Lecture 17:

Generative AI for Image Creation - Part II
(More detail on Control and Performance Optimization)

Visual Computing Systems
Stanford CS348K, Spring 2023

Review: diffusion-based image synthesis

Idea: iterative MCMC process to generate a sample x (an image) from distribution p(x) of observed images

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

Review: diffusion-based image synthesis

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

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

("score function")

Guided diffusion

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

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

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)

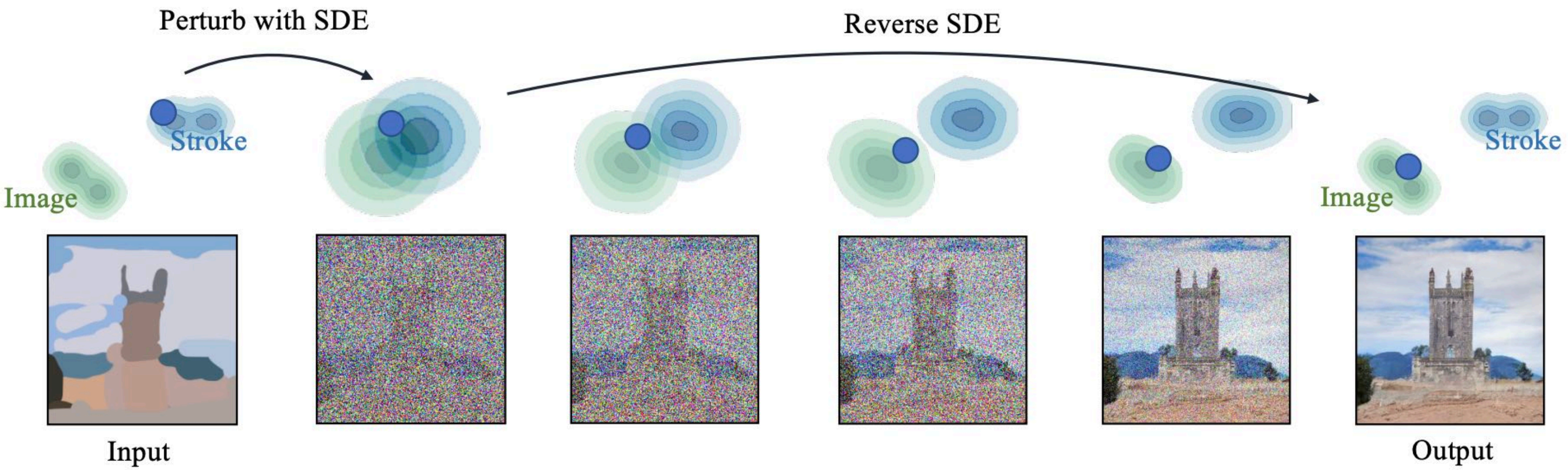
Modify image x to make the prompt a more likely description of the image

Controlling the output of diffusion models

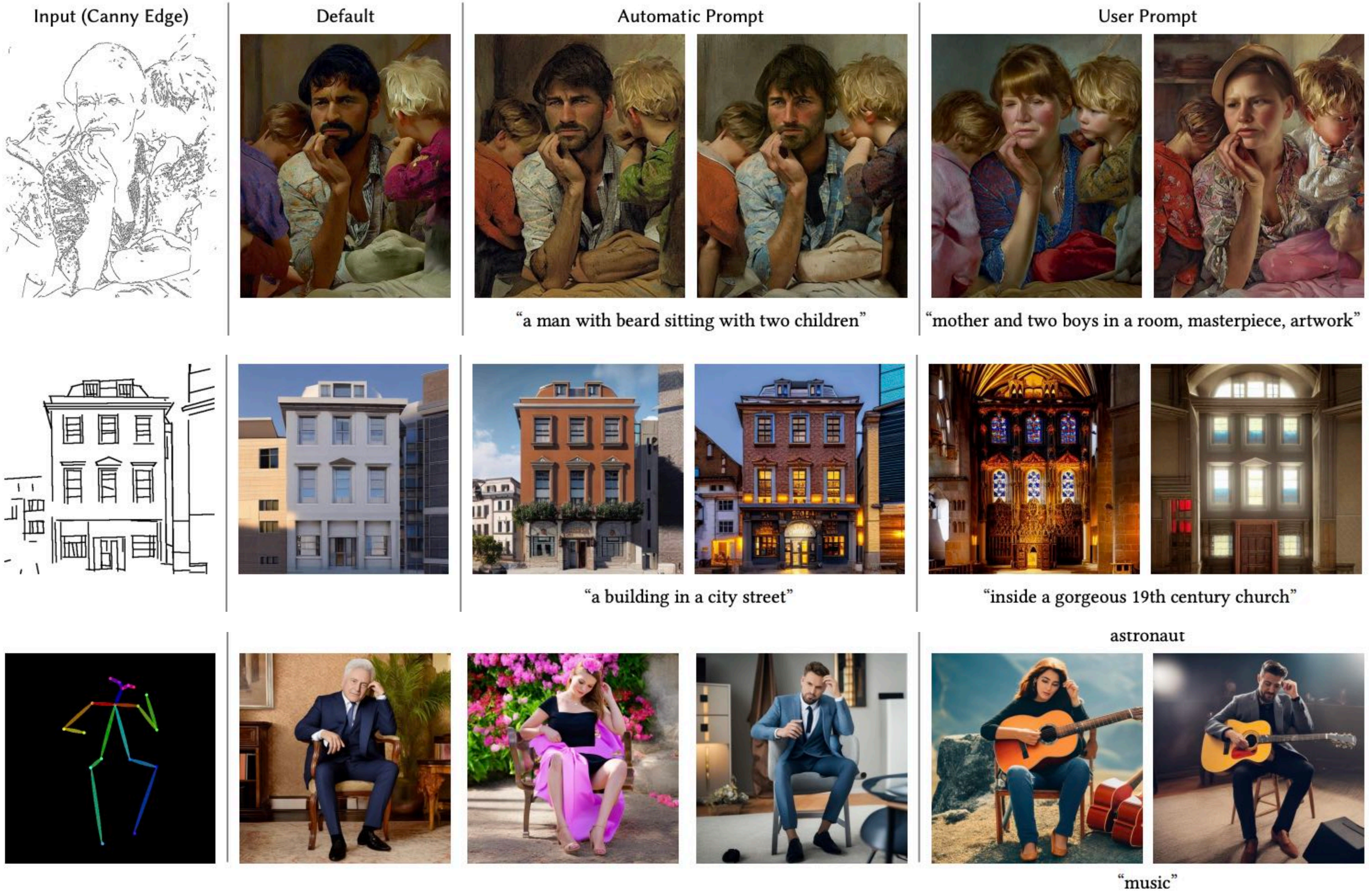
img2img (enabling image-based guidance)

1. Start with a guide image (a target)
2. Add "small" amount of noise
3. Iteratively denoise to produce sample from p(x)

"Guide toward a visual target"



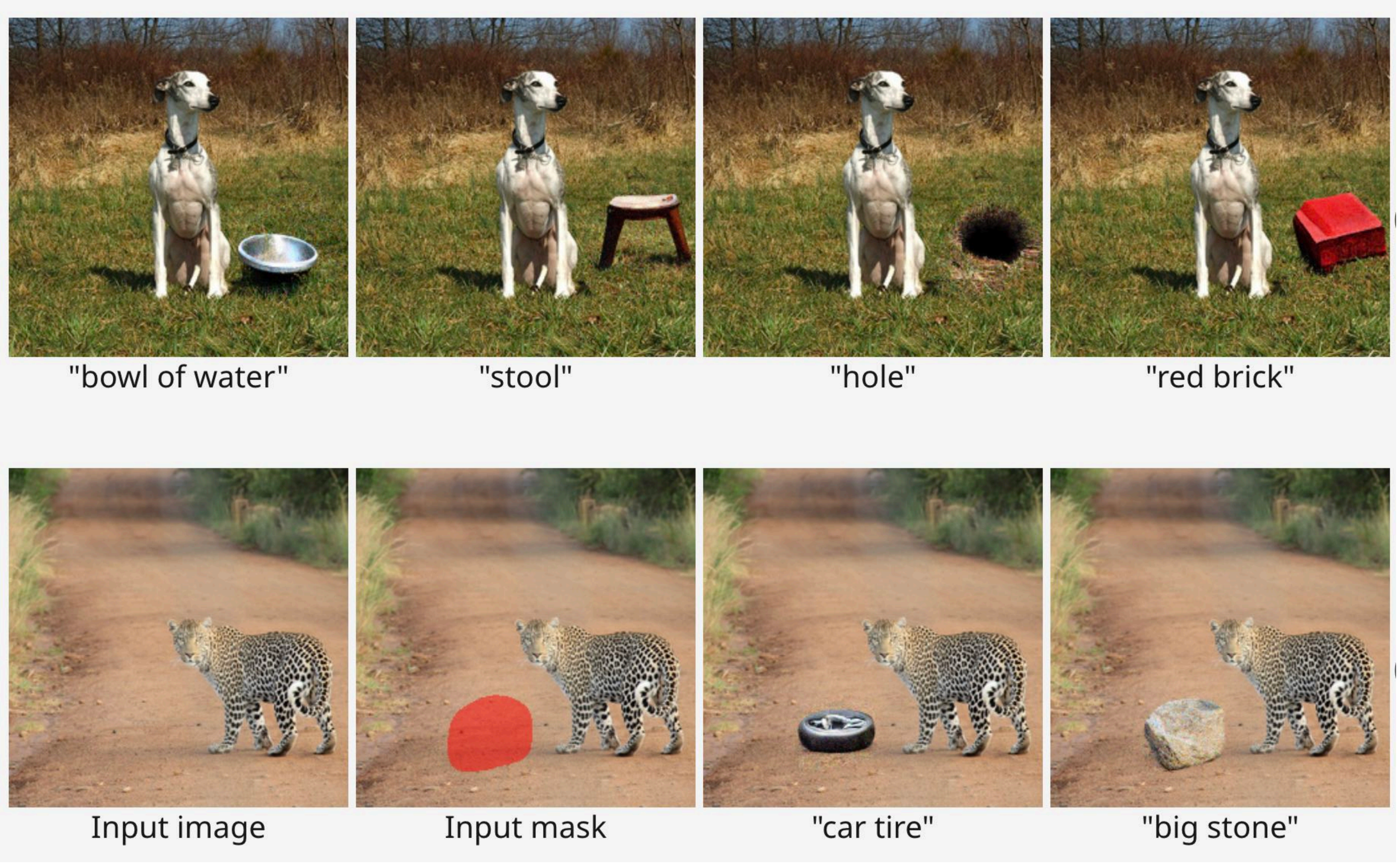
Other forms of guidance (not just text)



Inpainting (apply [new] prompt to a region)

User specifies mask for region of interest and text prompt for that region.

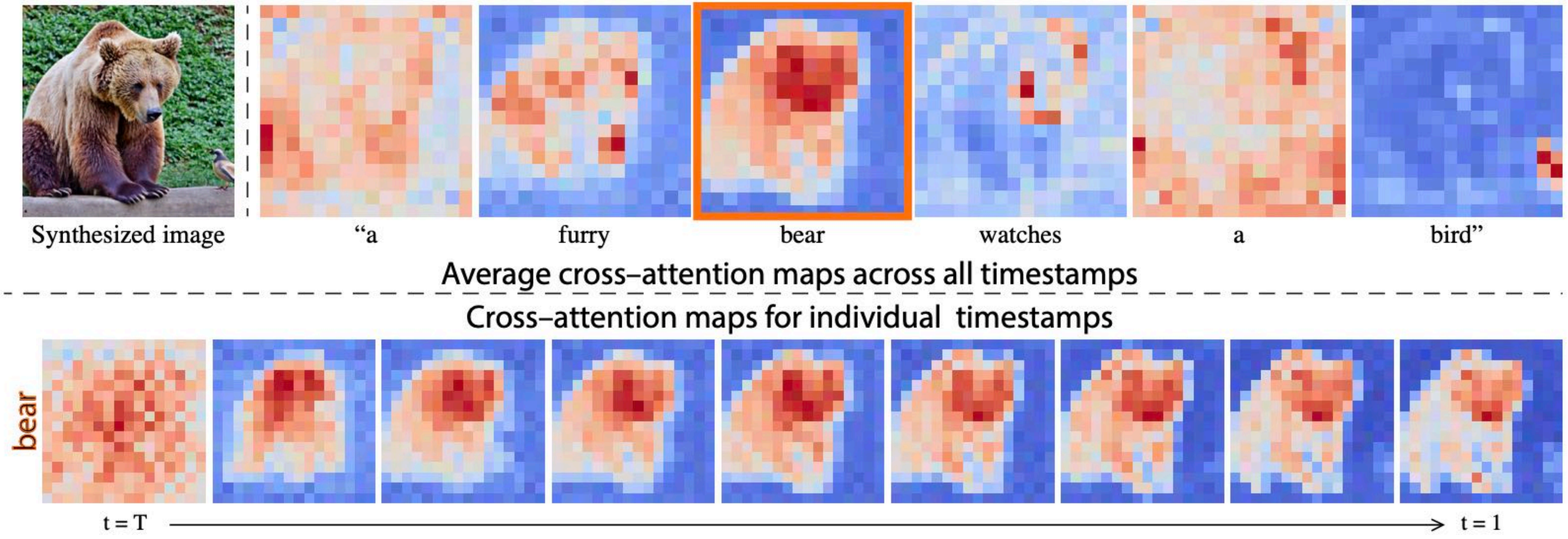
Image outside of region remains almost the same.



Use change in text prompt to trigger change in image

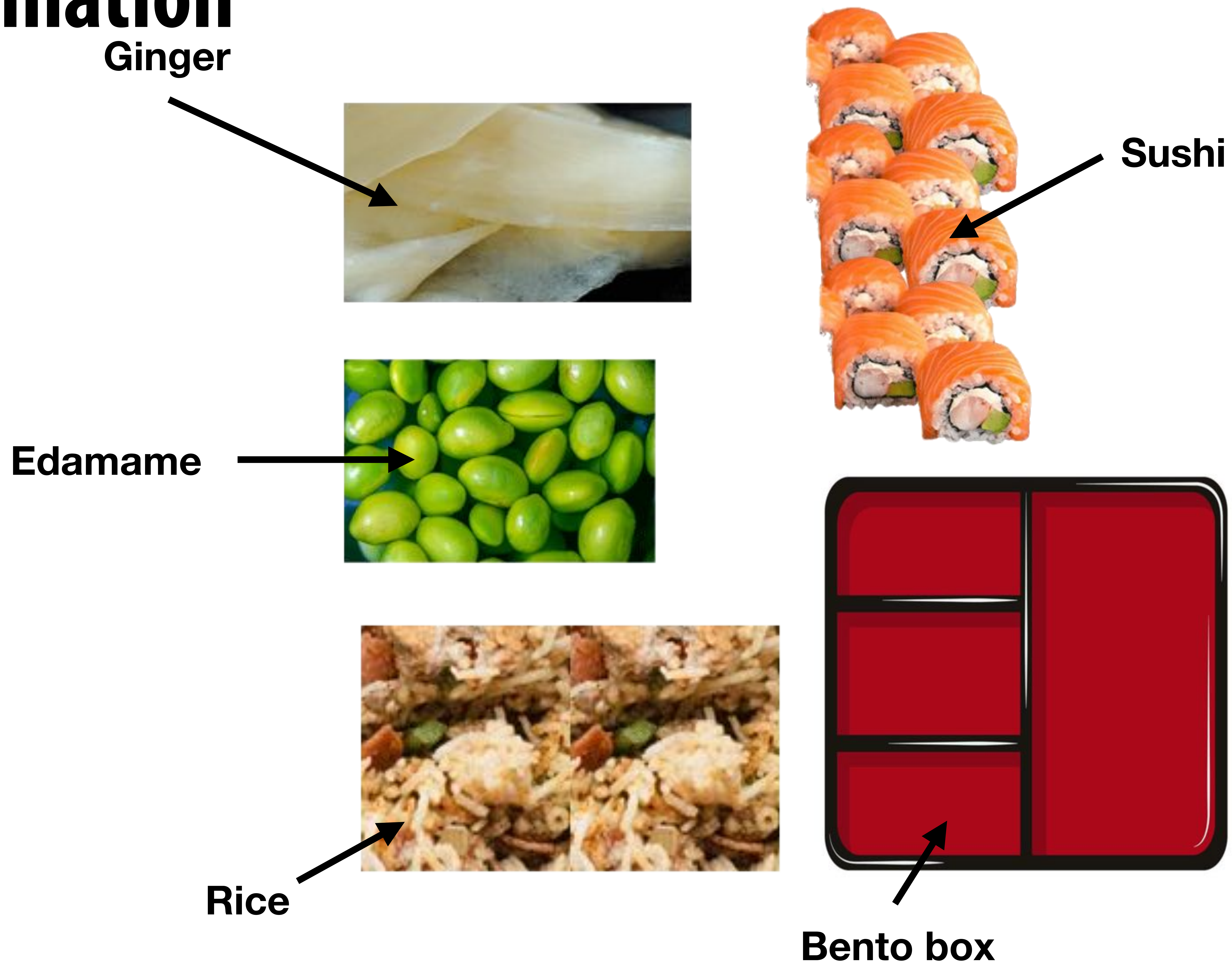
"Masks" come from learned attention

- Use masks from original generation process to constrain what pixels can change after prompt is edited



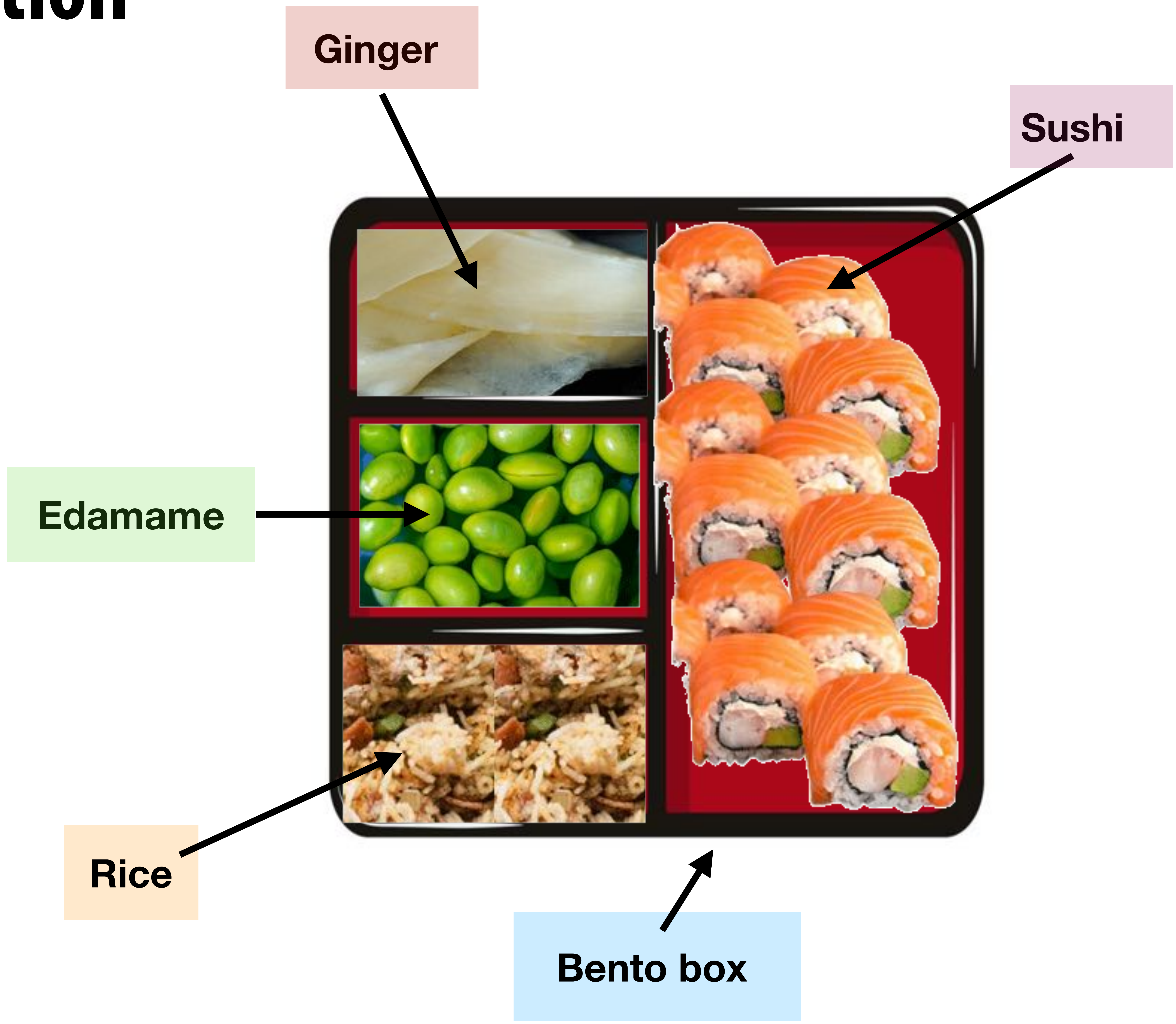
Leveraging layer information

Prompt + a rgba per layer



Leveraging layer information

"A bento box with
rice,
edamame,
ginger, and
sushi."



Leveraging layer information

"A bento box with
rice,
edamame,
ginger, and
sushi."

Using text to describe how to change the image

"Swap sunflowers with roses"

"Add fireworks to the sky"

"Replace the fruits with cake"

"What would it look like if it were snowing?"

"Turn it into a still from a western"

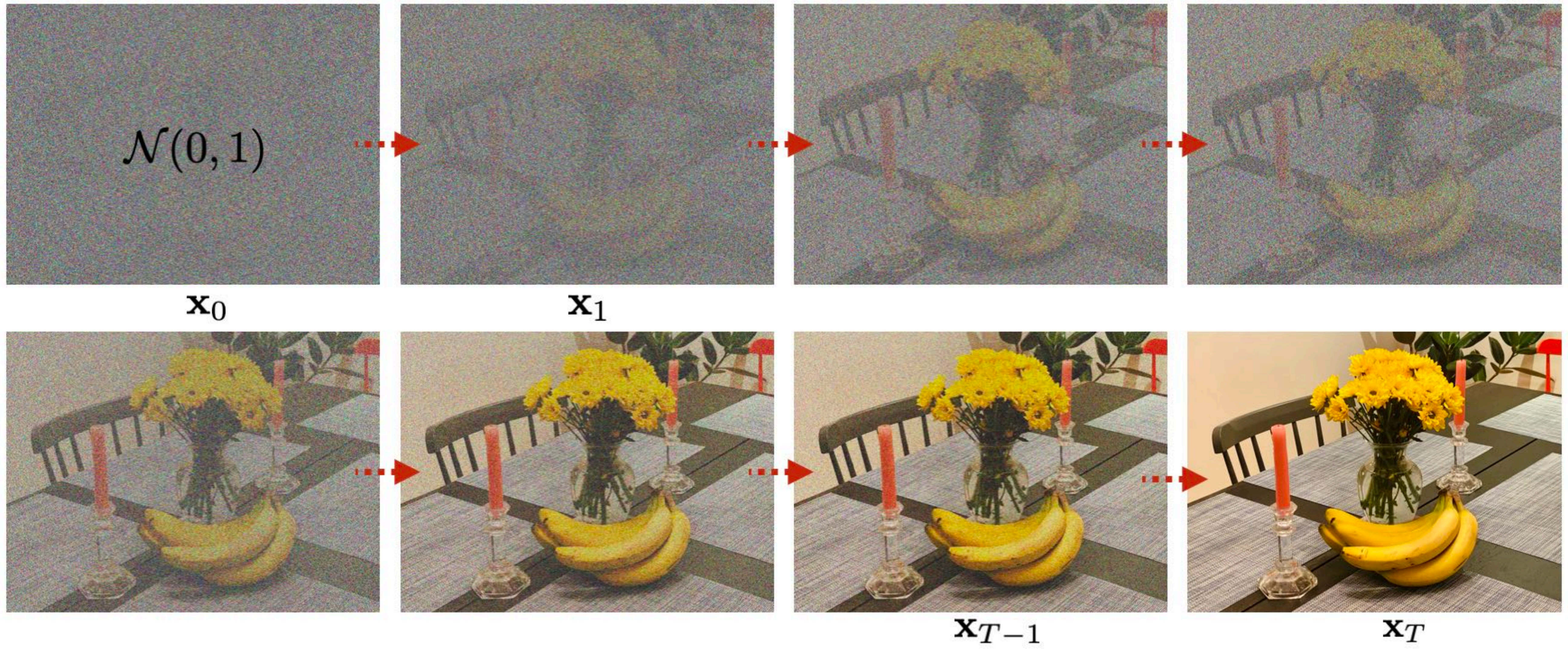
"Make his jacket out of leather"

Performance/efficiency optimizations

Challenge

- **Diffusion is an iterative process:**
  - Many steps to convergence
  - Each step involves evaluating up to two diffusion models

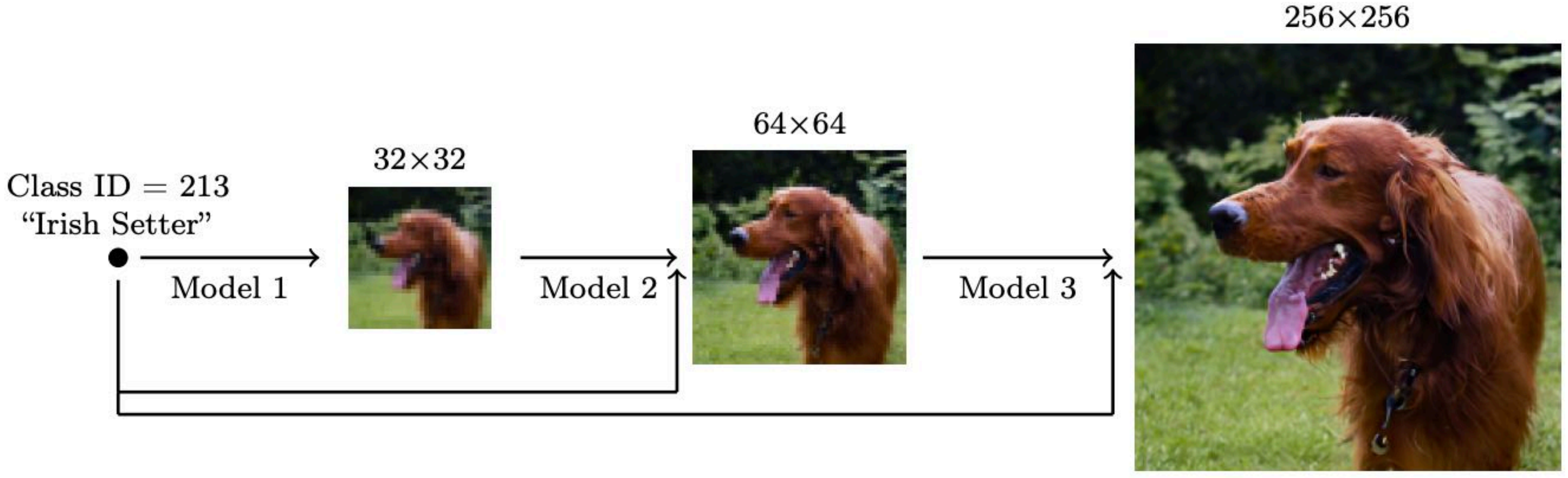
    Recall score function: $$\nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})$$



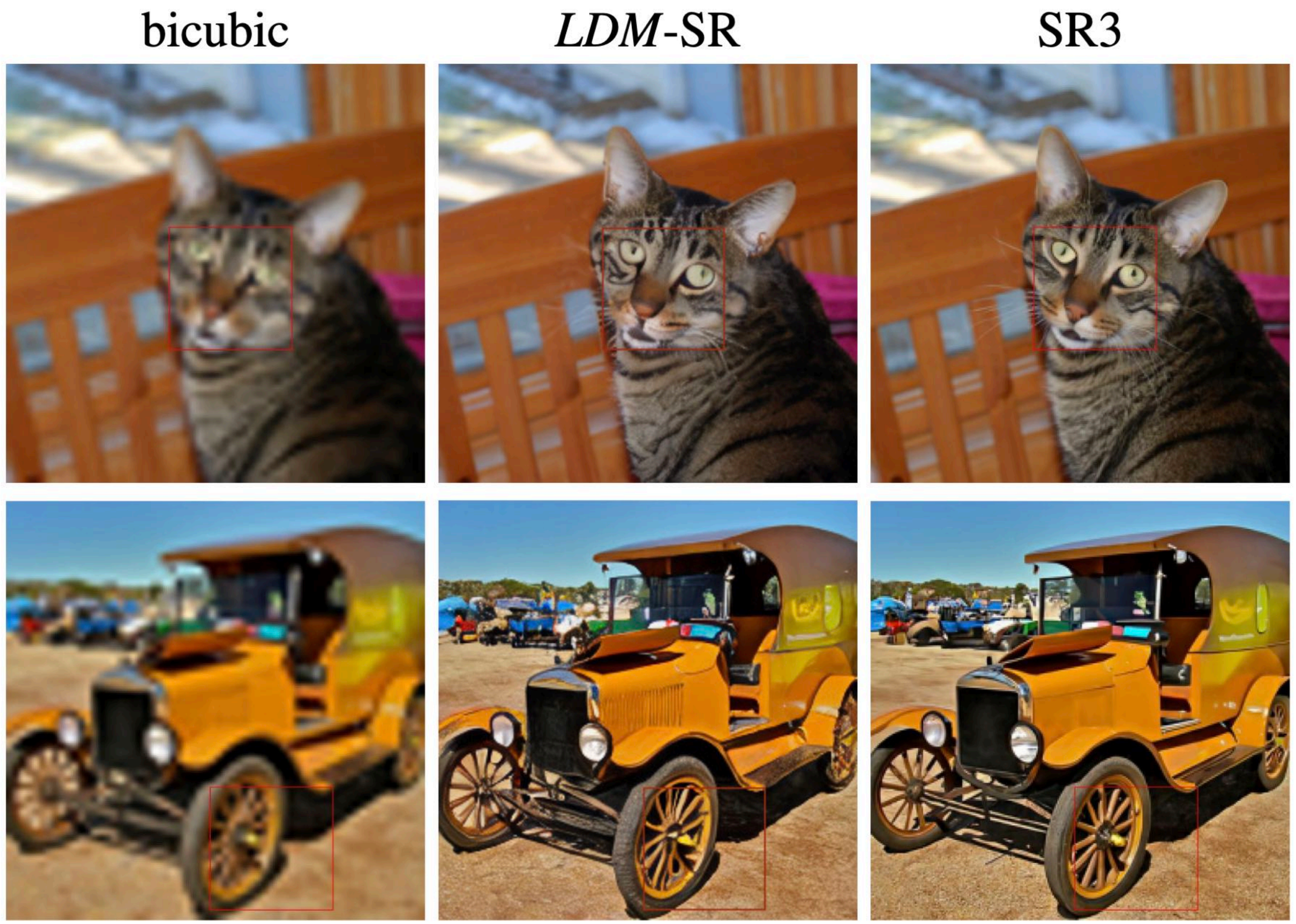
- **Ways to improve inference efficiency**
  - Diffuse in latent space
  - Superresolution techniques
  - Learn to take bigger steps

Superresolution

- Diffusion produces low-resolution image
- Then subsequent models perform neural superresolution
  - Other diffusion models take low res to high res
  - Other other super resolution technique



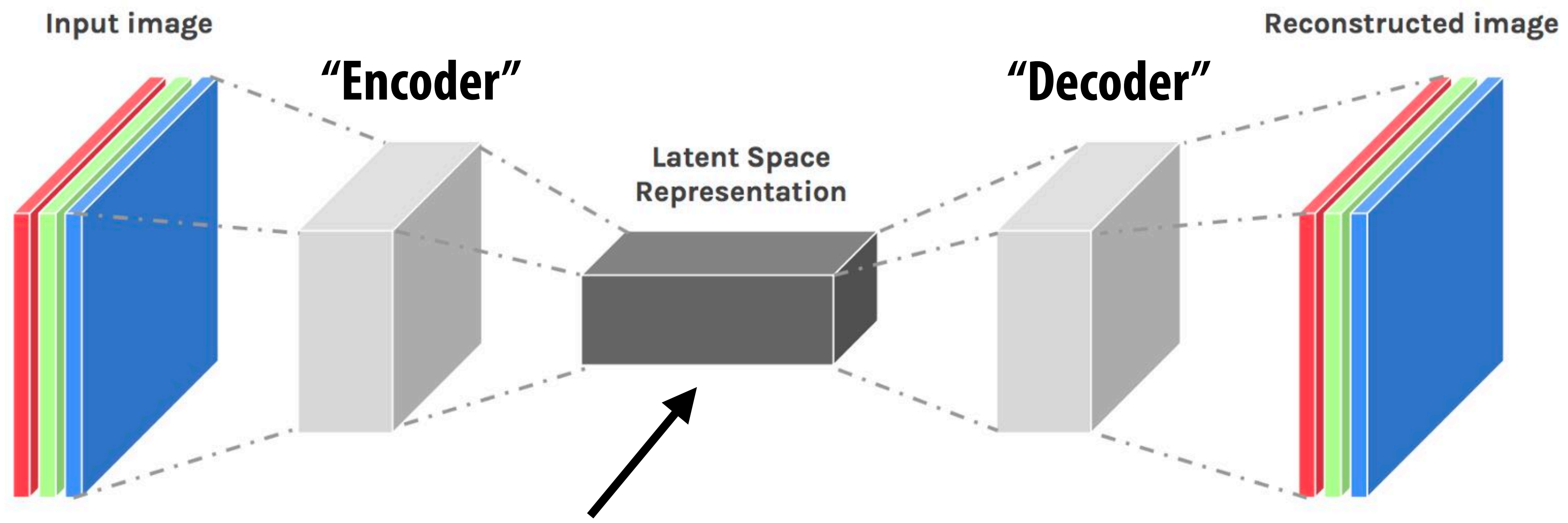
Cascade of diffusion models



Bicubic upsampling vs. two forms of learned upsampling

Perform diffusion in latent space

- Main idea: perform diffusion in the lower dimension latent space of images, not in high-dimensional 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

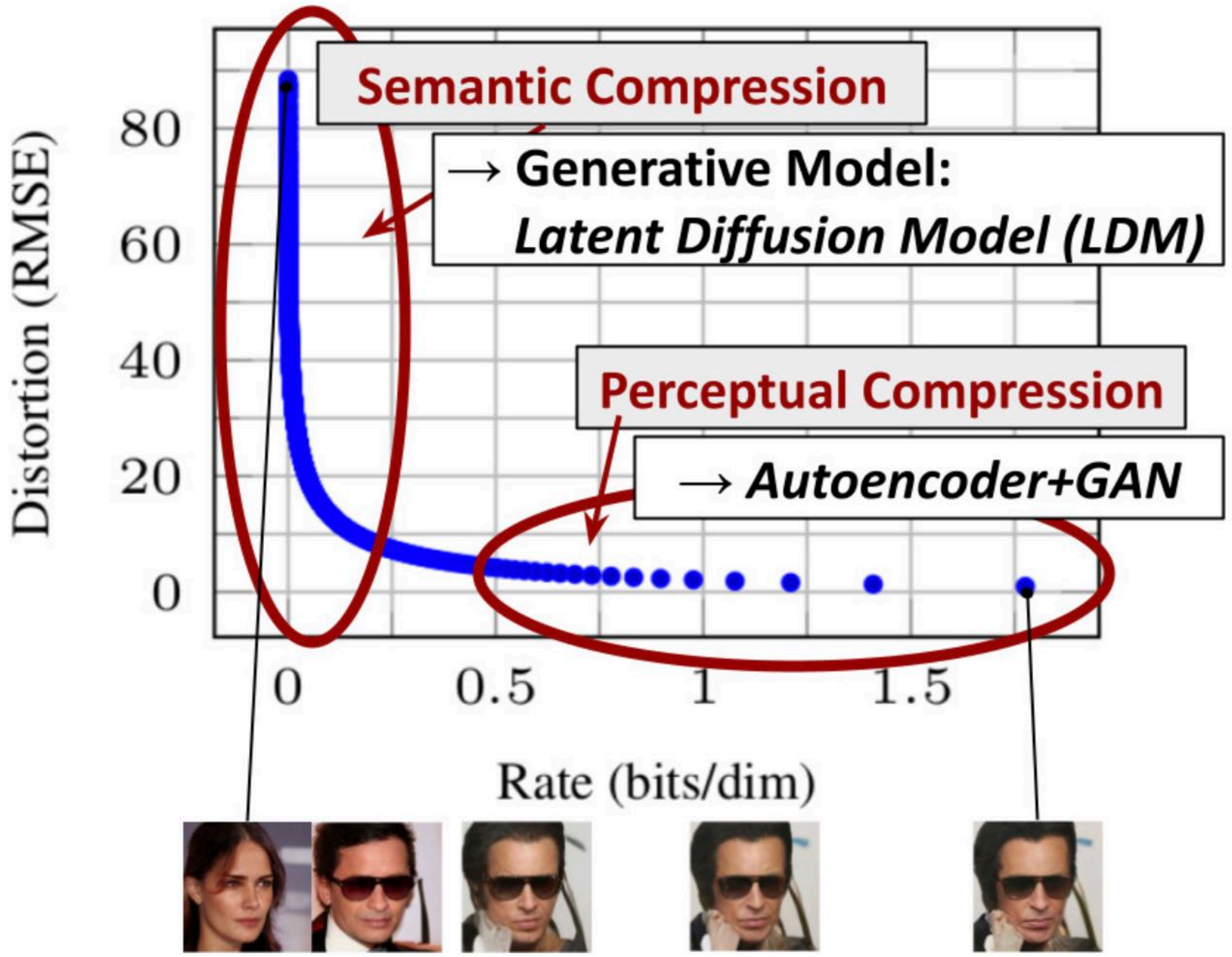
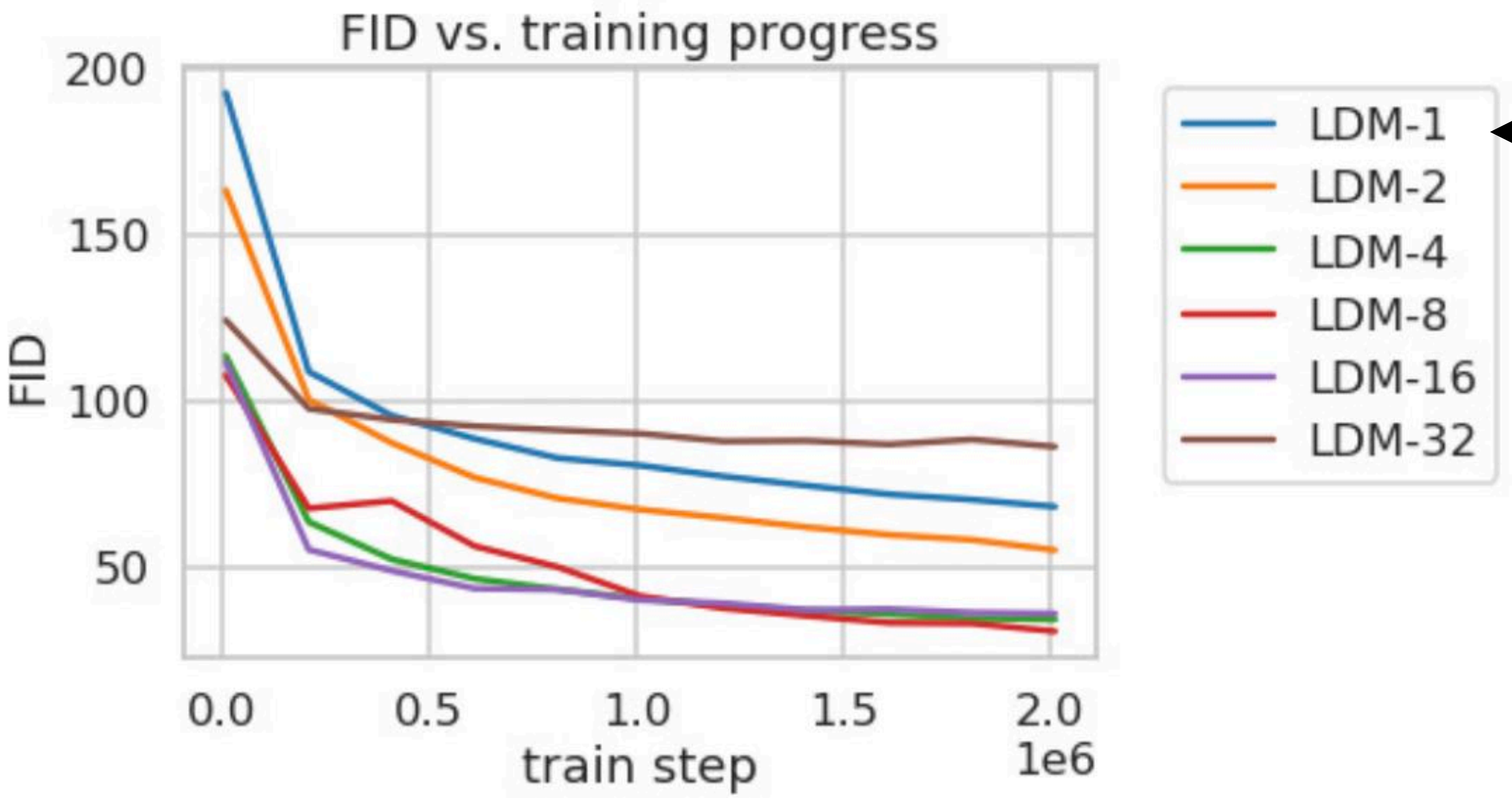


Figure credit: https://medium.com/@birla.deepak26/autoencoders-series-daad78df9350

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

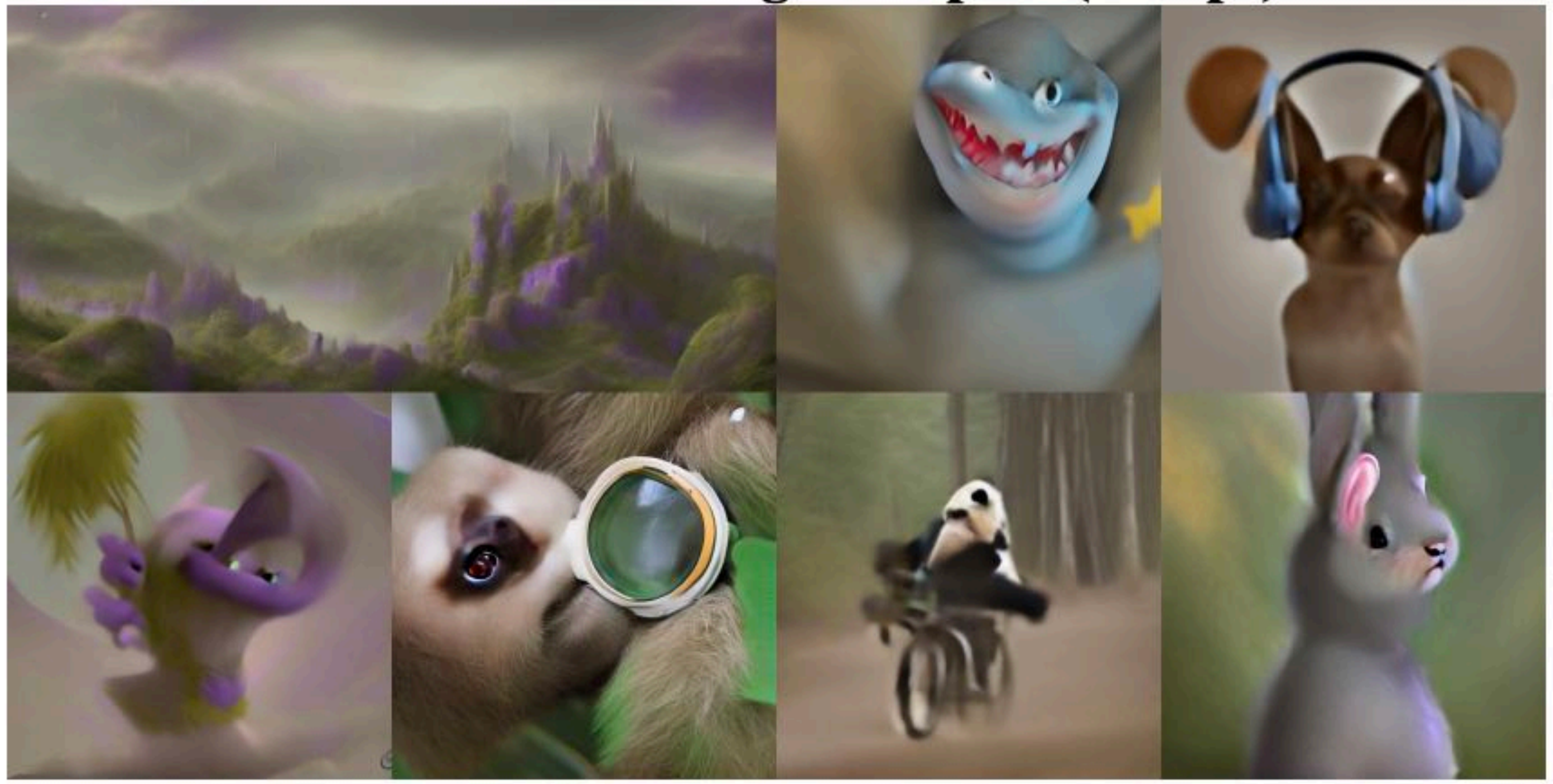
Ours 2 steps   Ours 4 steps   DDIM 2×2 steps   DDIM 4×2 steps

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)

Summary

- Diffusion-based generation produces high quality ("plausible") image output
  - Step 1: get generation to produce output that models the training data well

- Step 2: ongoing research on
  - New ways to help users exert guide/control over the generation process
  - Improving the efficiency of diffusion model training/evaluation

- This line of work is a great example of many of the issues and concerns we've discussed in this class
  - Are we optimizing for the right metrics?
  - How to achieve high performance (through better algorithms, or systems techniques)
  - What are the implications of these technologies?