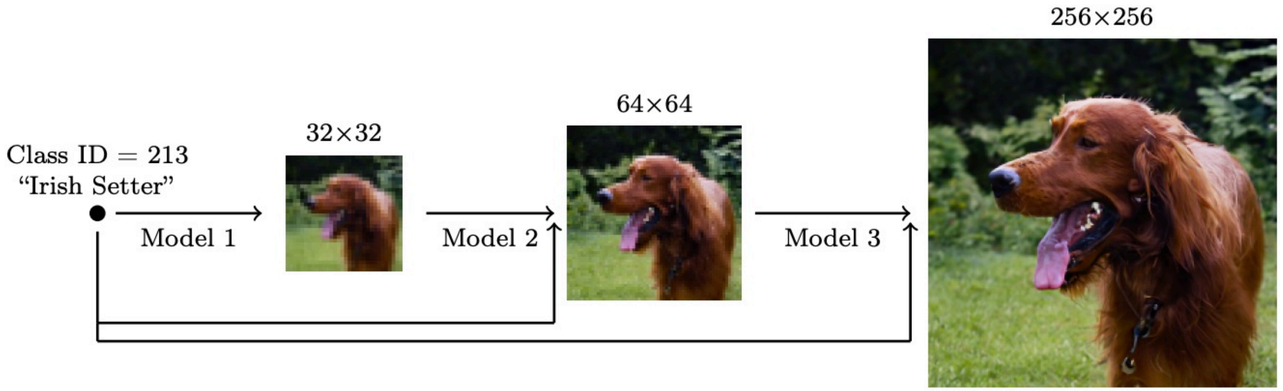
- **Diffusion is an iterative process:**
  - Requires many steps to convergence



- **Ways to improve inference efficiency**
  - Diffuse in low-dimensional (latent) space
  - Superresolution techniques
  - Learn to take bigger steps

Superresolution

- Use diffusion to produce a low-resolution image
- Then subsequent models perform neural superresolution

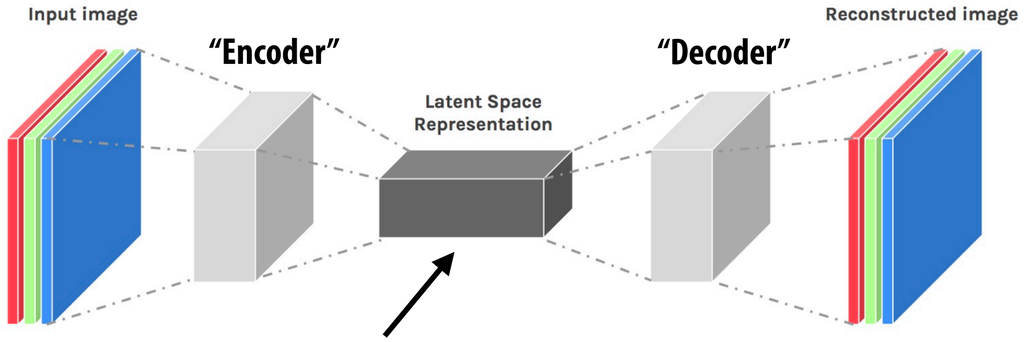


Cascade of diffusion models

Bicubic upsampling vs. two forms of learned upsampling

Perform diffusion in lower-dimensional latent space

- Main idea: perform diffusion in the lower dimensional latent space of images, not in high-dimensional RGB pixel space
- After diffusing a latent representation, "decode" latent to final image



If this latent representation is compact, then it is a compressed representation of the input image

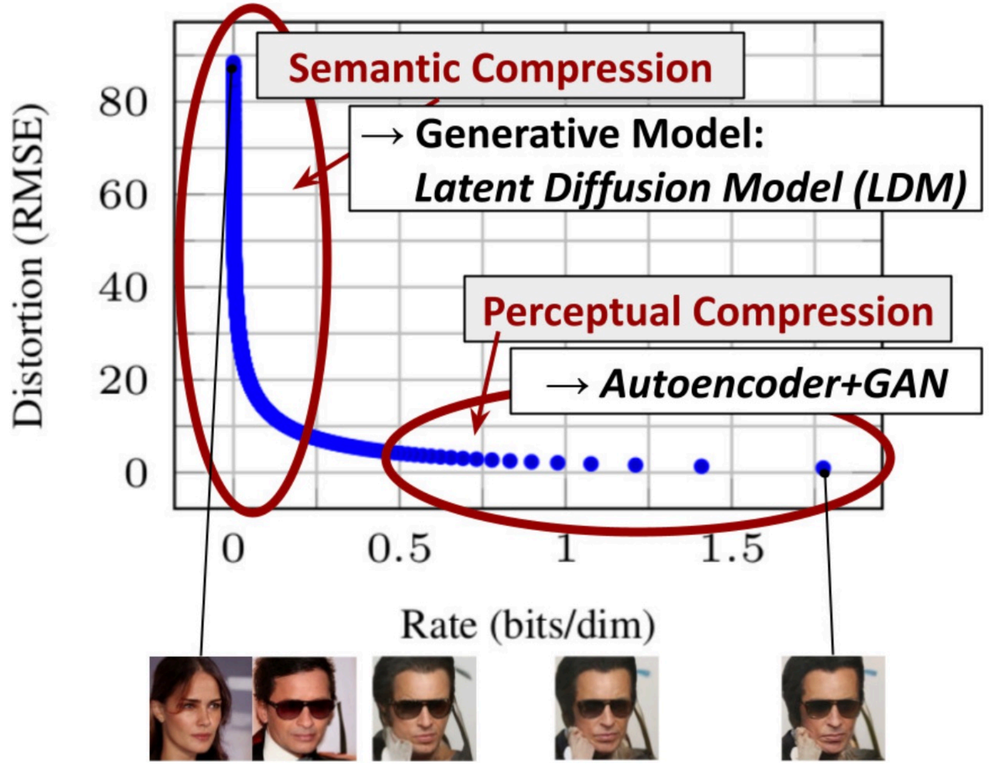
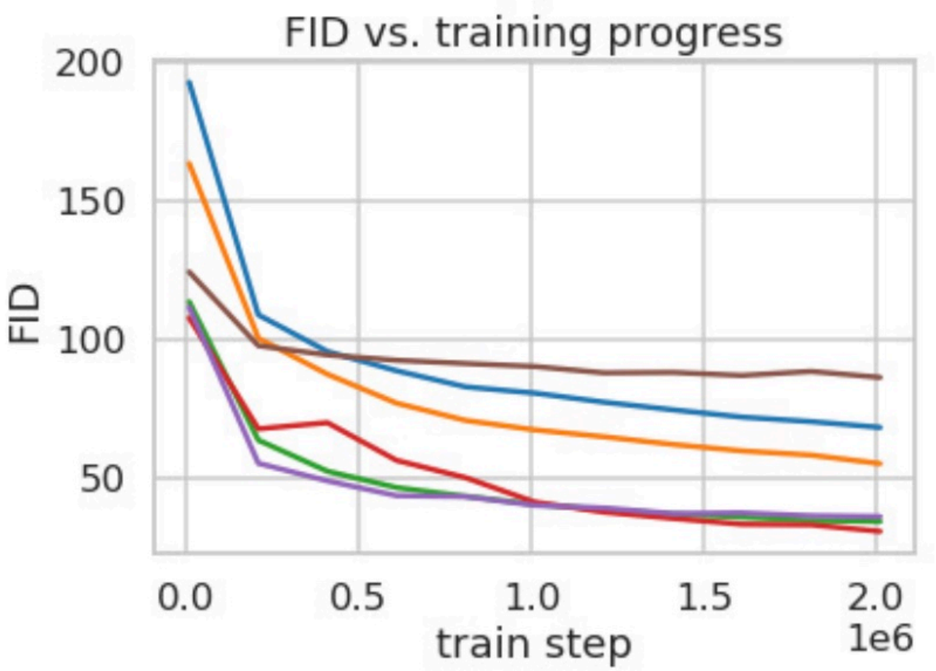


Figure credit: https://medium.com/@birla.deepak26/autoencoders-series-daad78df9350

Perform diffusion in latent space

- Implications to both training efficiency and inference efficiency



Per-pixel representations, can represent data well, but require significant training to learn good models

"Sweet spot": learns good model + trains quickly

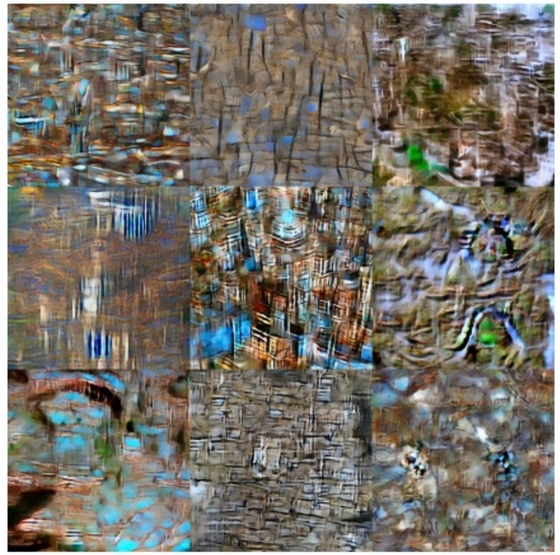
Latent representation too compressed (cannot represent data well)

Learn to take larger steps

- Given a diffusion model, learn a new (second) diffusion model that reproduces multiple steps of the diffusion process
- This is a form of "model distillation": training a "student" model to emulate the output of a teacher. Here, the teacher's output is multiple steps of the diffusion process

Ours 2 steps

Ours 4 steps



DDIM 2×2 steps

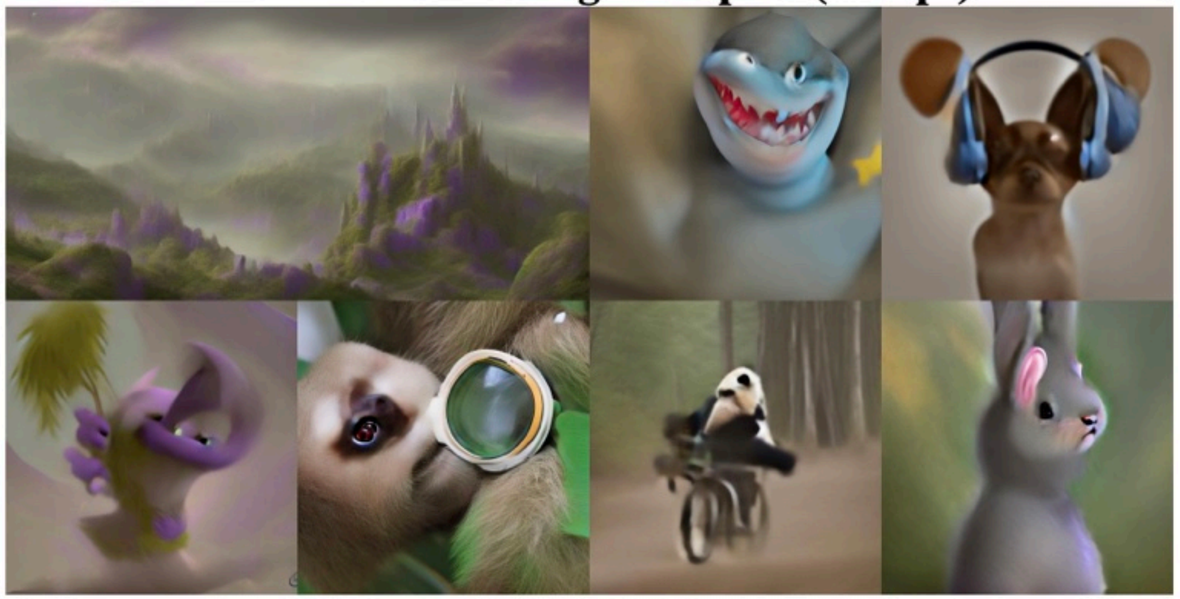
DDIM 4×2 steps

Prompt: "*A beautiful castle, matte painting.*"

Learn to take larger steps

Distilled Text-to-Image samples (4 steps)

Native Text-to-Image samples (4 steps)



Native Text-to-Image samples (8 steps)