# A practitioner's guide to diffusion models

## CS180/280A, 11/4

Aleksander Holynski

A racoon wearing a tuxedo and top hat with the moon in the background.


# What are diffusion models?
# How do I train one?
# How do I use them?

# What’s covered

- Intuition on diffusion models—how do they work?
- How to train one—(data, architecture, training processes)
- How to use them—(sampling, guidance, distillation, zero-shot editing)
- What can they be used for? (editing, image priors, controllability)

# What's not covered: *most of the math*

Understanding Diffusion Models: A Unified Perspective

**Calvin Luo**
Google Research, Brain Team
calvinluo@google.com

August 26, 2022

Contents



## Tutorial on Diffusion Models for Imaging and Vision

Stanley Chan[1]

March 28, 2024

**Abstract**. The astonishing growth of generative tools in recent years has empowered many exciting applications in text-to-image generation and text-to-video generation. The underlying principle behind these generative tools is the concept of *diffusion*, a particular sampling mechanism that has overcome some shortcomings that were deemed difficult in the previous approaches. The goal of this tutorial is to discuss the essential ideas underlying the diffusion models. The target audience of this tutorial includes undergraduate and graduate students who are interested in doing research on diffusion models or applying these models to solve other problems.

Contents



## Sander Dieleman

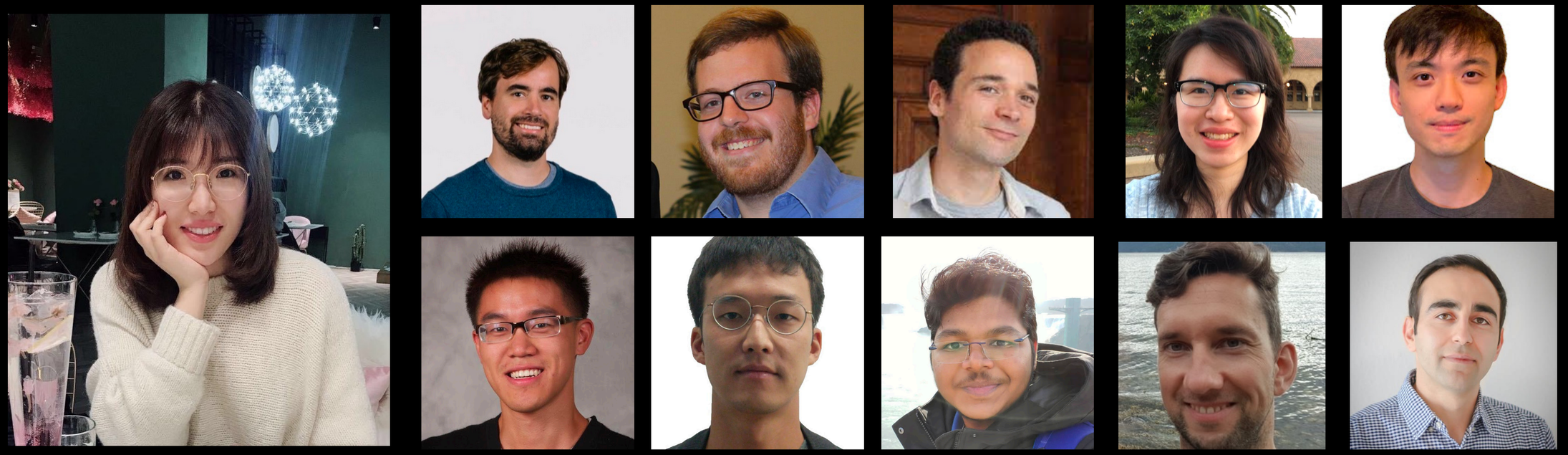
LATEST POSTS

FEBRUARY 28, 2024 BY SANDER DIELEMAN ⏲ READING TIME ~46 MINUTES

### The paradox of diffusion distillation

Diffusion models split up the difficult task of generating data from a high-dimensional distribution into many denoising tasks, each of which is much easier. We train them to solve just one of these tasks at a time. To sample, we make many predictions in sequence. This **iterative refinement** is where their power comes from.

# Acknowledgements



Some slides were borrowed from
Denoising Diffusion-based Generative Modeling CVPR2022 tutorial
(https://cvpr2022-tutorial-diffusion-models.github.io/)

# Part 1:
# What are diffusion models?

# Denoising Diffusion Models

## Emerging as powerful generative models, outperforming GANs

"Diffusion Models Beat GANs on Image Synthesis"
Dhariwal & Nichol, OpenAI, 2021

"Cascaded Diffusion Models for High Fidelity Image Generation"
Ho et al., Google, 2021

# Deep generative models

Learning to generate data

Samples from a Data Distribution

Train

Neural Network

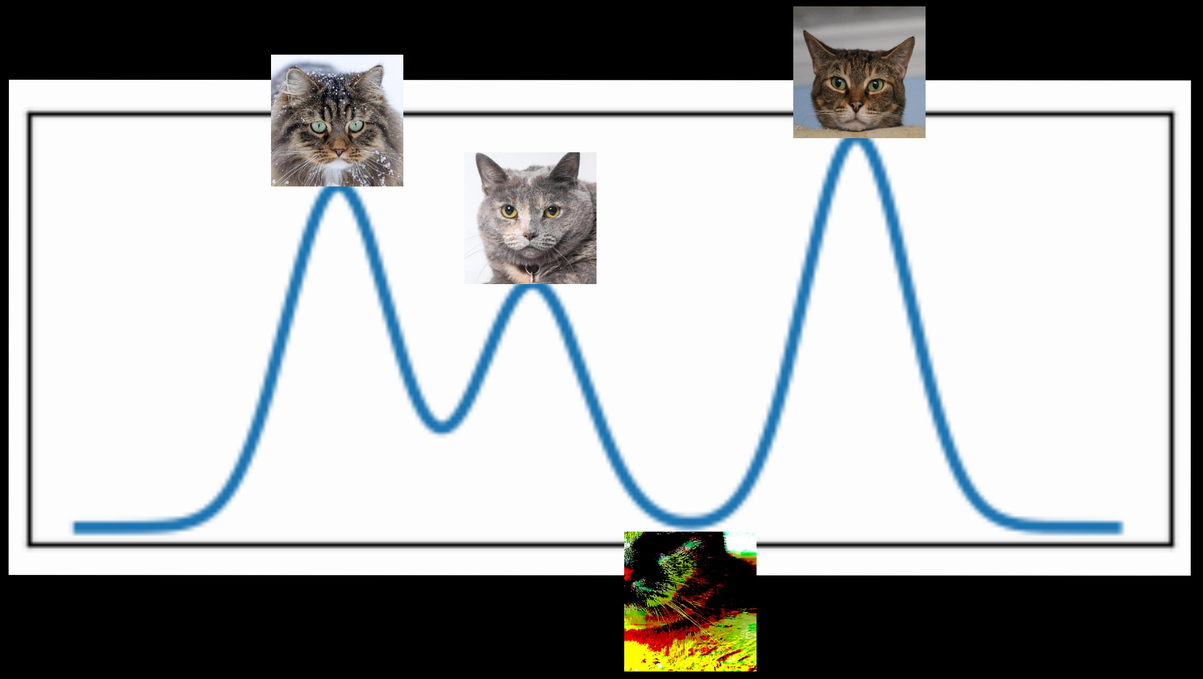
Sample

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Deep generative models

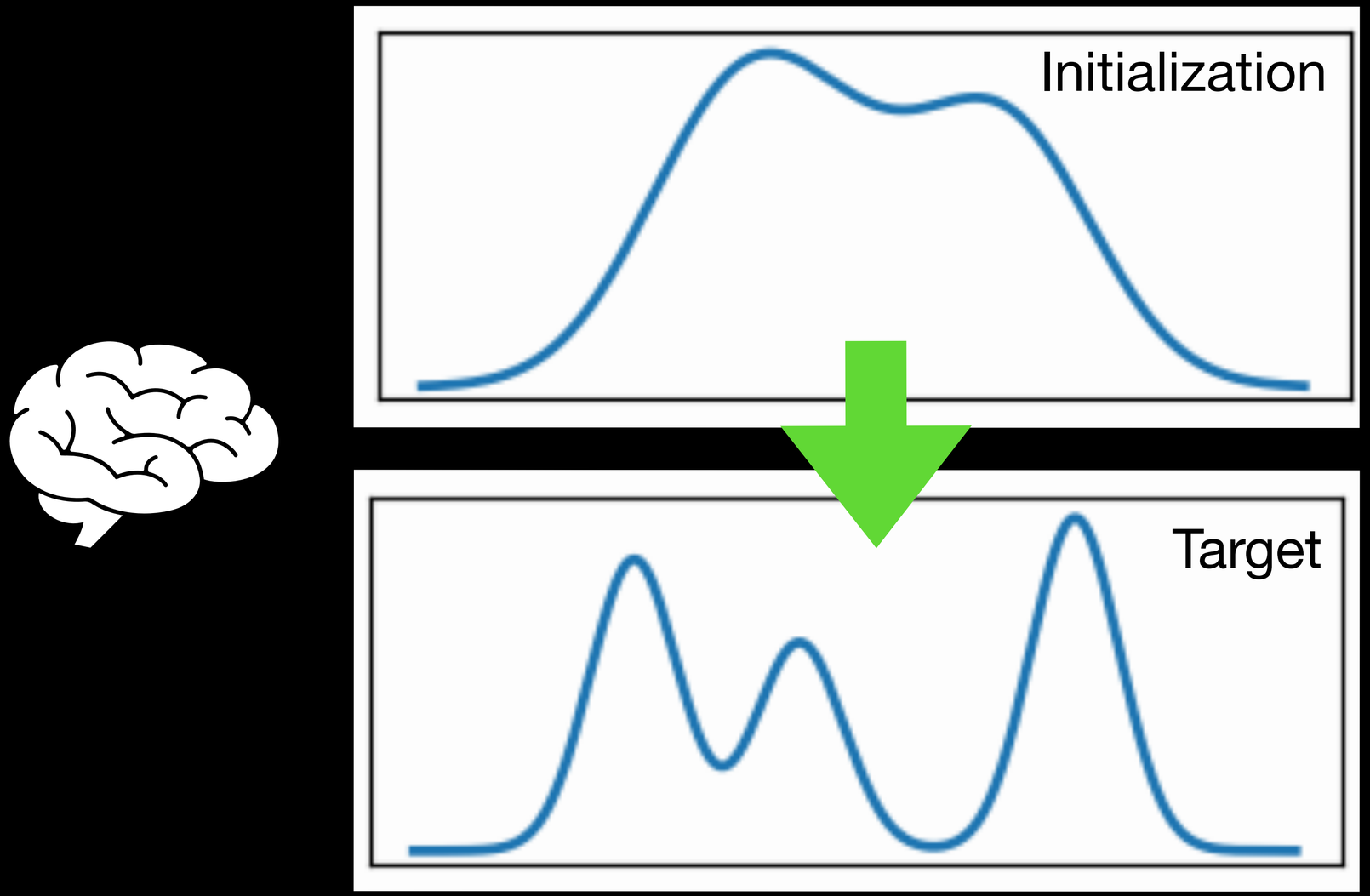
Learning to generate data

Neural Network



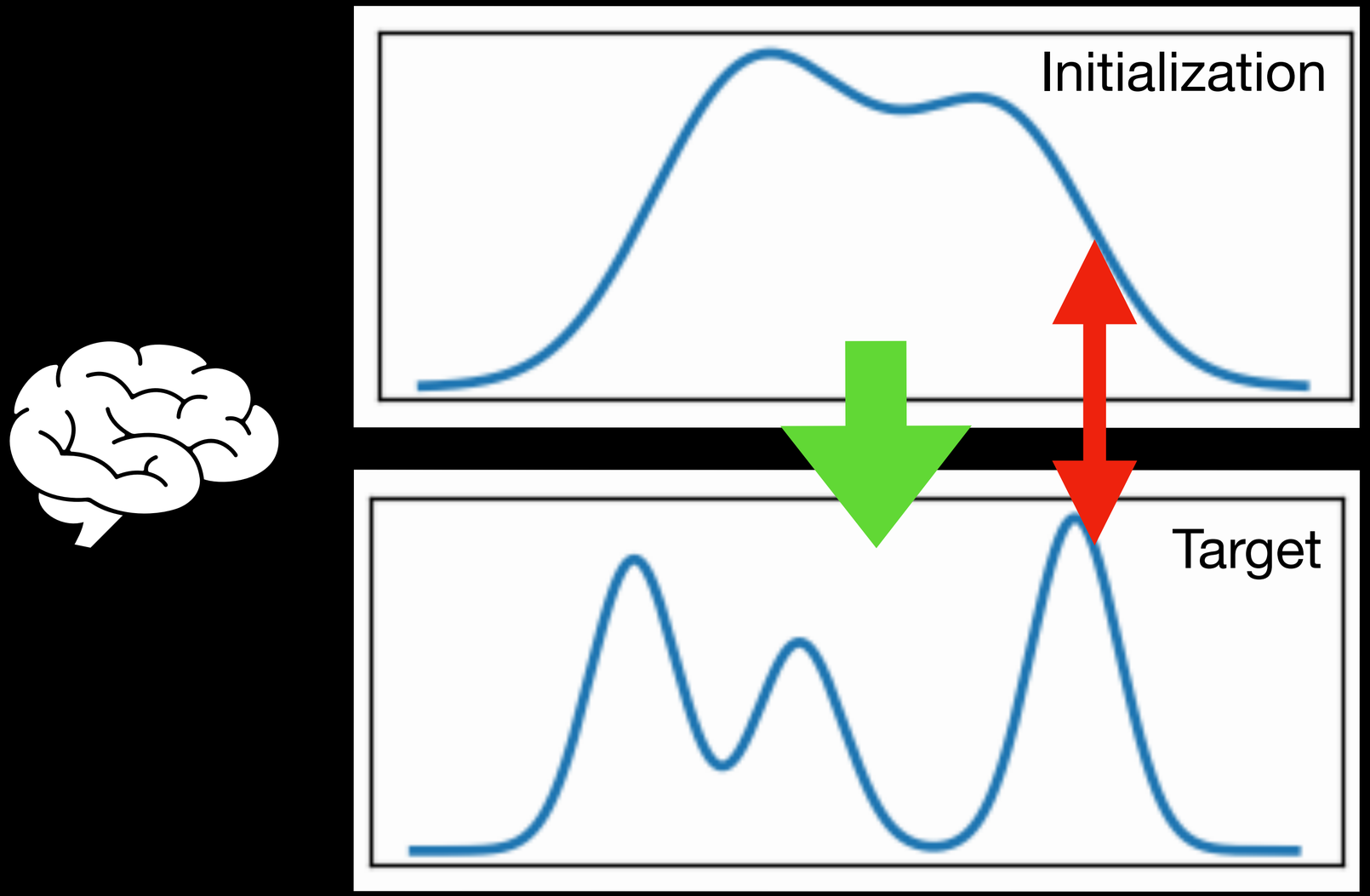
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Training a generative model




slide from https://cvpr2022-tutorial-diffusion-models.github.io/

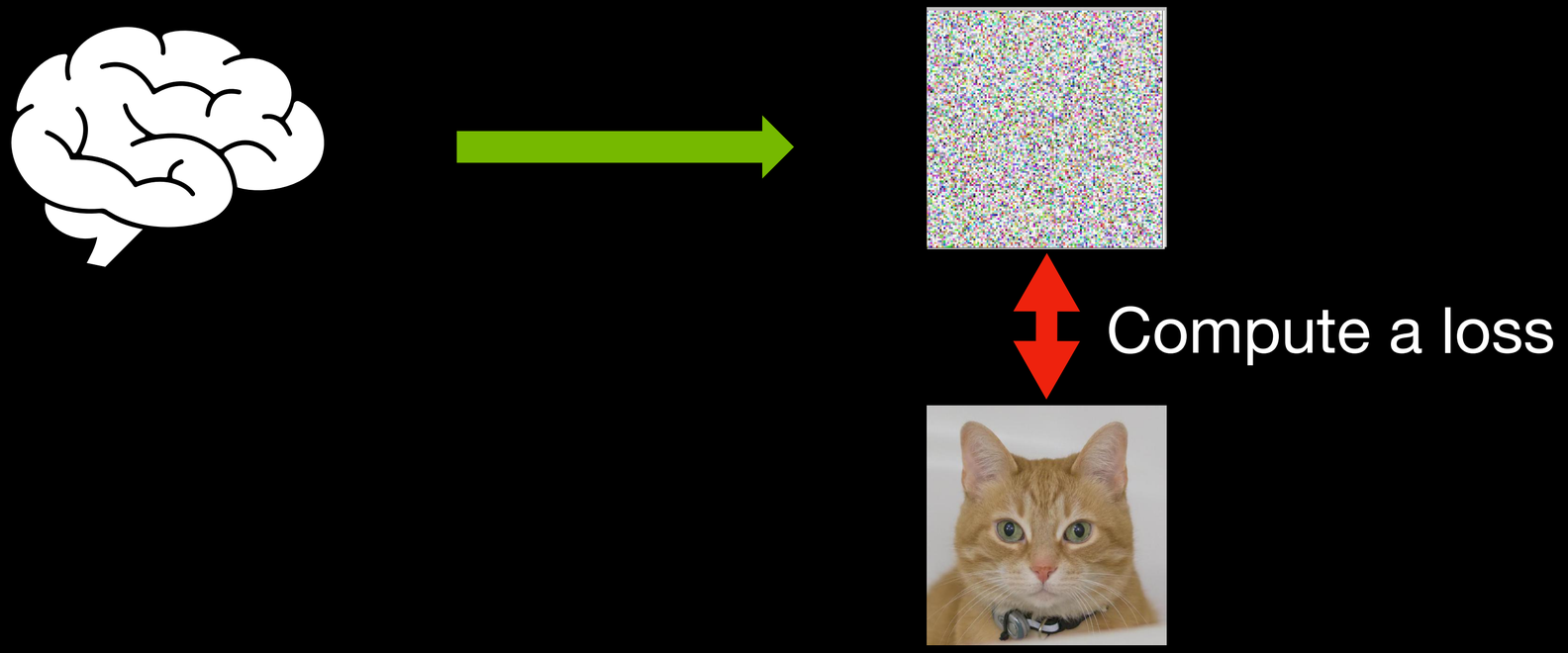
# Training a generative model




slide from https://cvpr2022-tutorial-diffusion-models.github.io/

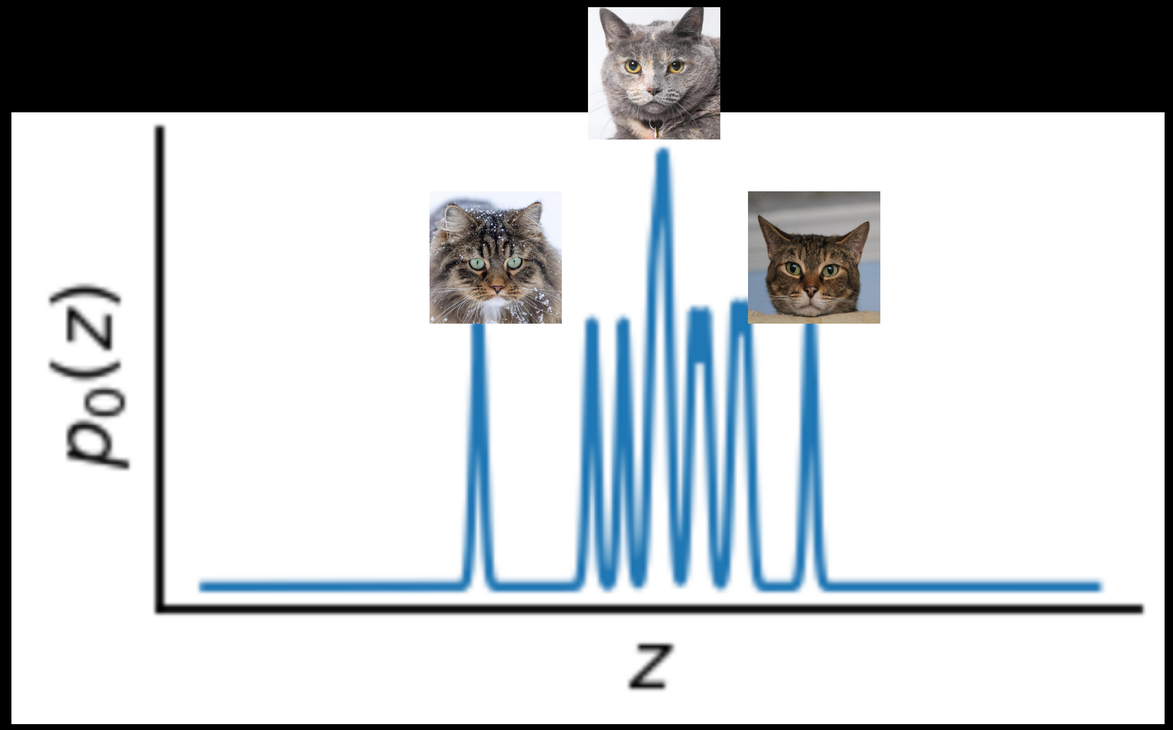
# Training a generative model

“Give me image #57838905”




slide from https://cvpr2022-tutorial-diffusion-models.github.io/

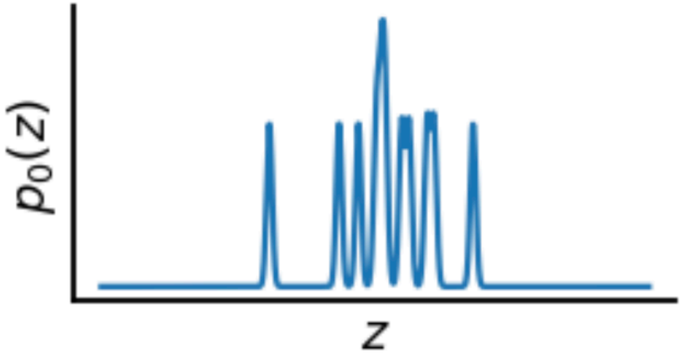
# Training a generative model

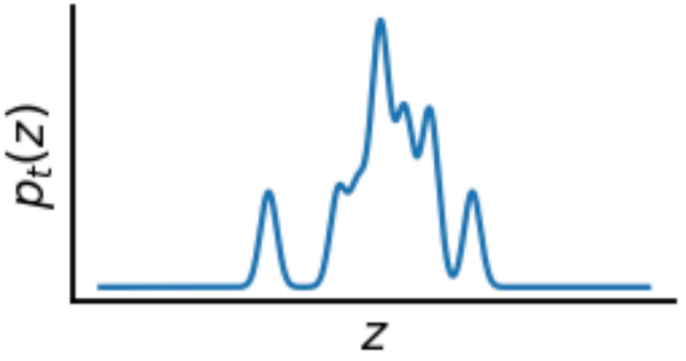


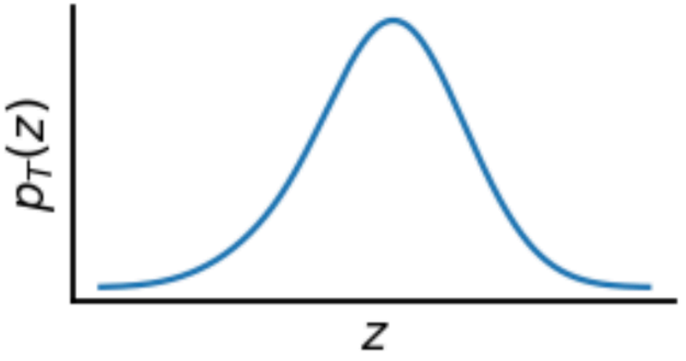

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Part 2:
# Intuition on diffusion

$$p_0(z_0), \cdots\cdots\cdots, p_t(z_t), \cdots\cdots\cdots, p_T(z_T)$$



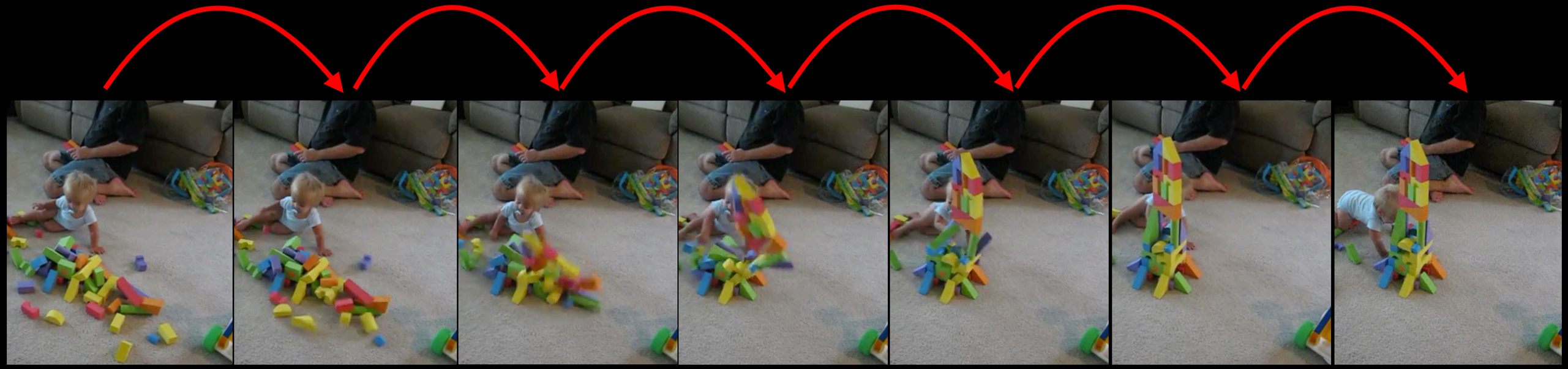



# Creating is hard!

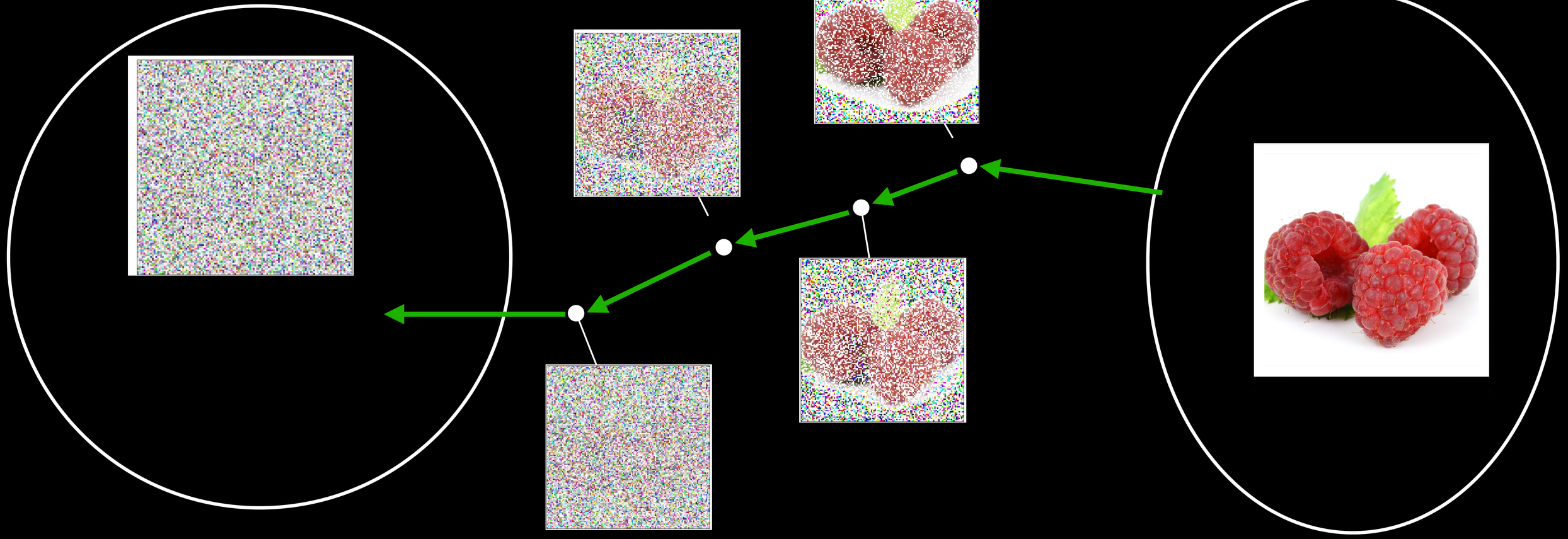
Creating is hard!

Destroying is easy!

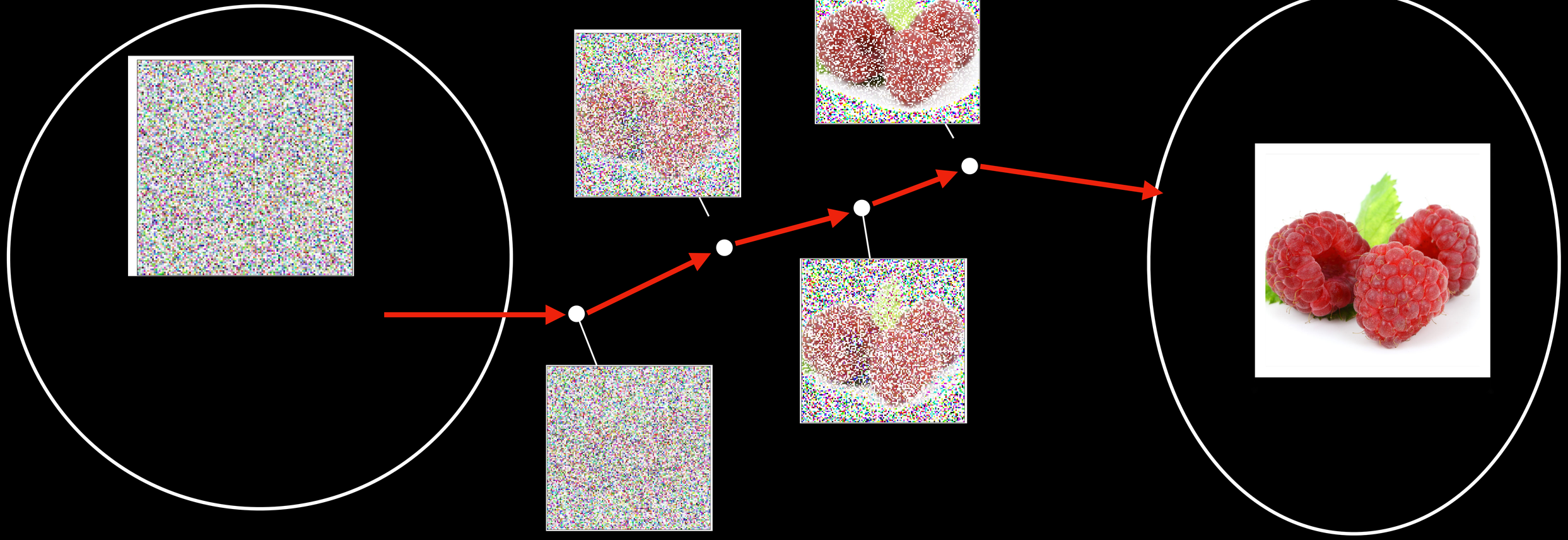
Forward destruction trajectory



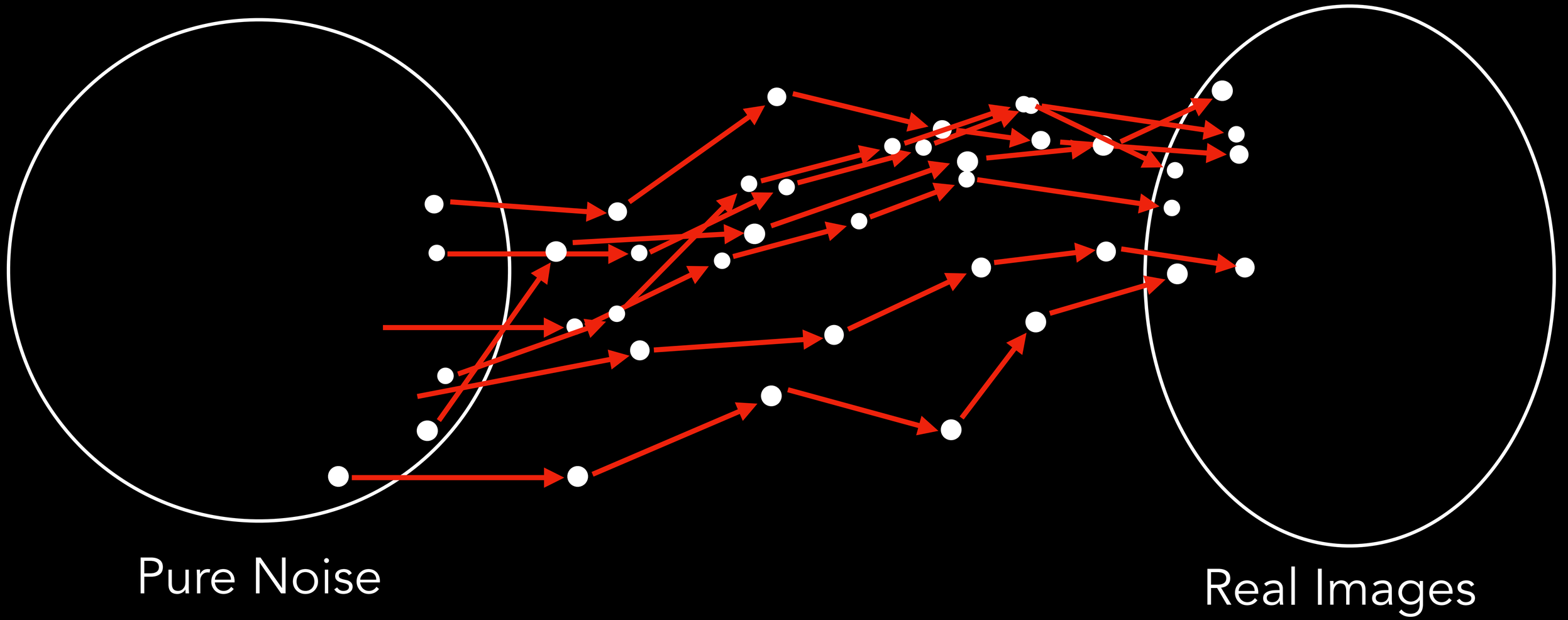
**Reverse** destruction trajectory



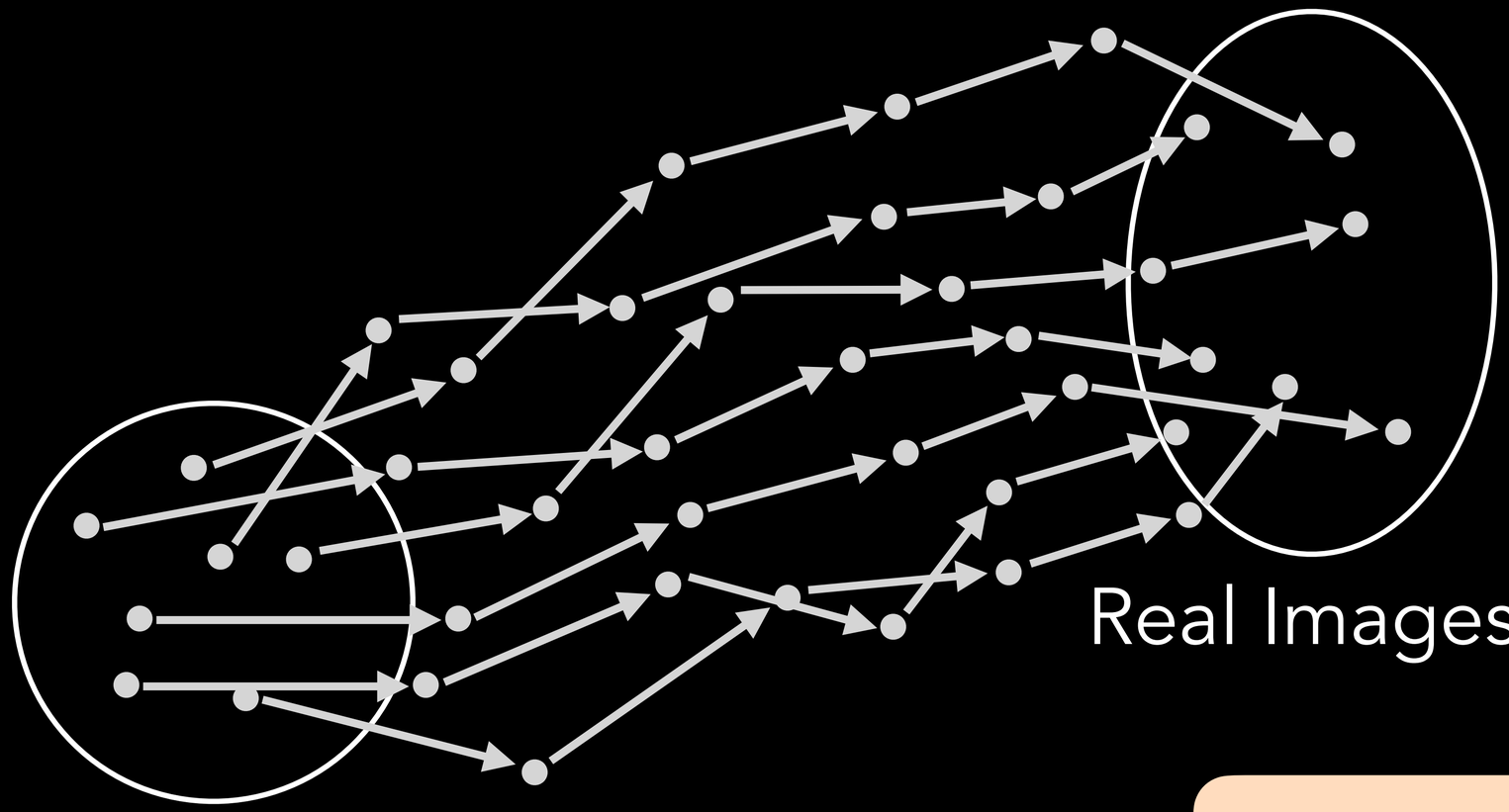

Steve Seitz, Graphics in 5 Minutes




Steve Seitz, Graphics in 5 Minutes

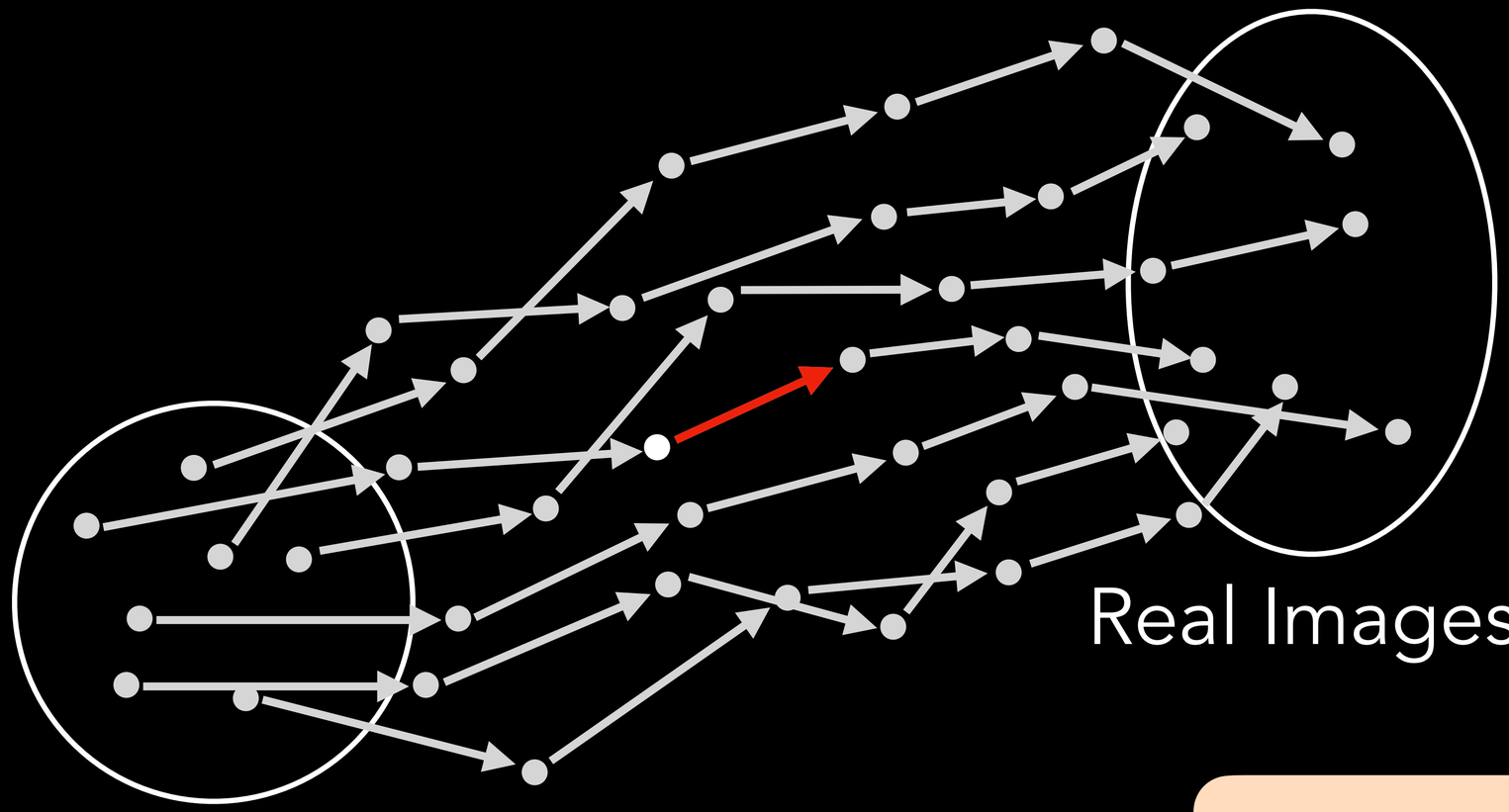



Steve Seitz, Graphics in 5 Minutes



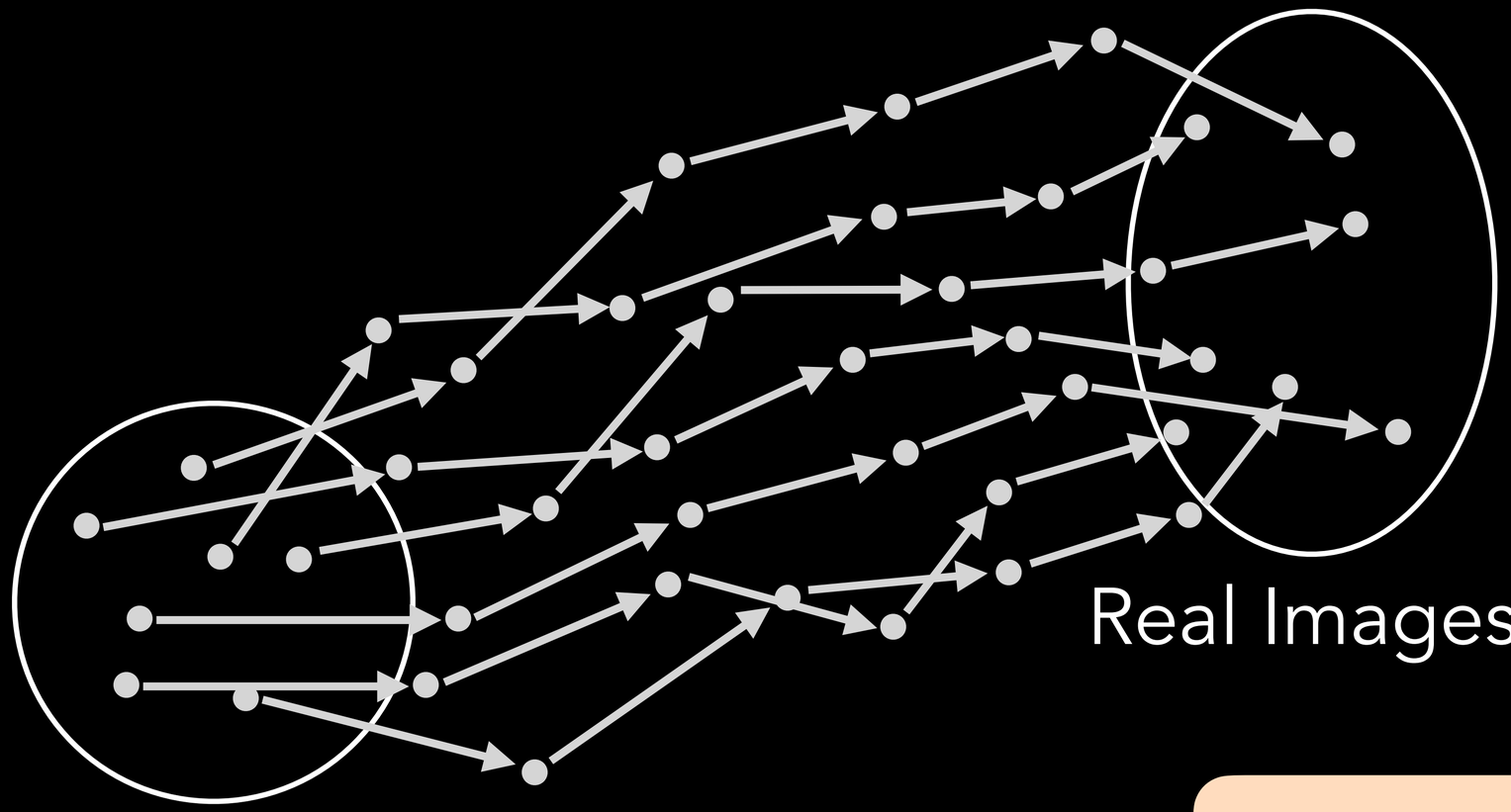

**diffusion neural network**

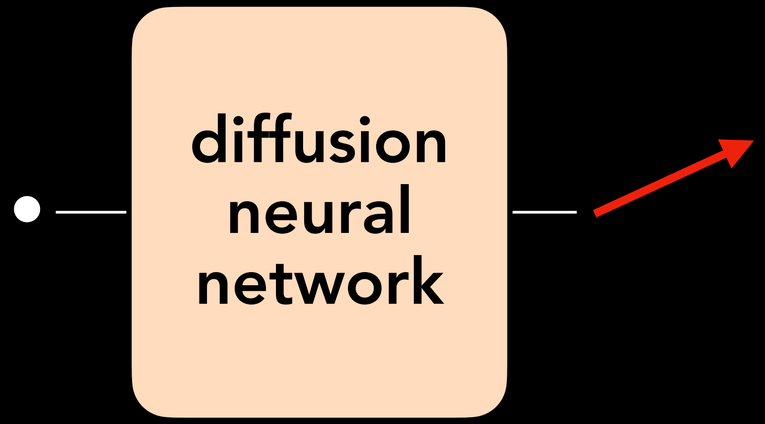
Steve Seitz, Graphics in 5 Minutes



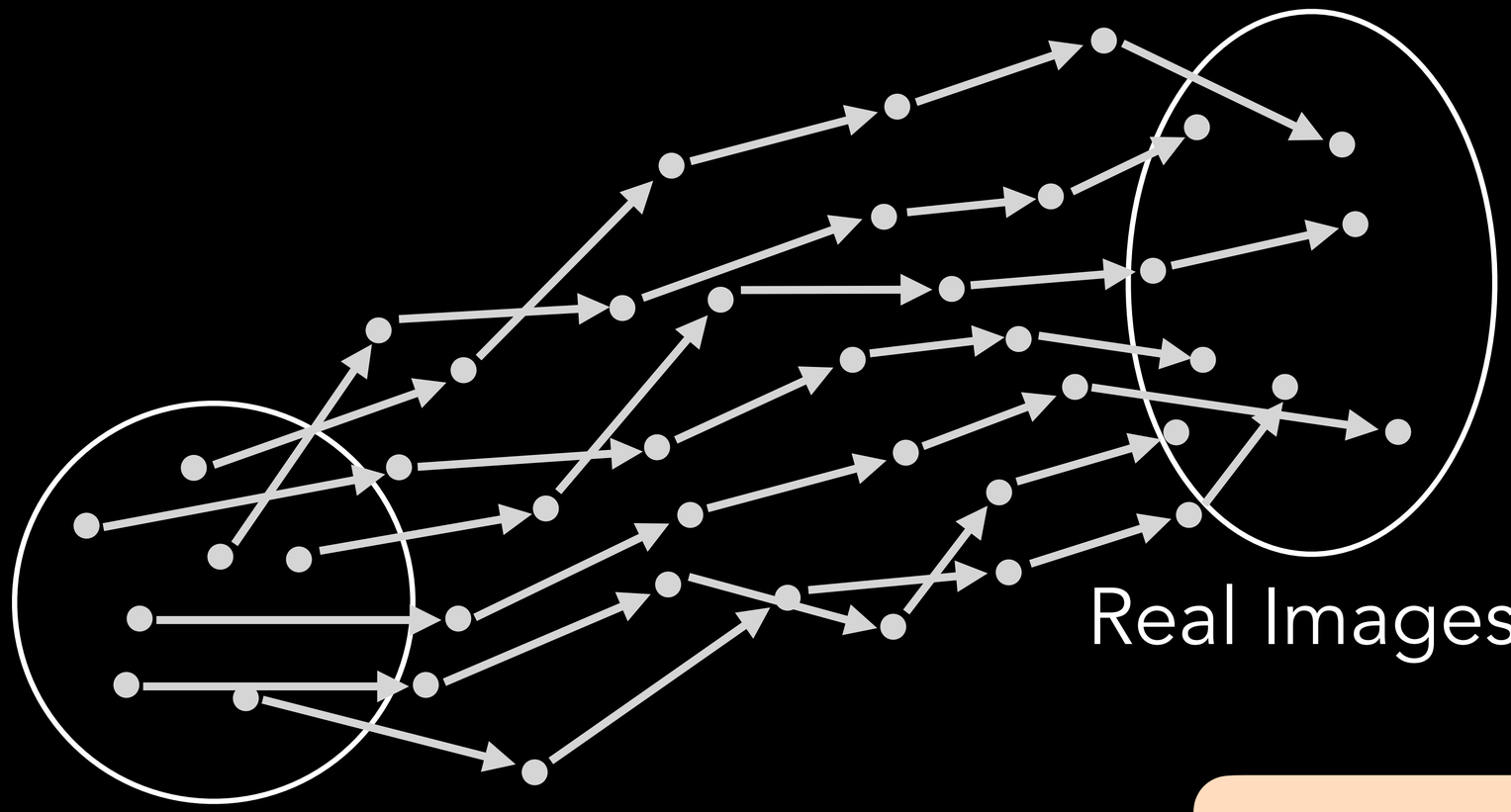

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes



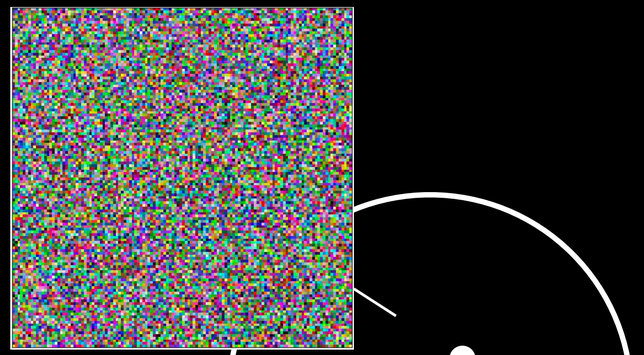

Steve Seitz, Graphics in 5 Minutes



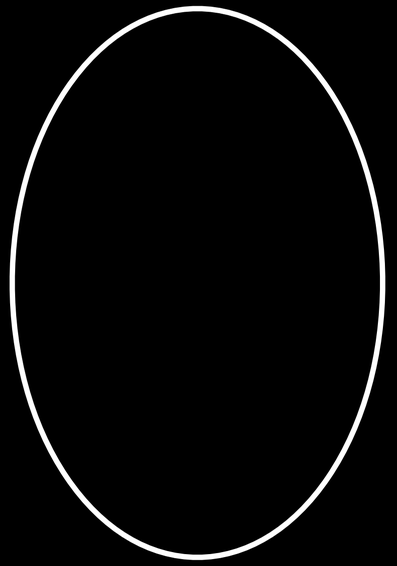

“raspberry”

diffusion neural network

Steve Seitz, Graphics in 5 Minutes



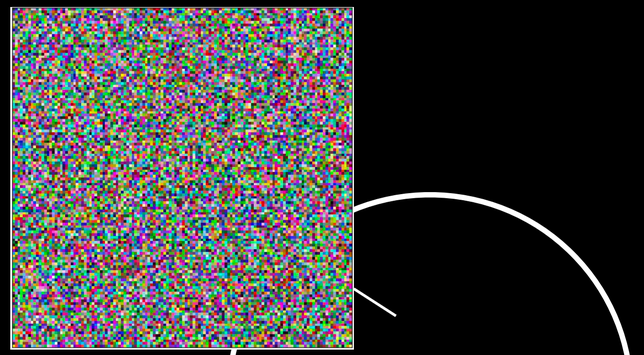
Pure Noise



Real Images

“raspberry”

diffusion neural network

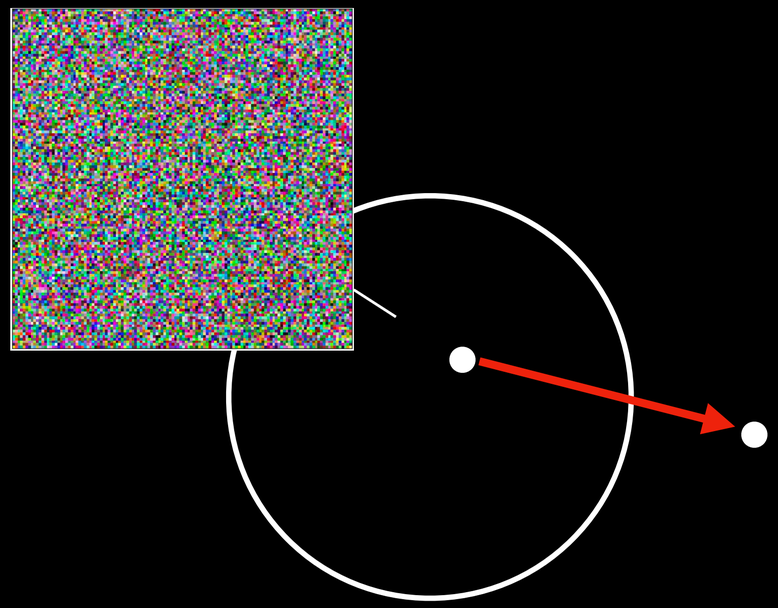


Pure Noise

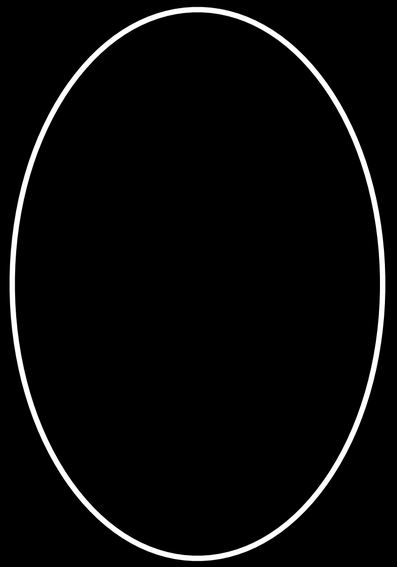
Real Images

“raspberry”

diffusion neural network



Pure Noise

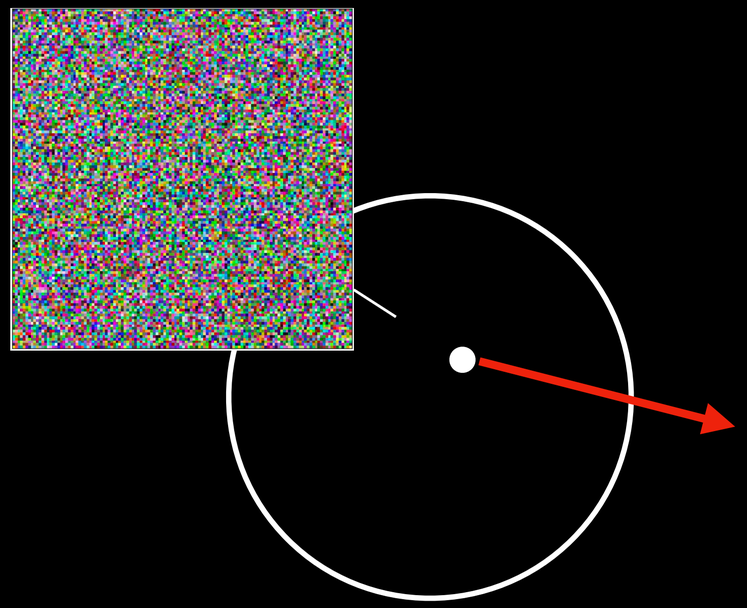


Real Images

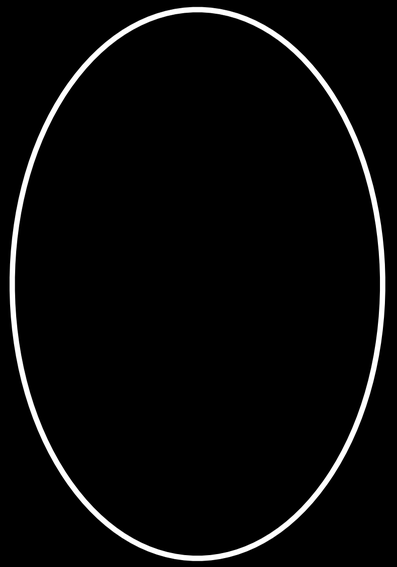
**“raspberry”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes



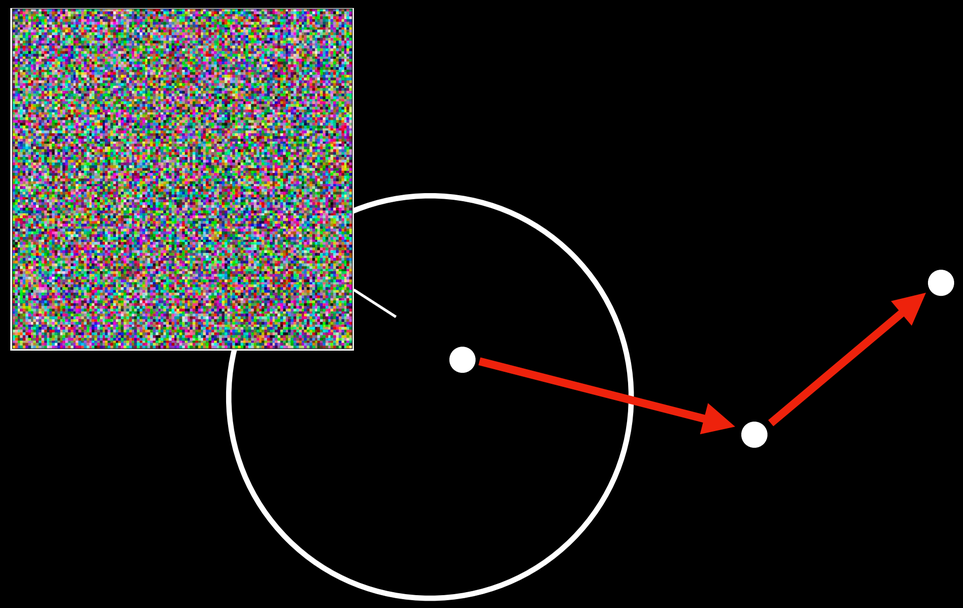
Pure Noise



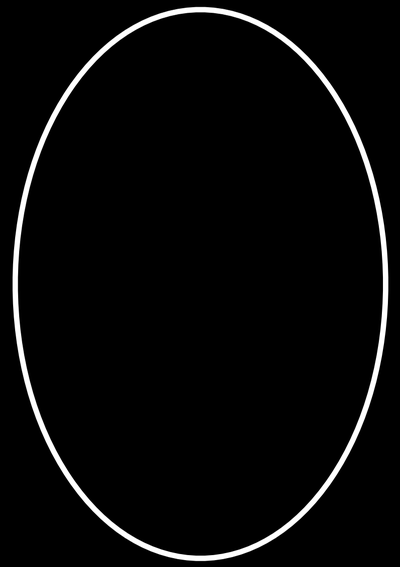
Real Images

“raspberry”

diffusion neural network



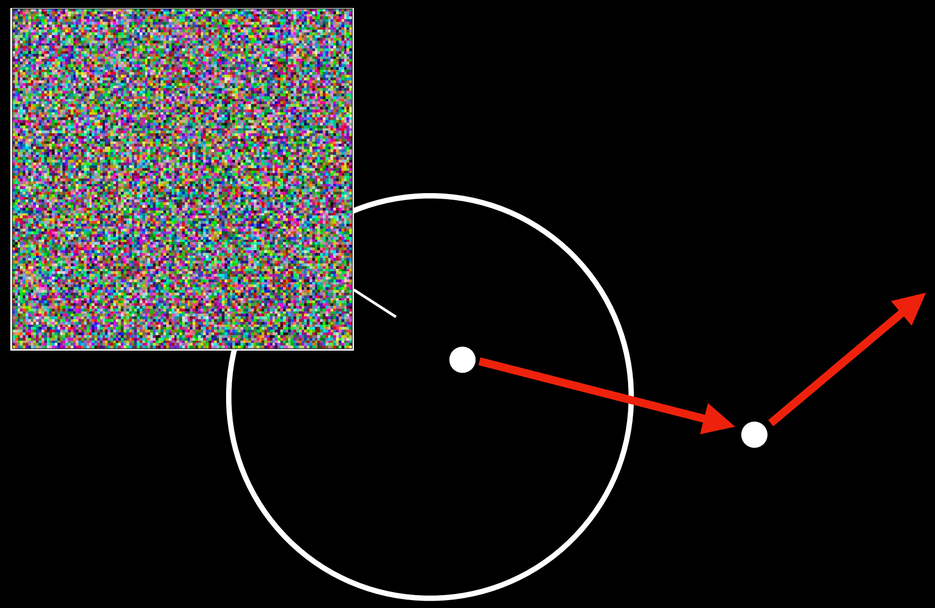
Pure Noise



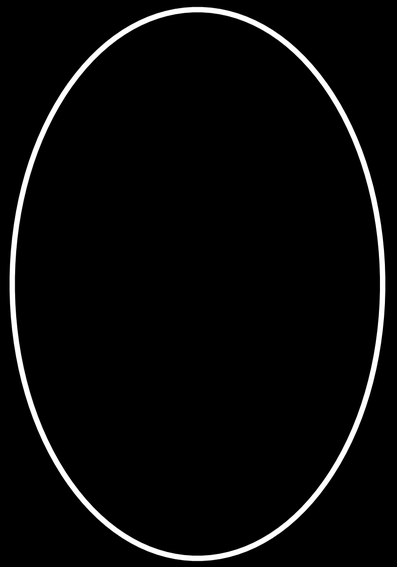
Real Images

**“raspberry”**

**diffusion neural network**



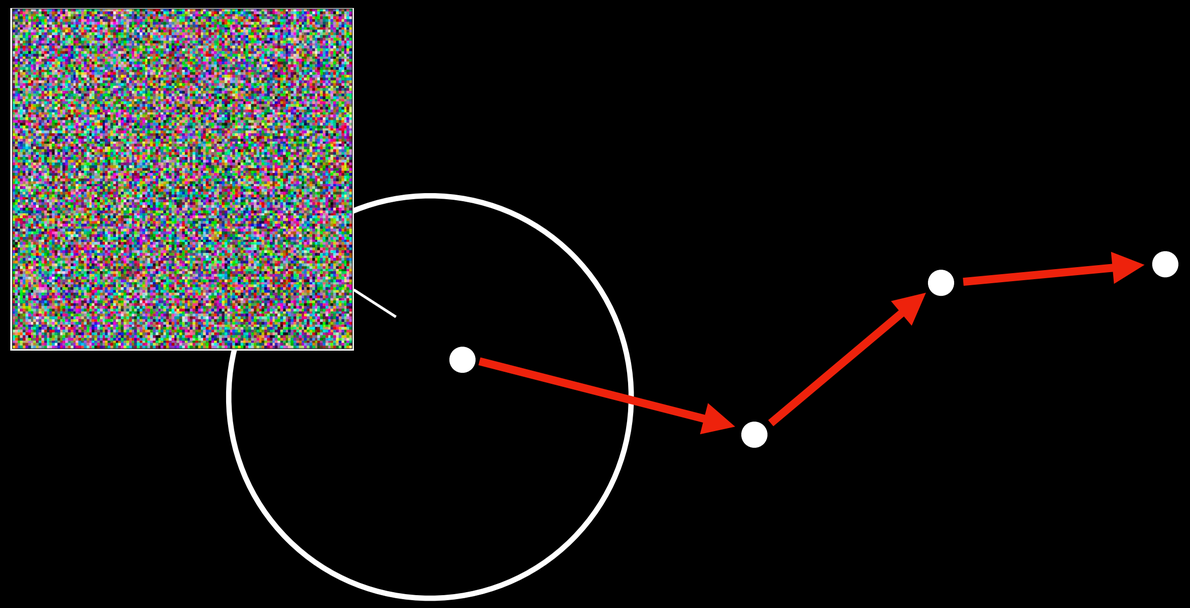
Pure Noise



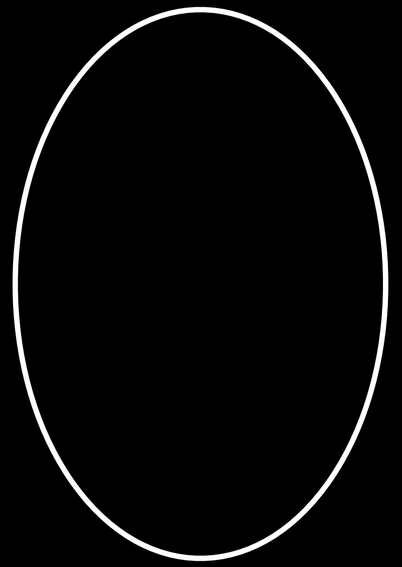
Real Images

“raspberry”

diffusion neural network



Pure Noise

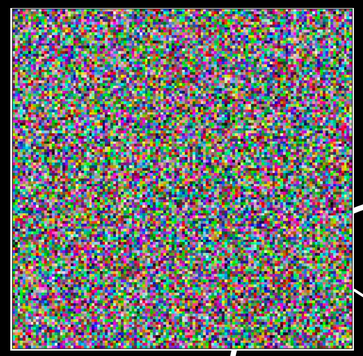


Real Images

**“raspberry”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes



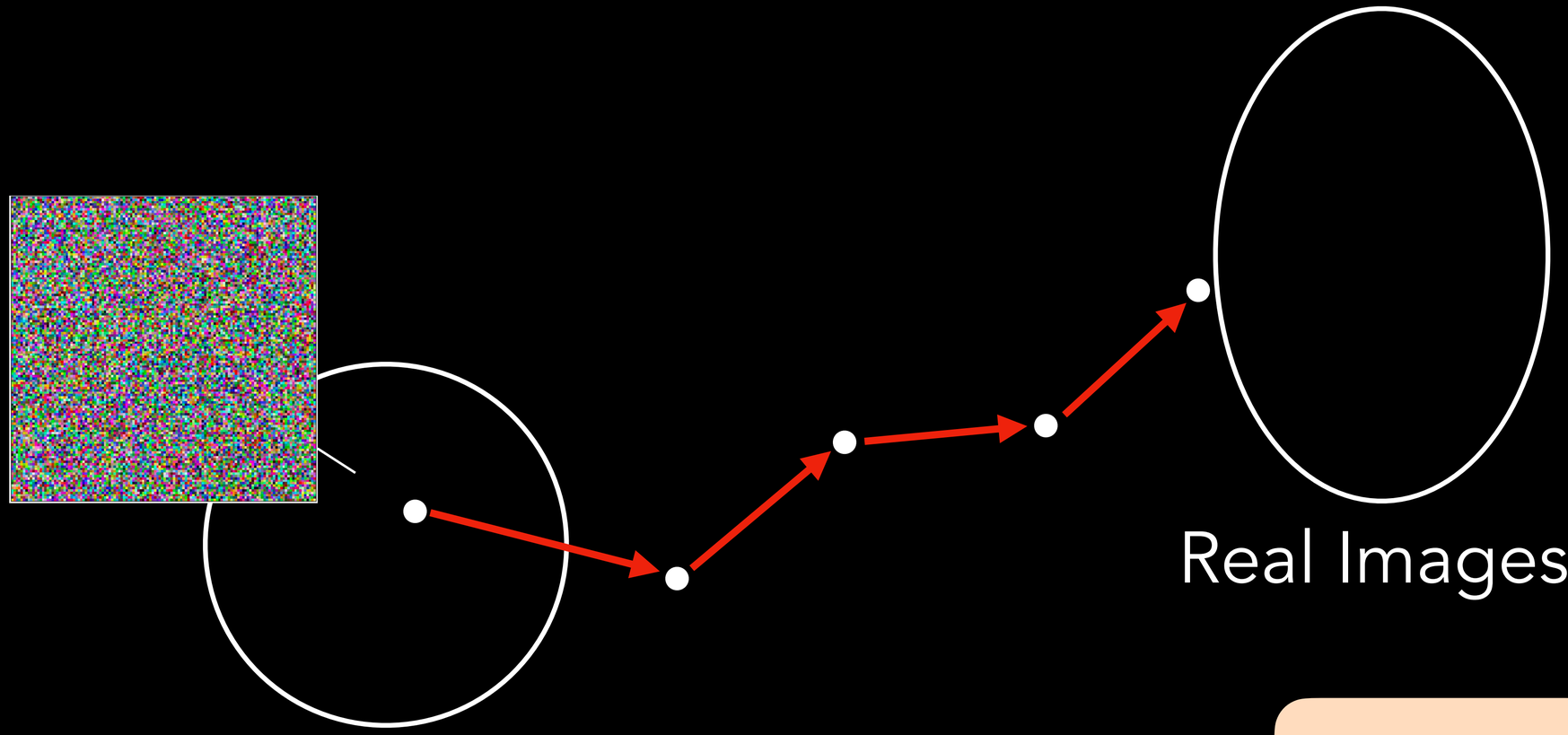
Pure Noise

Real Images

“raspberry”

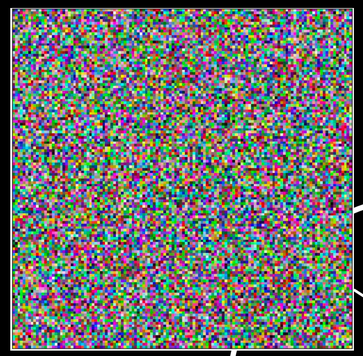
diffusion neural network

Steve Seitz, Graphics in 5 Minutes




**“raspberry”** → **diffusion neural network**

Steve Seitz, Graphics in 5 Minutes



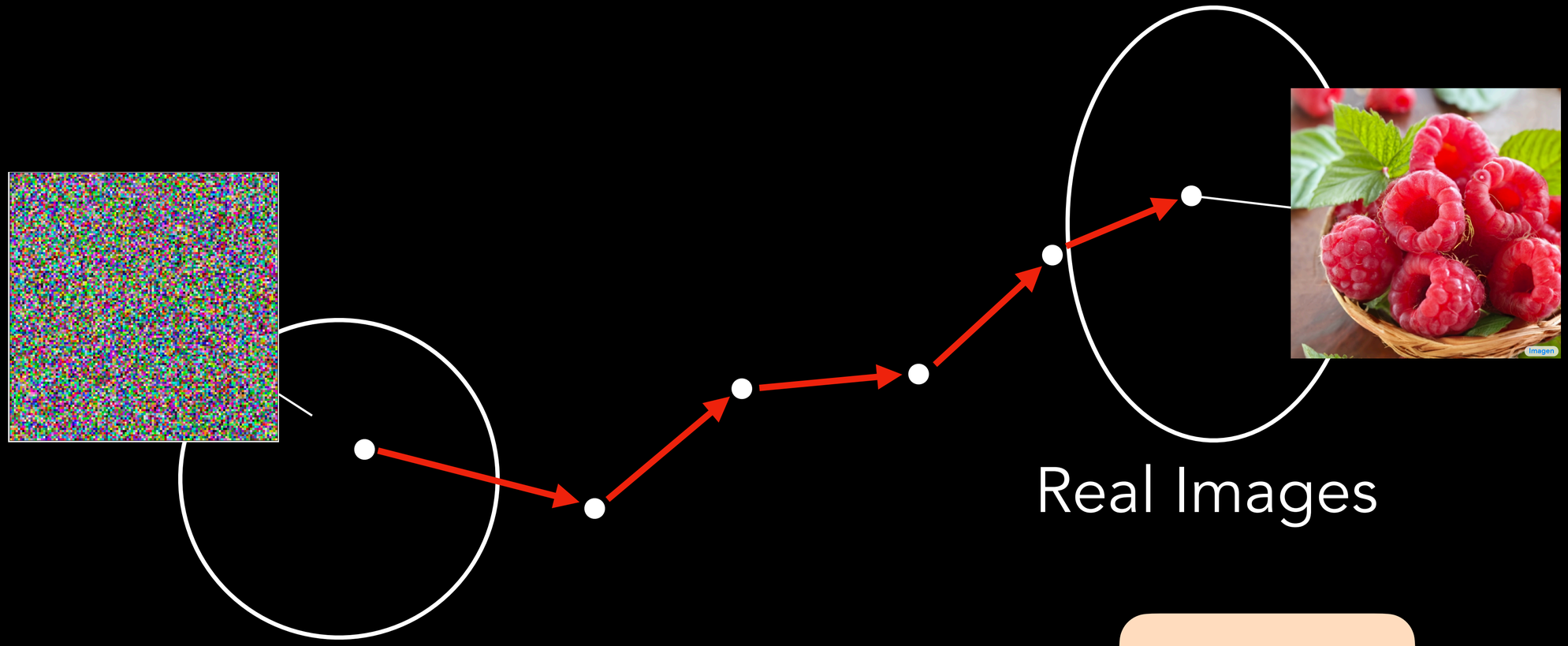
Pure Noise

Real Images

“raspberry”

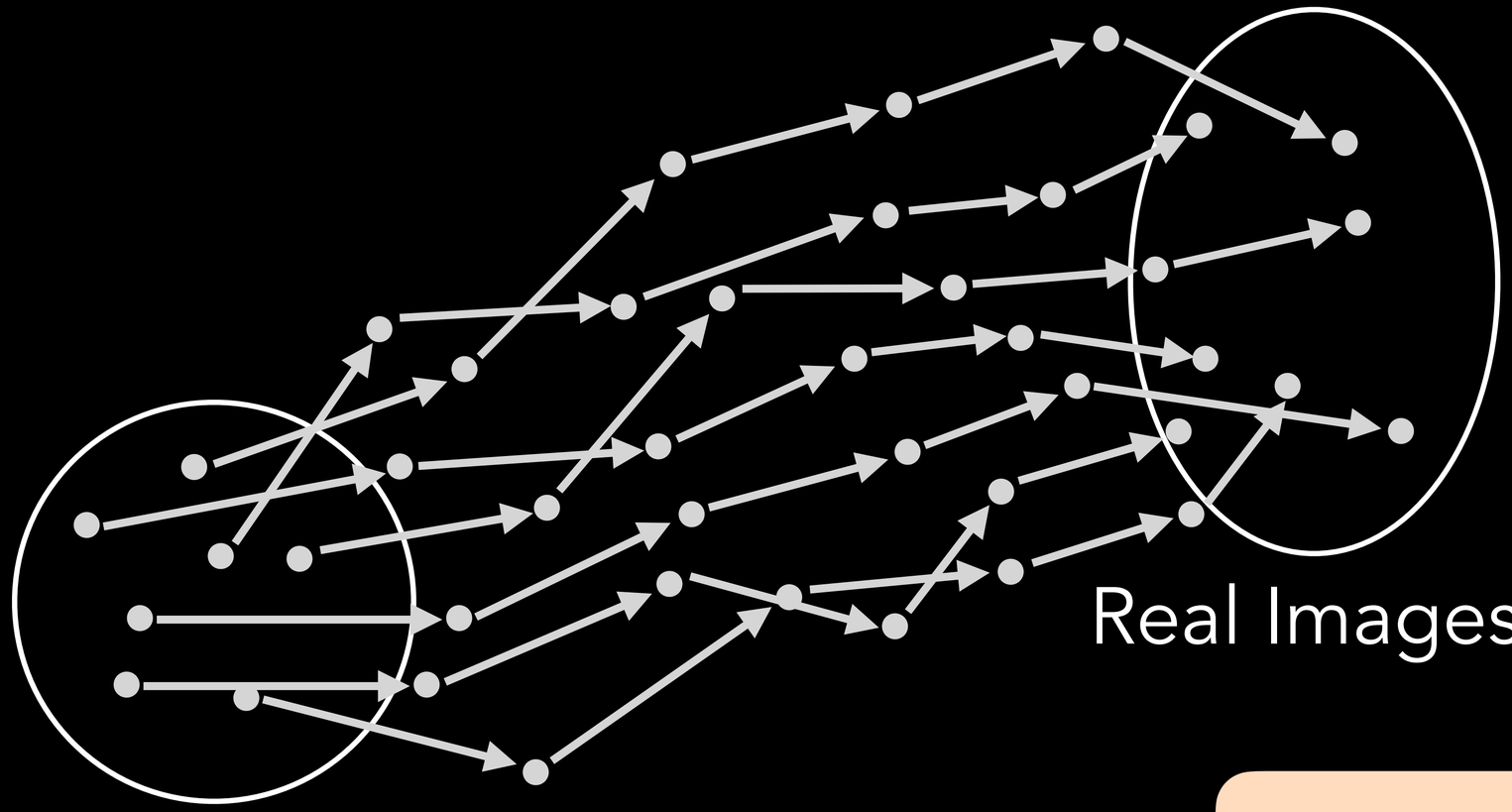
diffusion neural network

Steve Seitz, Graphics in 5 Minutes




Steve Seitz, Graphics in 5 Minutes

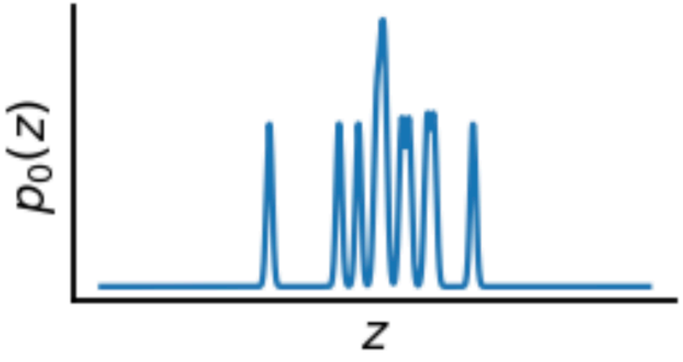
# Learns more than just the training images!

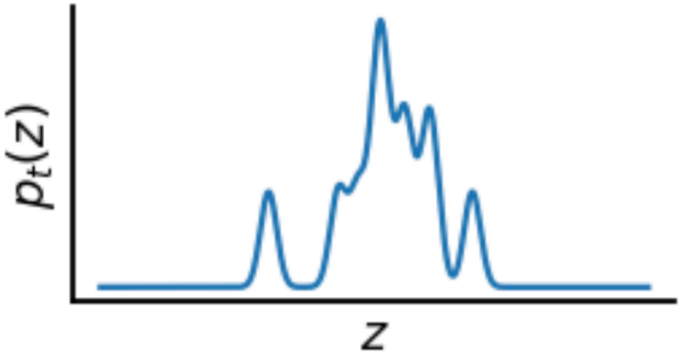


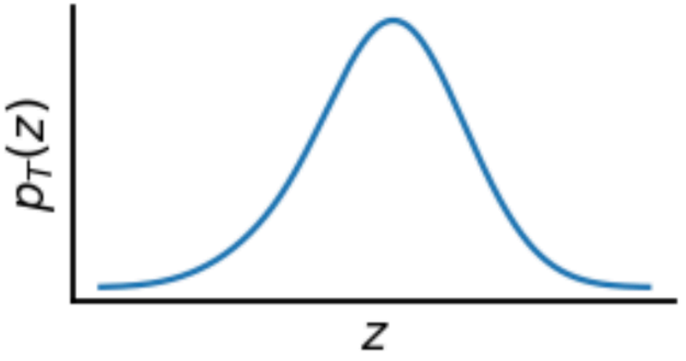

diffusion neural network

Steve Seitz, Graphics in 5 Minutes

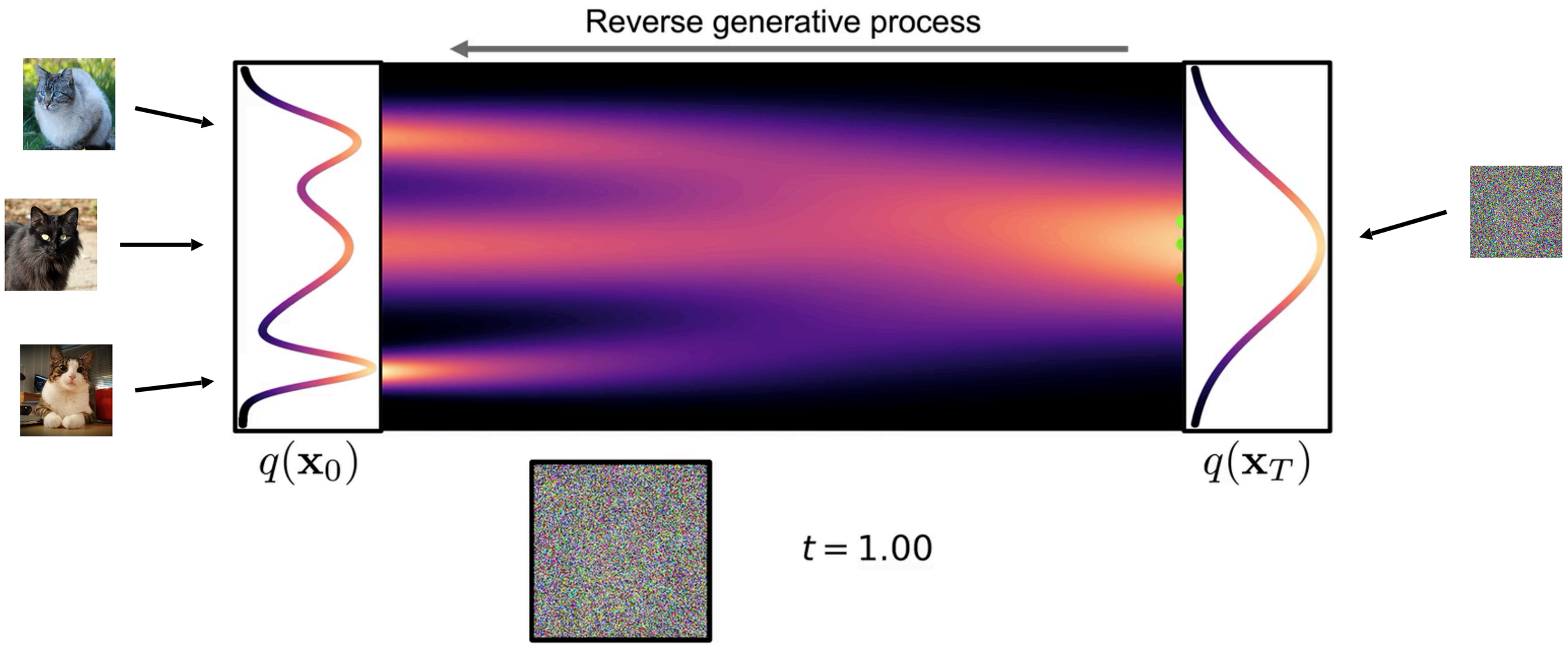
$$p_0(z_0), \cdots\cdots\cdots, p_t(z_t), \cdots\cdots\cdots, p_T(z_T)$$



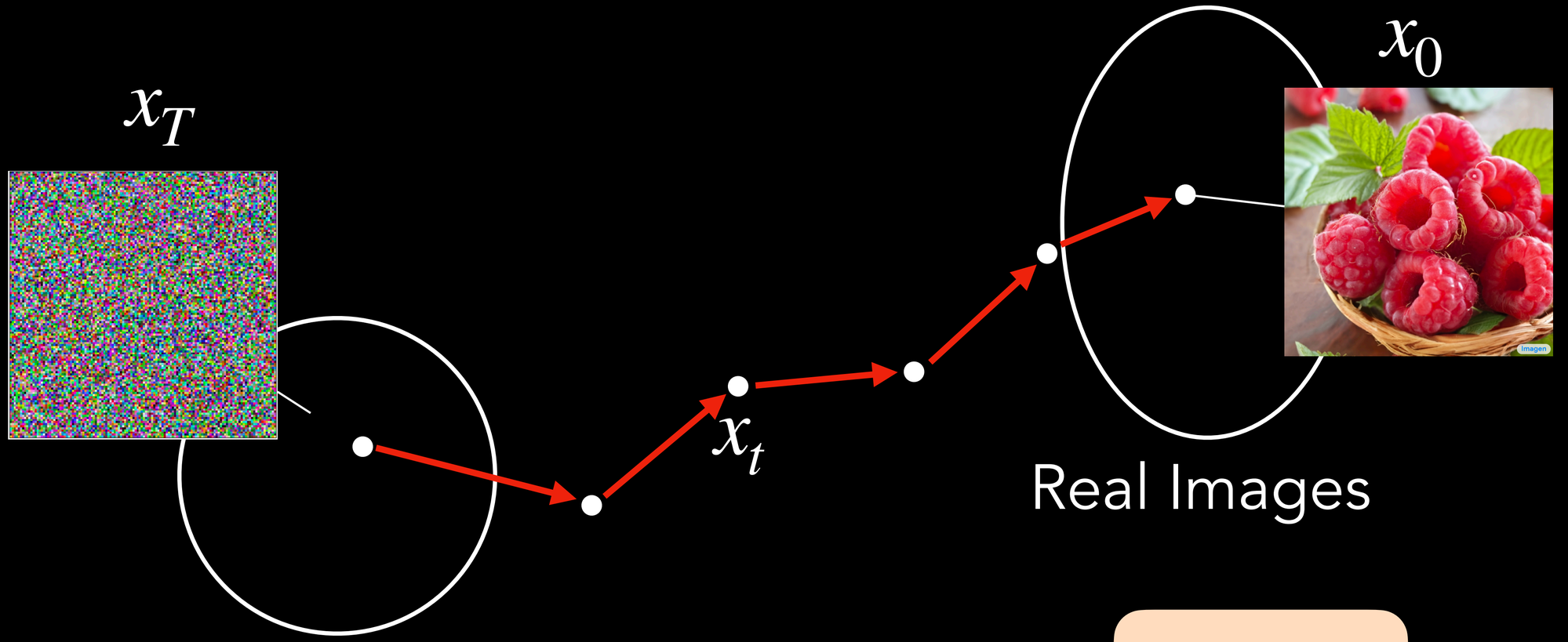
# The Generative Reverse Stochastic Differential Equation




slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Part 3:
# How to train a diffusion model

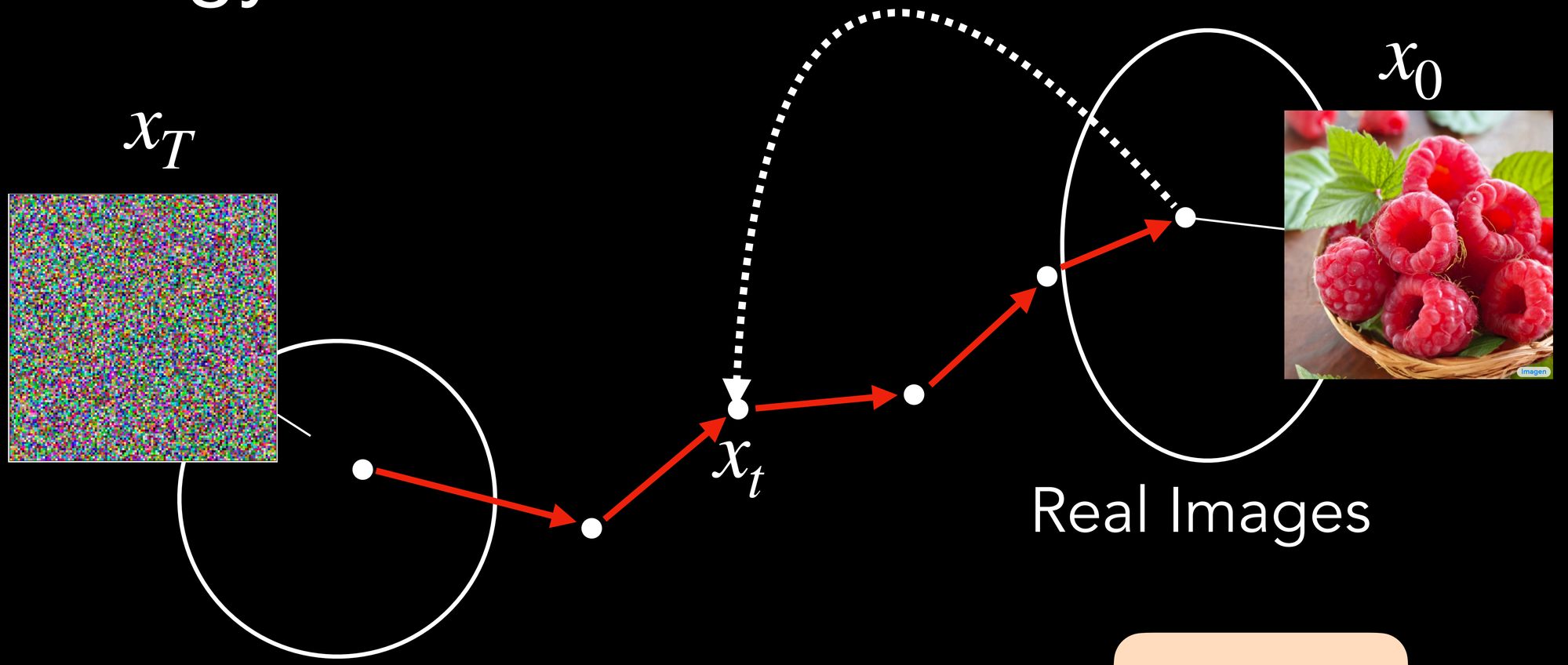
# Terminology




**“raspberry”** → **diffusion neural network**

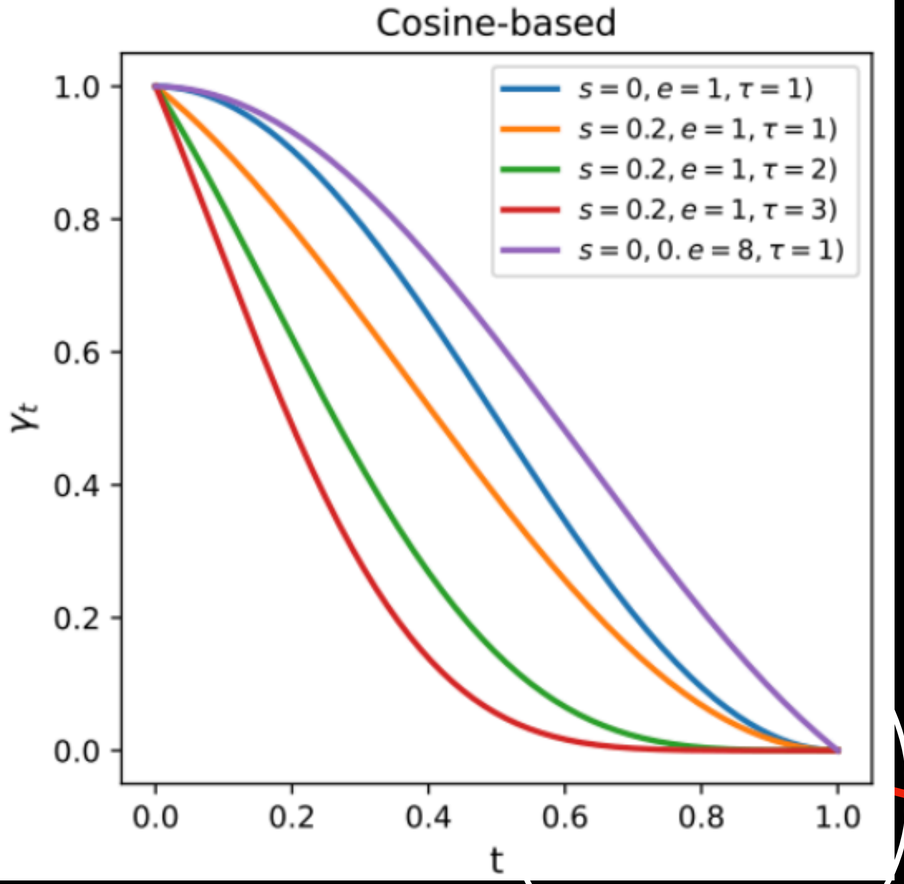
# Terminology

$$\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$




**“raspberry”** → **diffusion neural network**

Steve Seitz, Graphics in 5 Minutes




$$\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

$x_0$

$x_t$

Real Images

Pure Noise

**“raspberry”** — **diffusion neural network** —

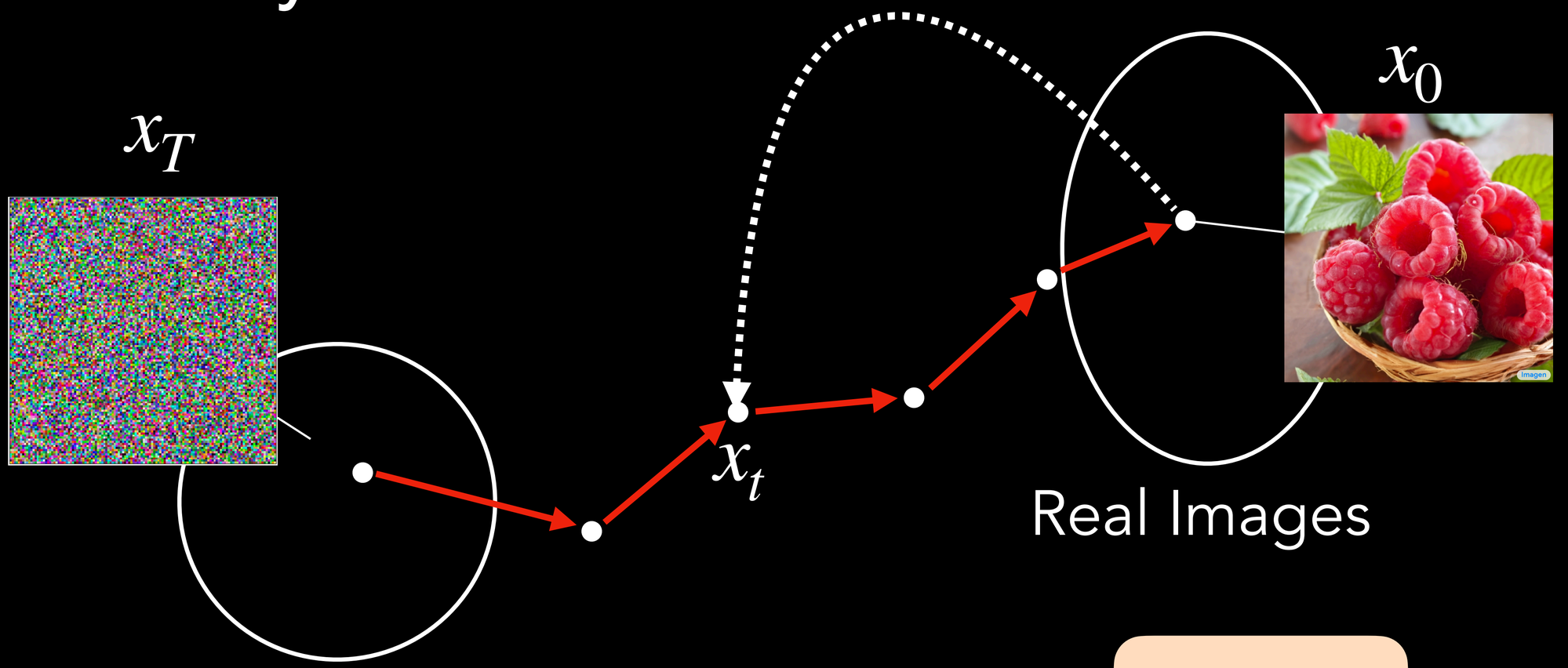
Steve Seitz, Graphics in 5 Minutes

training algorithm

```
repeat

   x0 := sample from dataset
   t  := sample from uniform distribution [1:T]
   epsilon := sample from standardized normal

   take gradient descent step

until converged
```

$$\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|$$

# More visually…




**“raspberry”** — **diffusion neural network** —

# What’s important?

Architecture

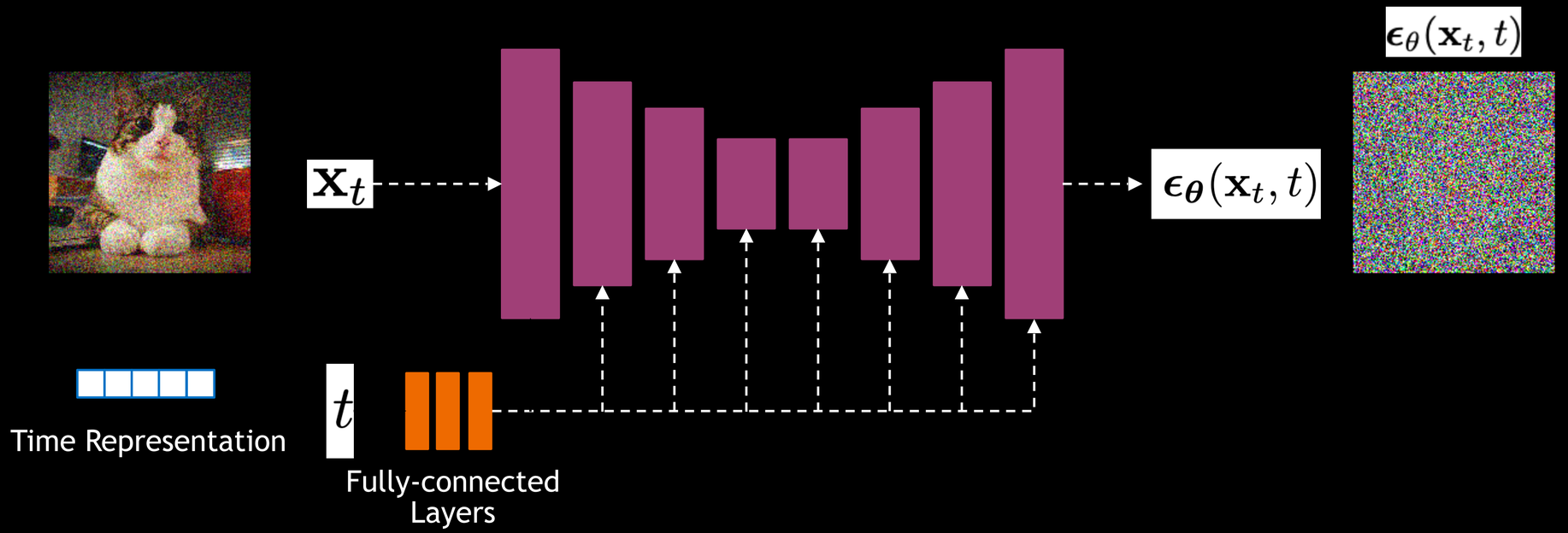
Capacity

Data

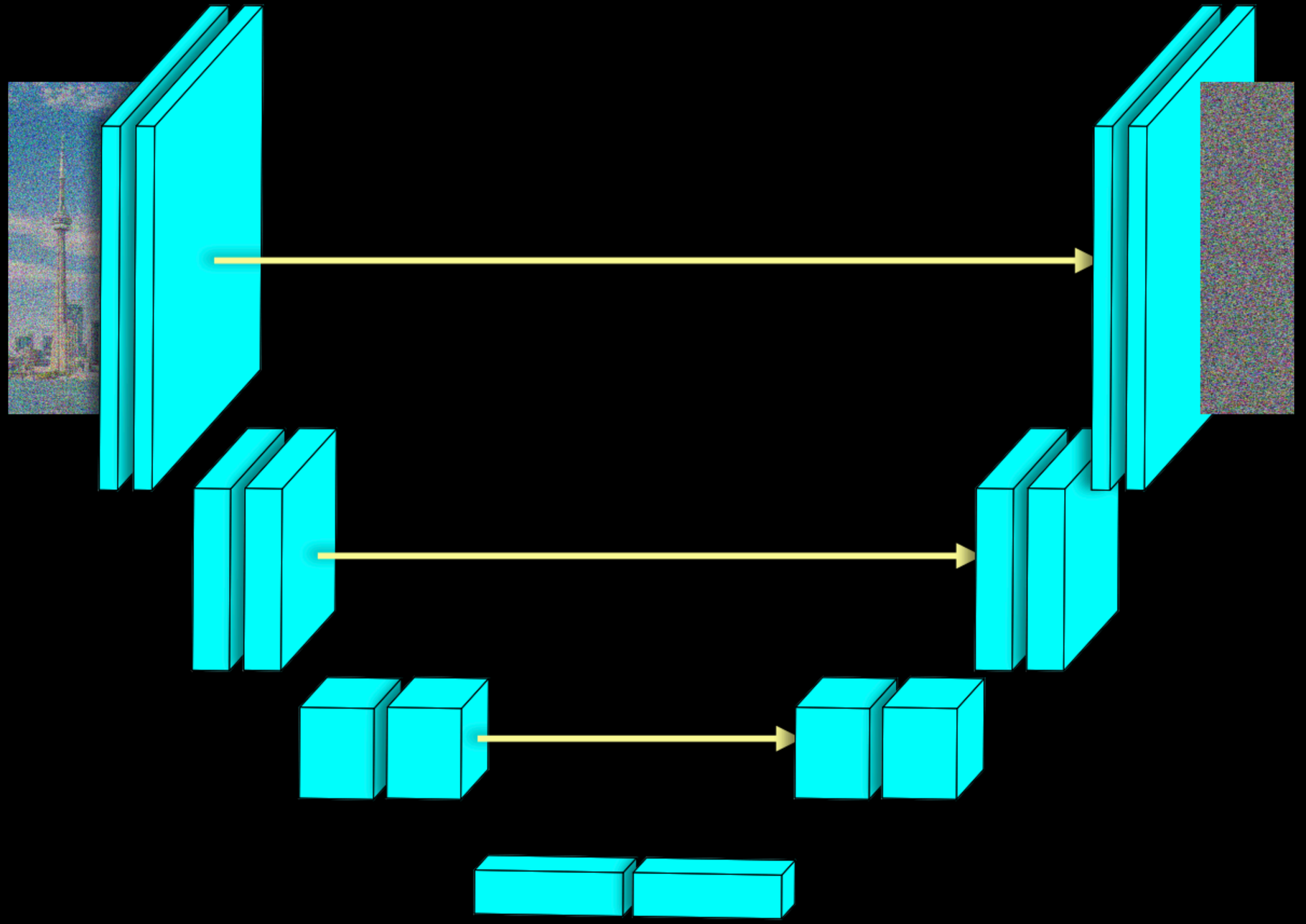
Noise schedule

Conditioning

# Architecture



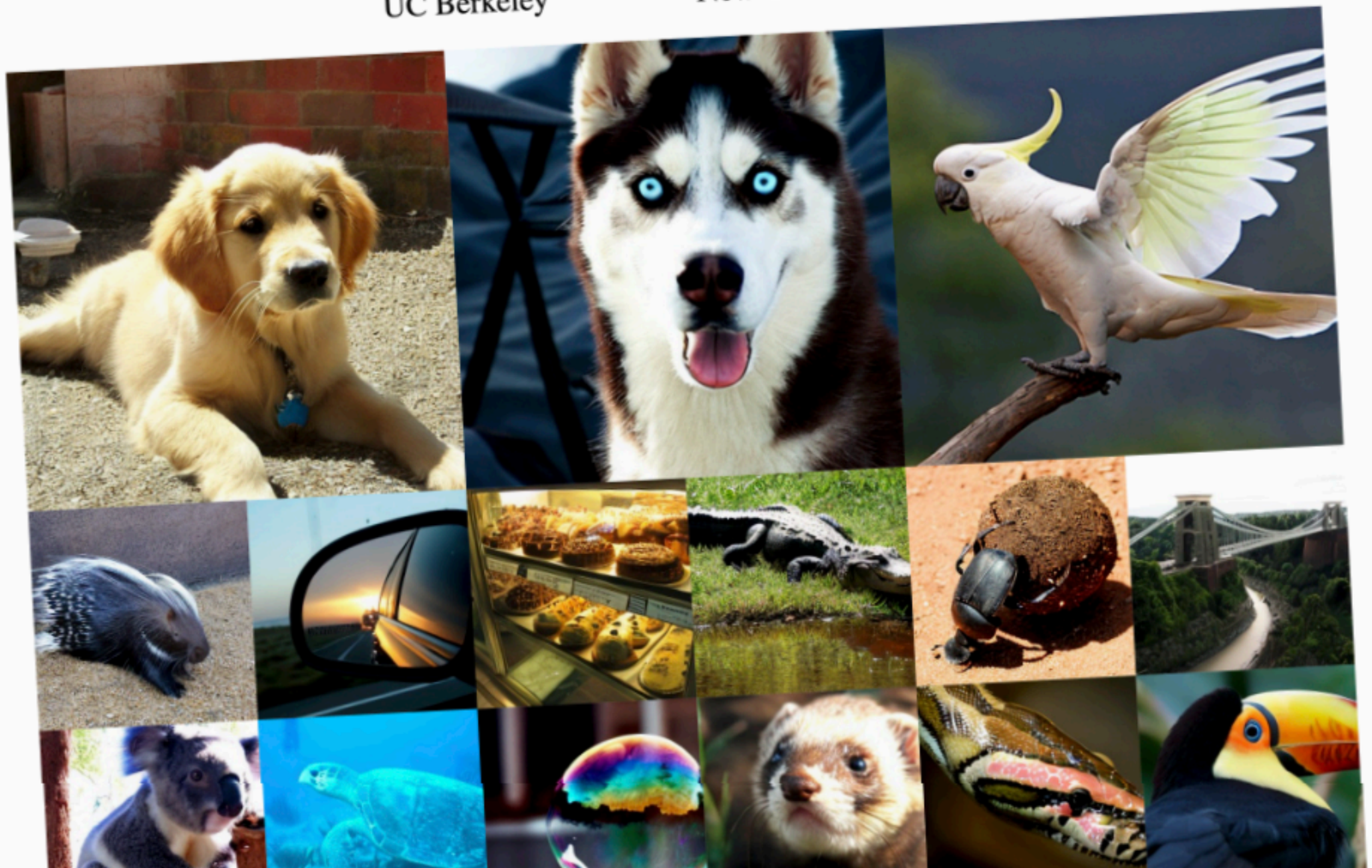

slide from https://cvpr2022-tutorial-diffusion-models.github.io/



# Scalable Diffusion Models with Transformers

William Peebles*
UC Berkeley

Saining Xie
New York University



# Latent models

Encode

Latent Image

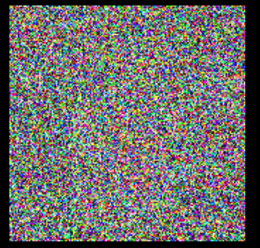
$\mathbf{x}_t$

$t$

Time Representation

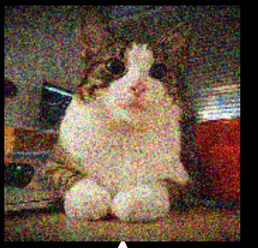
Fully-connected

$\epsilon_\theta(\mathbf{x}_t, t)$

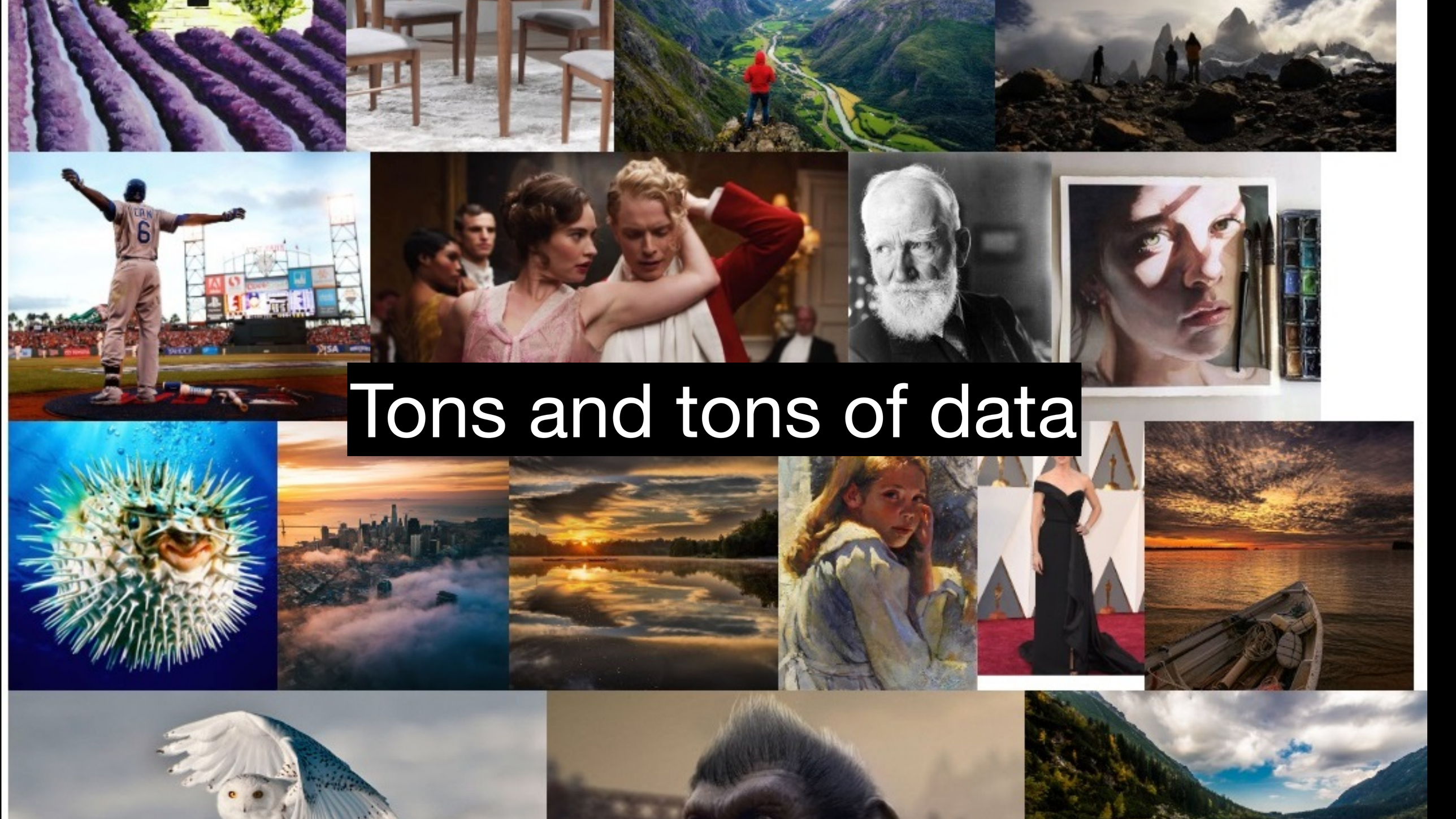


$\epsilon_\theta(\mathbf{x}_t, t)$

Decode

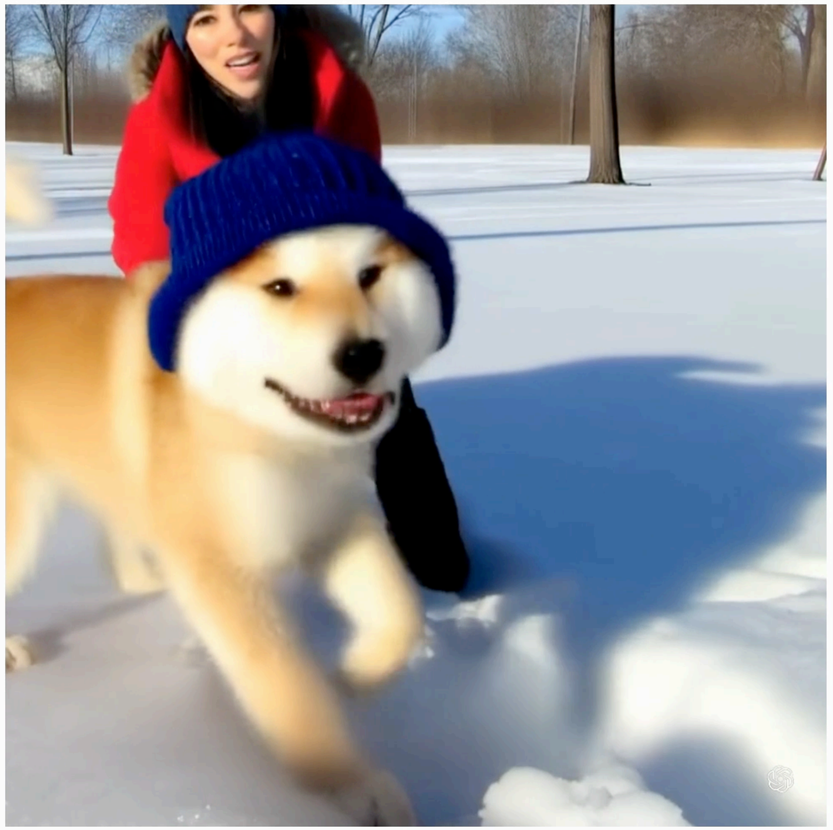


Latent Image


Tons and tons of data

# Bigger is better

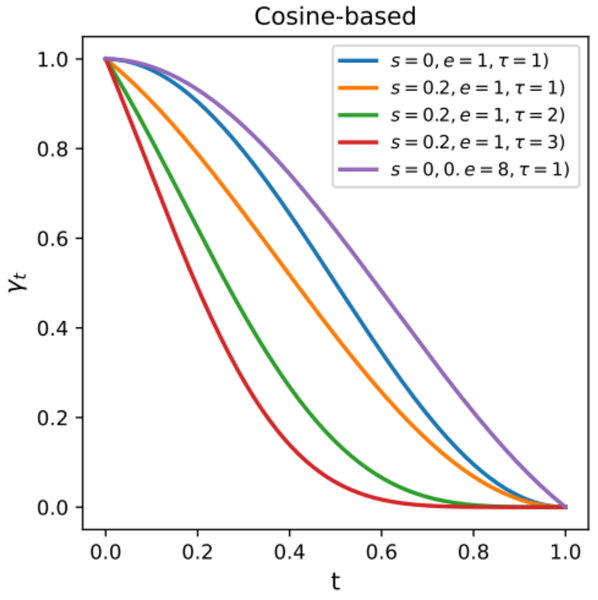
Base compute



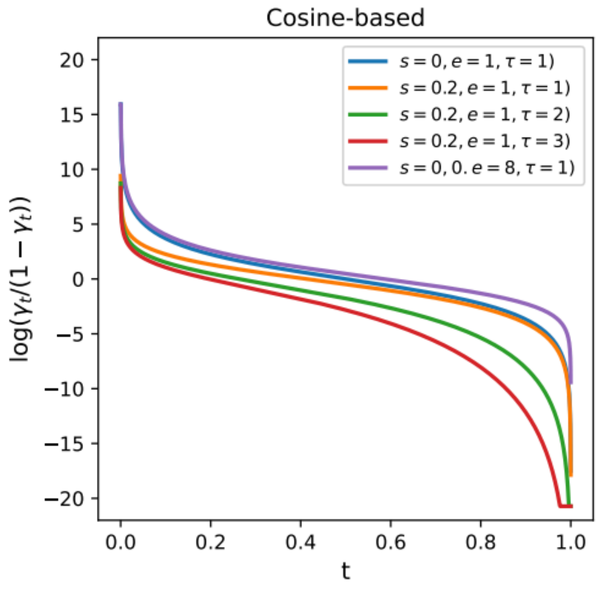
4x compute

32x compute

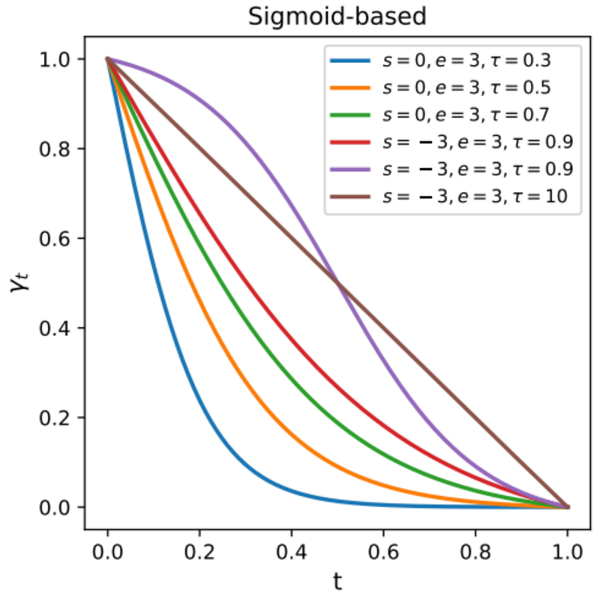
# Noise schedules!



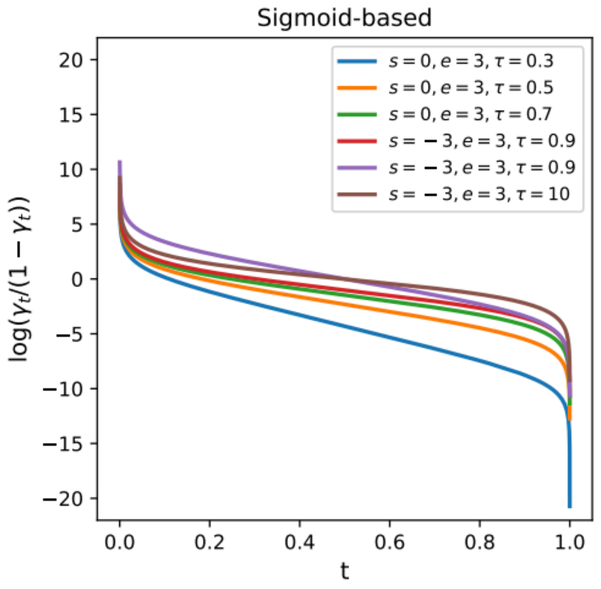

(a) Cosine




(b) Cosine (logSNR)




(c) Sigmoid




(d) Sigmoid (logSNR)

# Noise schedules!

(a) $64 \times 64$ (b) $128 \times 128$ (c) $256 \times 256$ (d) $512 \times 512$ (e) $1024 \times 1024$

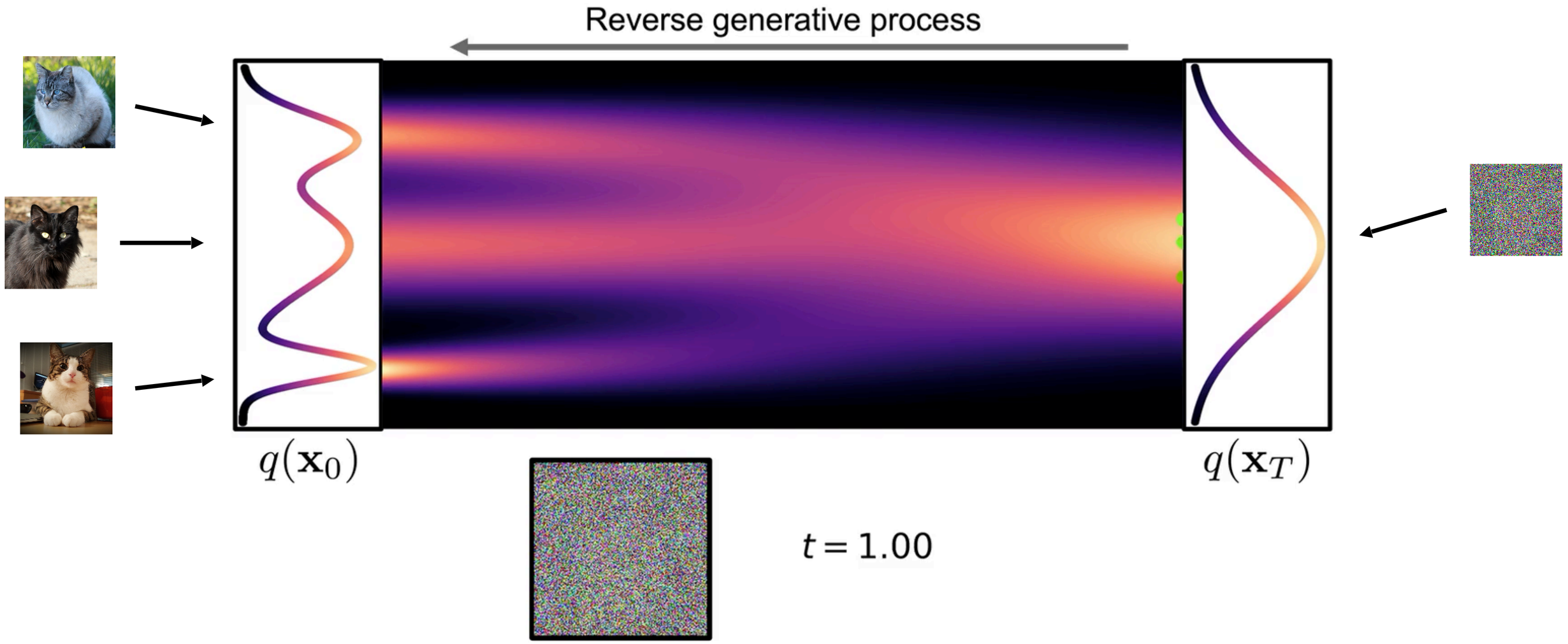
Figure 2: Noised images ($\boldsymbol{x}_t = \sqrt{\gamma}\boldsymbol{x}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}$) with the same noise level ($\gamma = 0.7$). We see that higher resolution natural images tend to exhibit higher degree of redundancy in (nearby) pixels, therefore less information is destroyed with the same level of independent noise.

# Training checklist

- Decide on a data distribution to model
  - Download or create a dataset
- Come up with a noise schedule
- Pick an architecture + conditioning signal
- Latent / not latent?
- Get access to a ton of compute
- Start training!

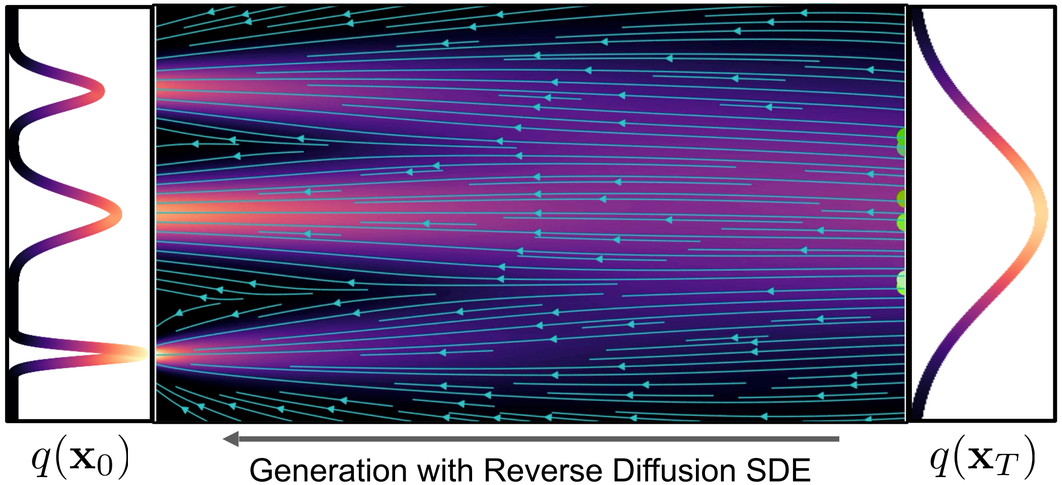
# Part 4:
# How do you use a diffusion model?

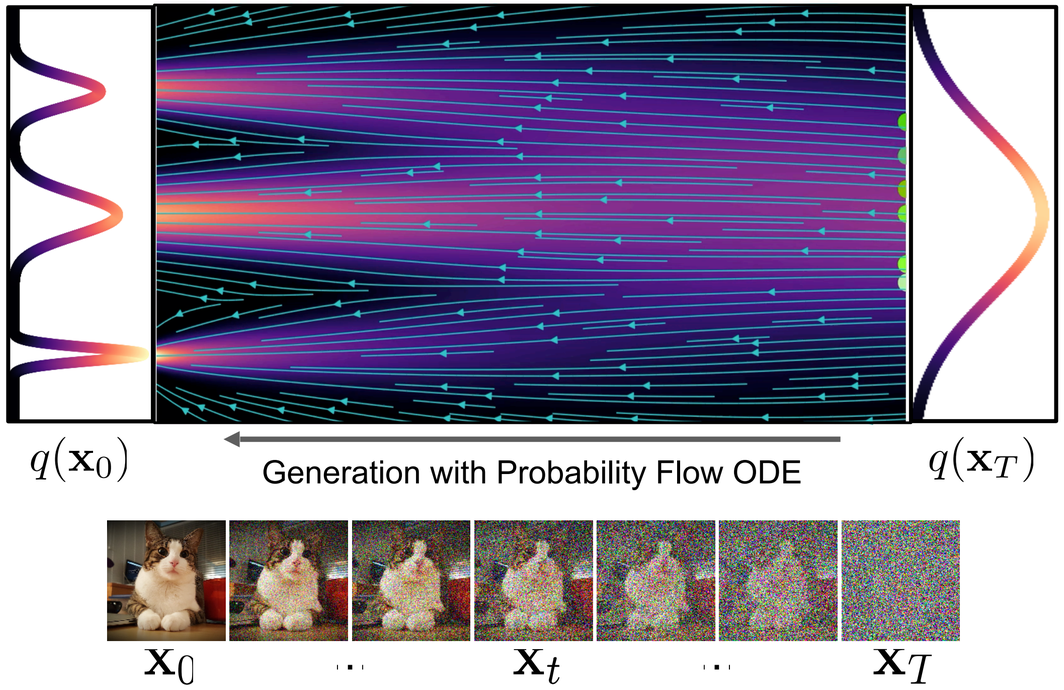
# The visualization of sampling we saw before




slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Two ways of sampling—stochastic and deterministic (SDE vs ODE)




- **Generative Reverse Diffusion SDE (stochastic):**

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\left[\mathbf{x}_t + 2\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t$$

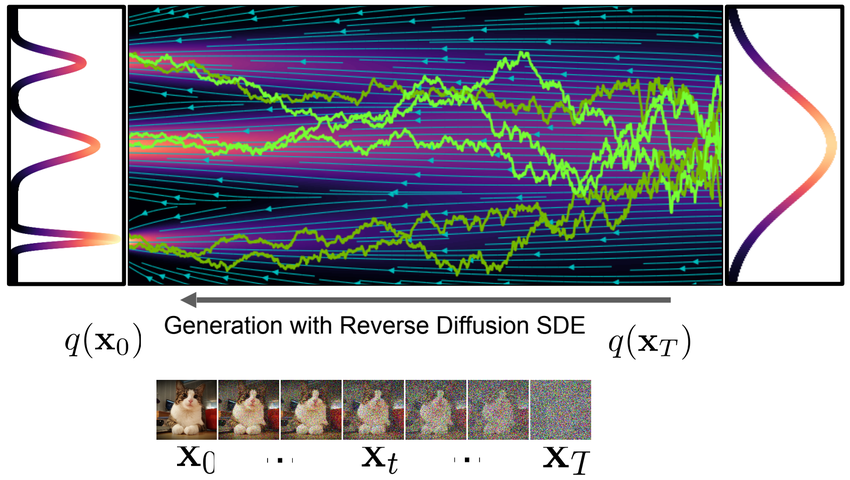
- **Generative Probability Flow ODE (deterministic):**

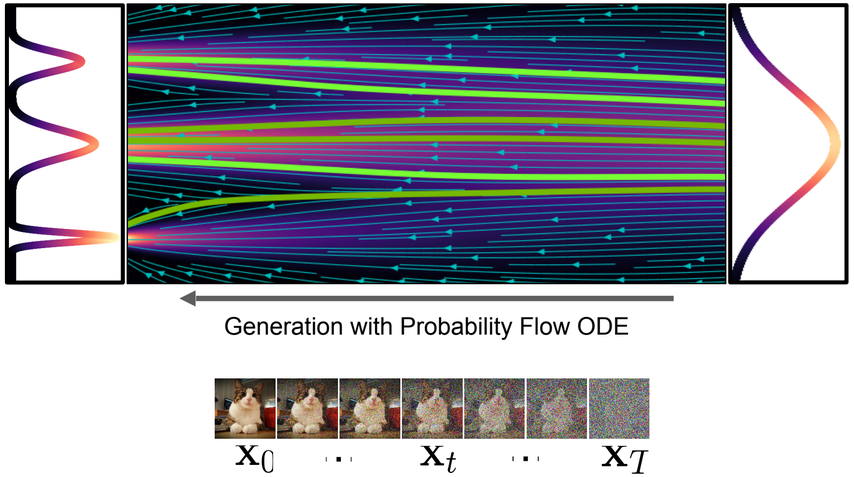
$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\left[\mathbf{x}_t + \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right]\mathrm{d}t$$

t al., *ICLR*, 2021

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

# Pros and cons




## SDE vs. ODE Sampling: Pro's and Con's

- *Pro:* Continuous noise injection can help to compensate errors during diffusion process (Langevin sampling actively pushes towards correct distribution).
- *Con:* Often slower, because the stochastic terms themselves require fine discretization during solve.

- *Pro:* Can leverage fast ODE solvers. Best when targeting very fast sampling.
- *Con:* No "stochastic" error correction, often slightly lower performance than stochastic sampling.

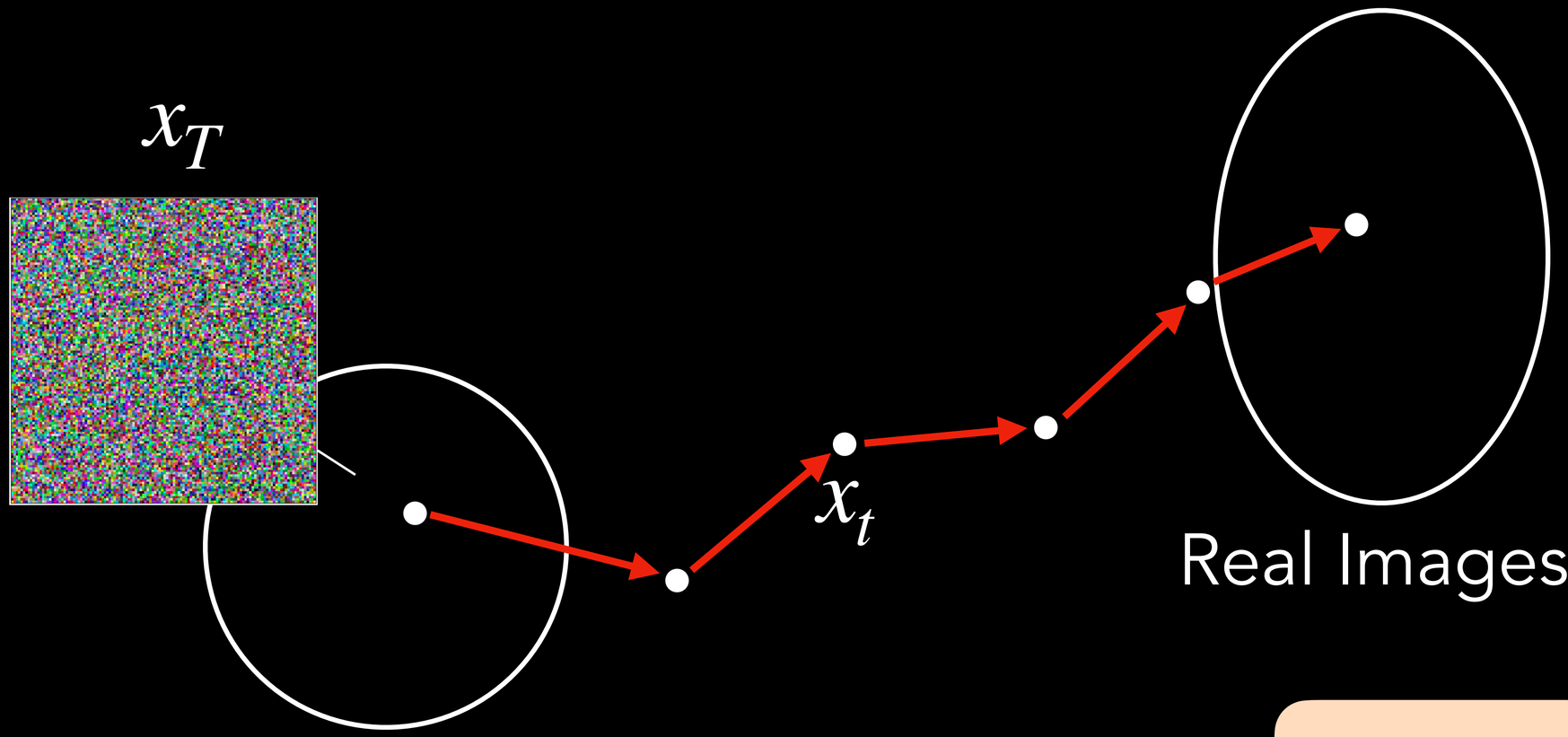
Karras et al., "Elucidating the Design Space of Diffusion-Based Generative Models", *arXiv*, 2022

One common issue:

Trained models may ignore conditioning signals!

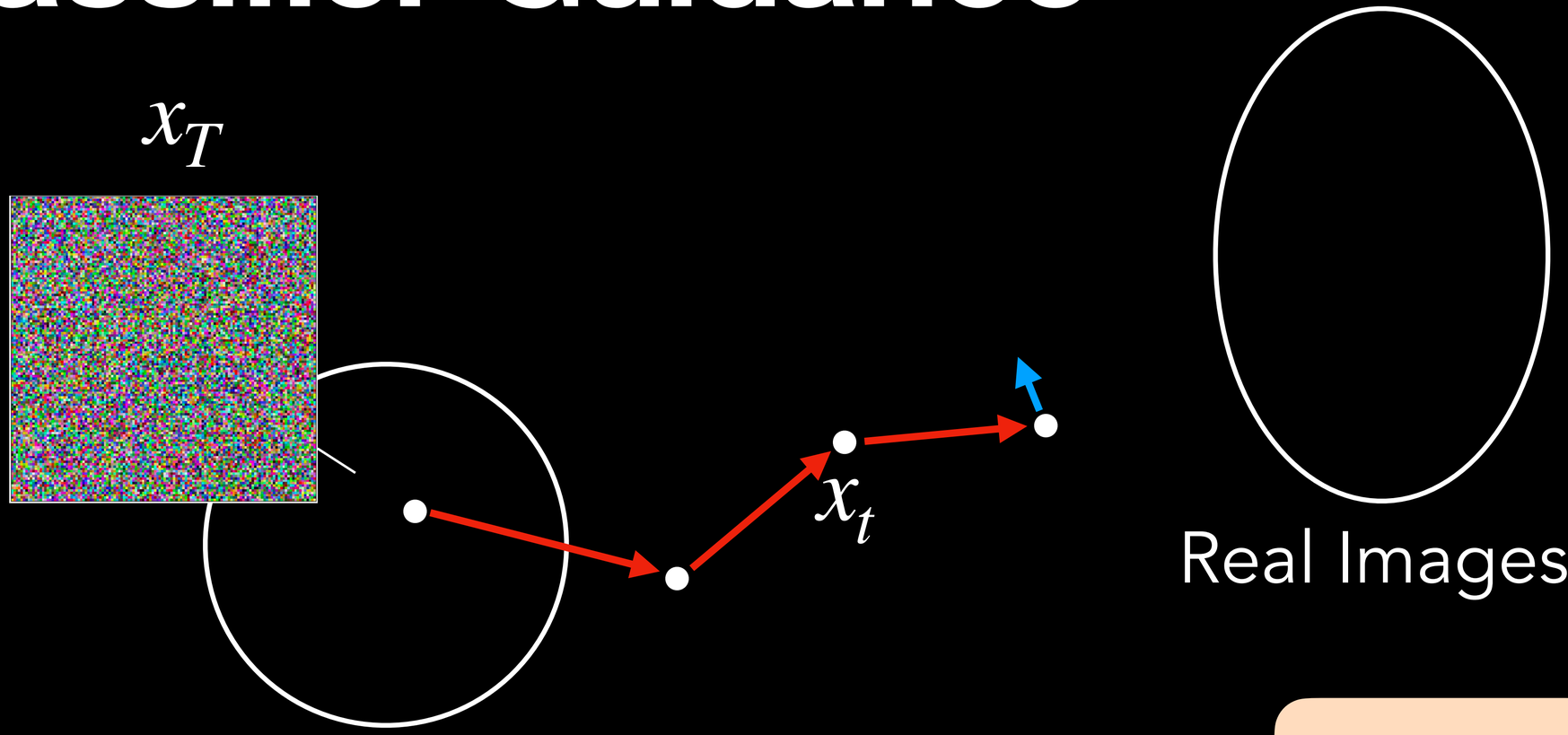
A solution:

# Guidance!




**“raspberry”** — **diffusion neural network** —

# Classifier Guidance




**“raspberry”** → **diffusion neural network**

# Classifier Guidance

$x_T$

Need a train a classifier for each conditioning input!

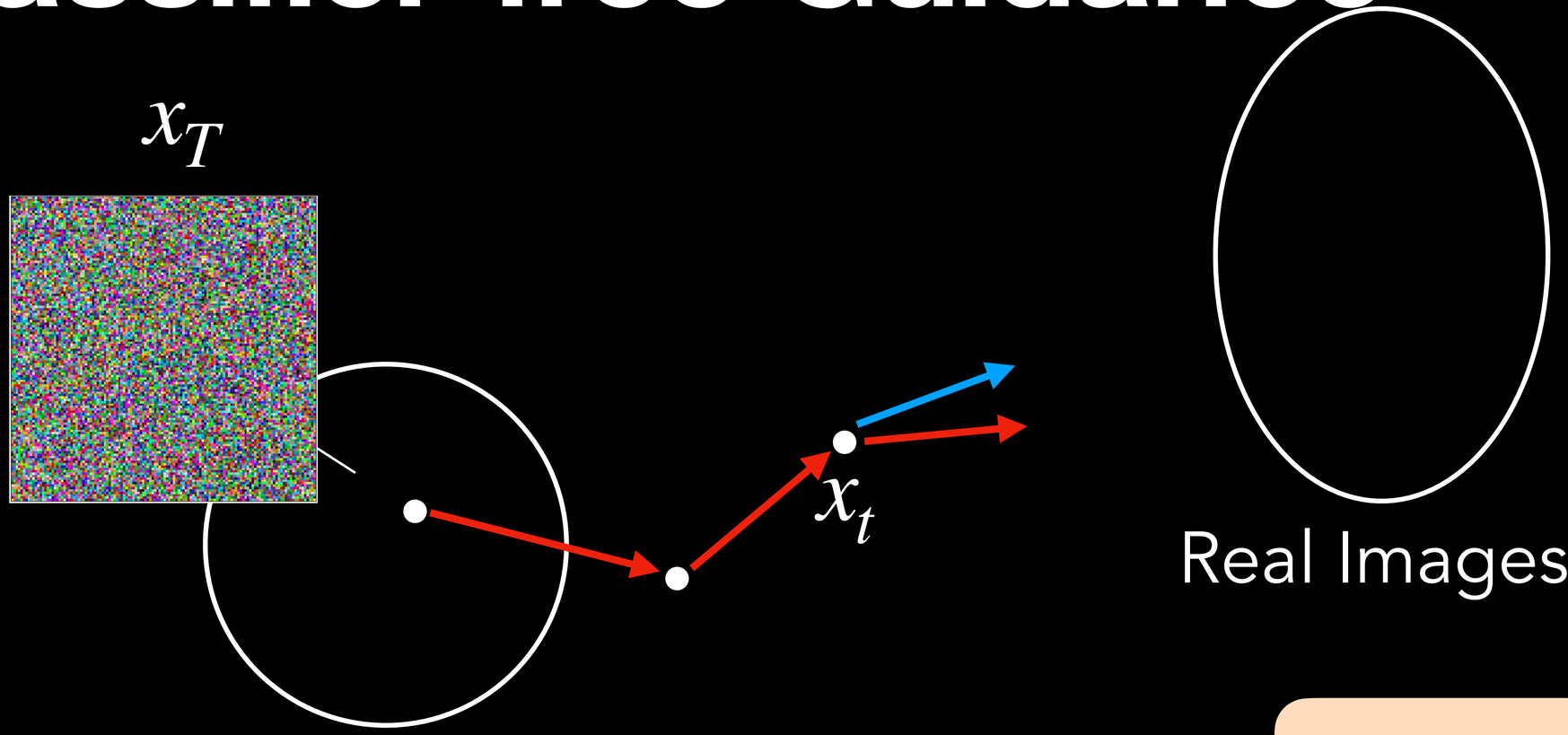
Real Images

Pure Noise

“raspberry”

diffusion neural network

# Classifier-free Guidance



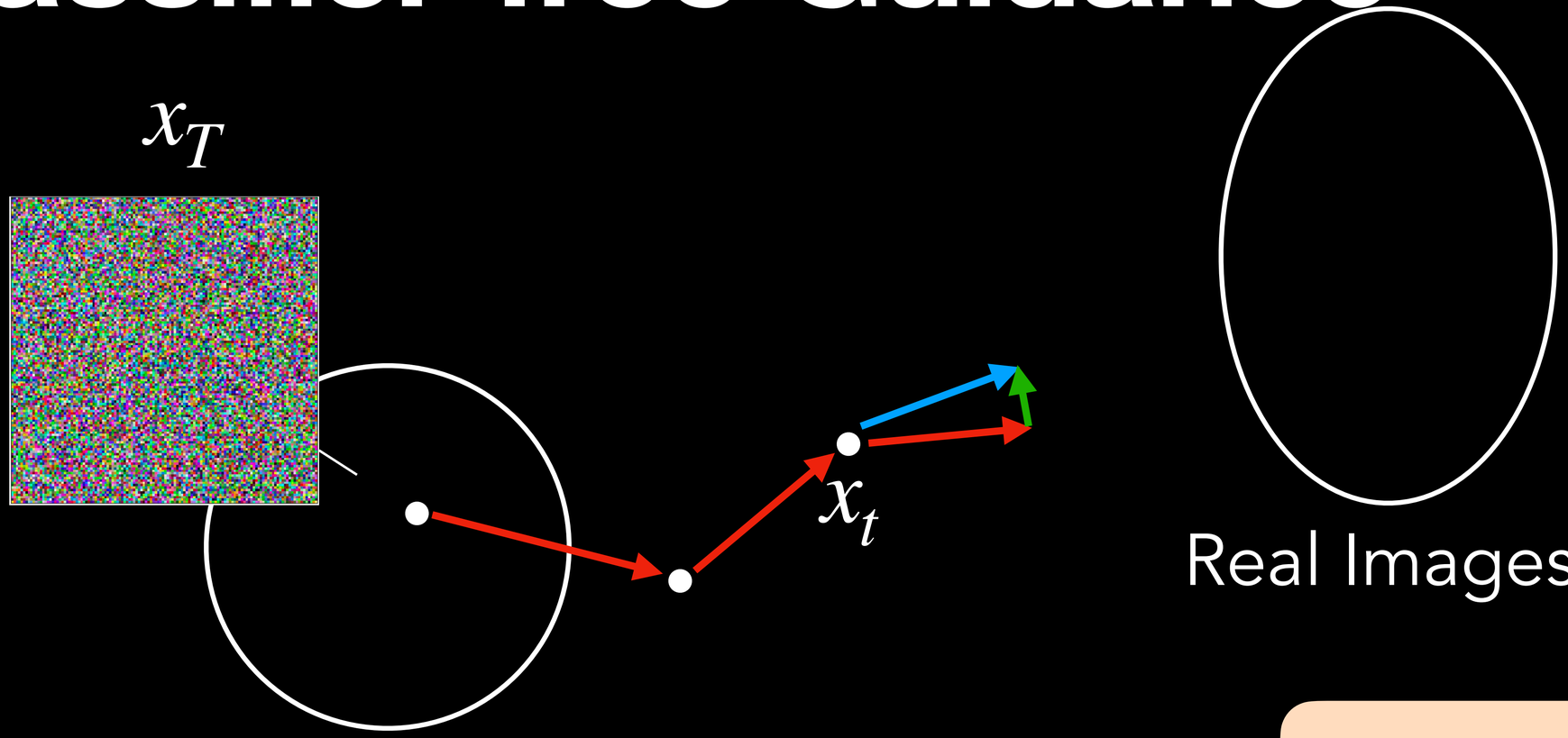

**“raspberry”**

**“”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# Classifier-free Guidance



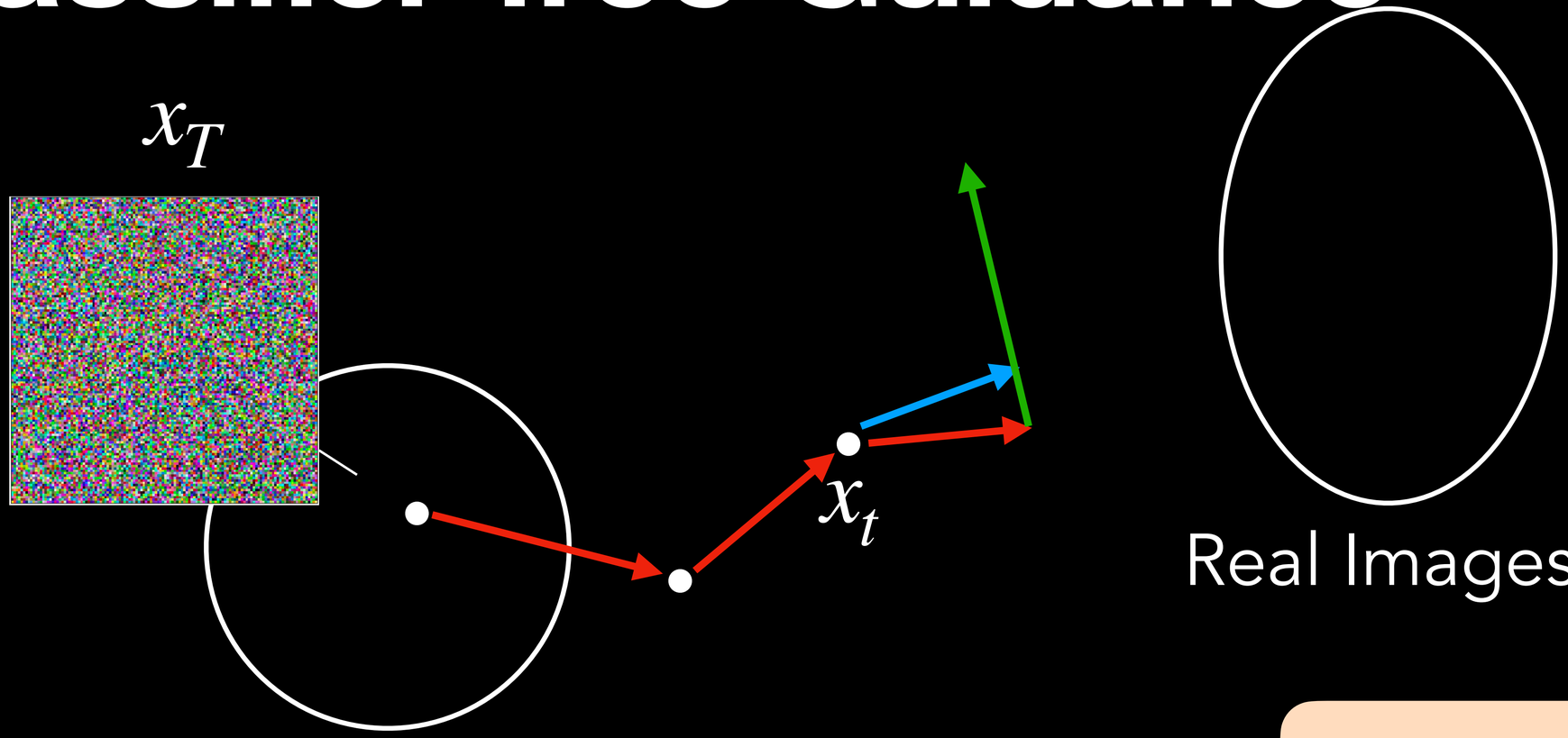

**“raspberry”**

**“”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# Classifier-free Guidance



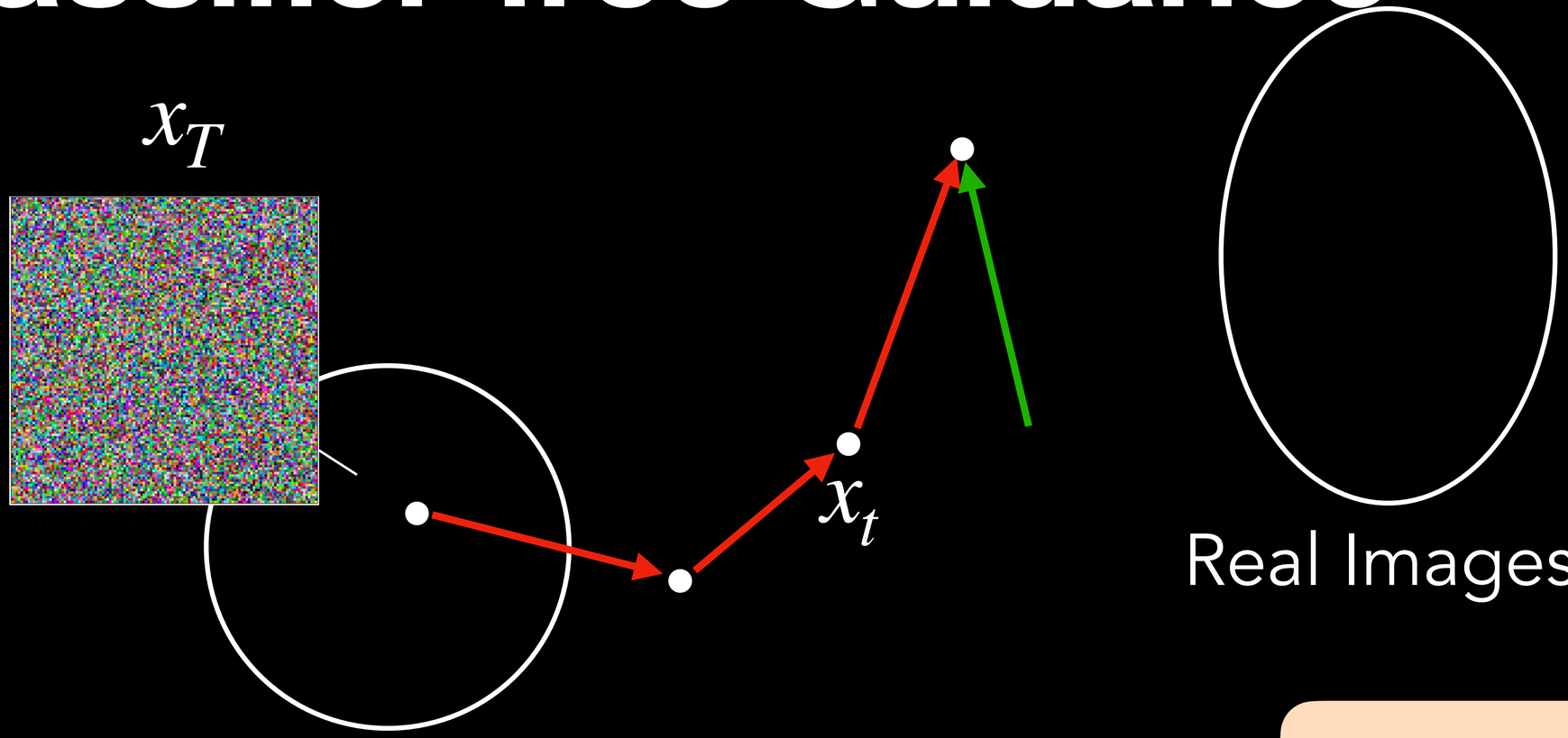

**“raspberry”**

**“”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# Classifier-free Guidance



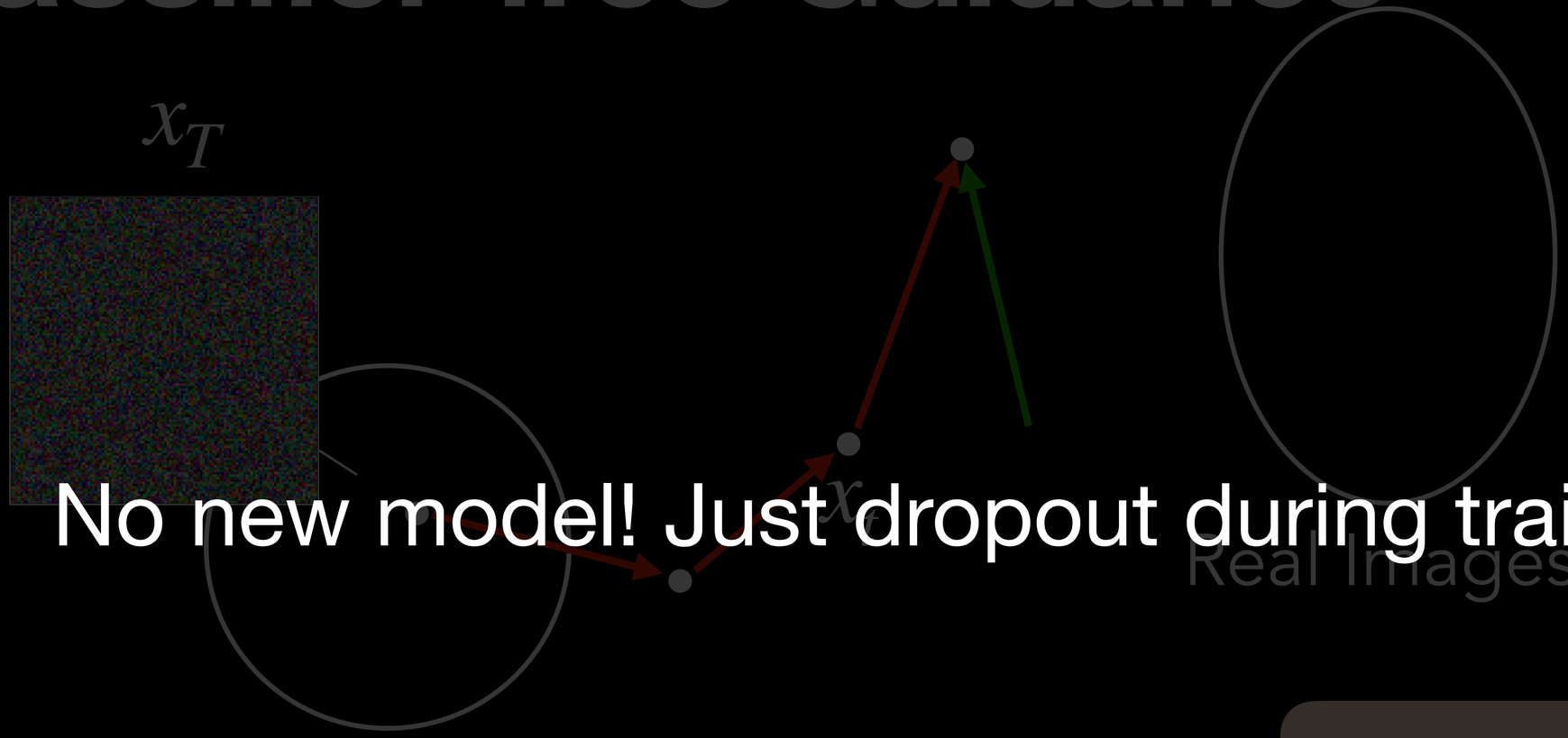

**“raspberry”**

**“”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# Classifier-free Guidance




No new model! Just dropout during training.

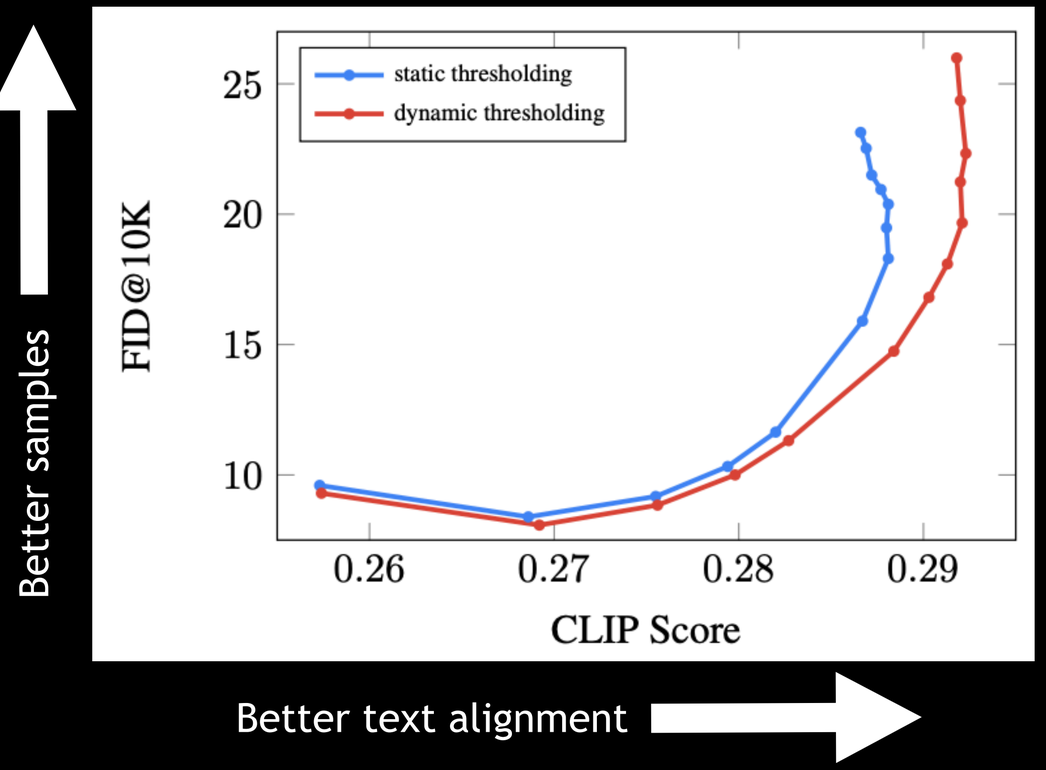
“raspberry”

“”

diffusion neural network

# Classifier-free guidance

- Large classifier-free guidance weights → better text alignment, worse image fidelity




Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, arXiv 2022.

Another issue:

# Sampling is slow!

Often need 10-1000+ steps to generate high quality samples

# How to make sampling faster?

One solution: distill diffusion models into models using just 4-8 sampling steps!

**One example:**

*Progressive distillation for fast sampling of diffusion models, Salimans & Ho, ICLR 2022*

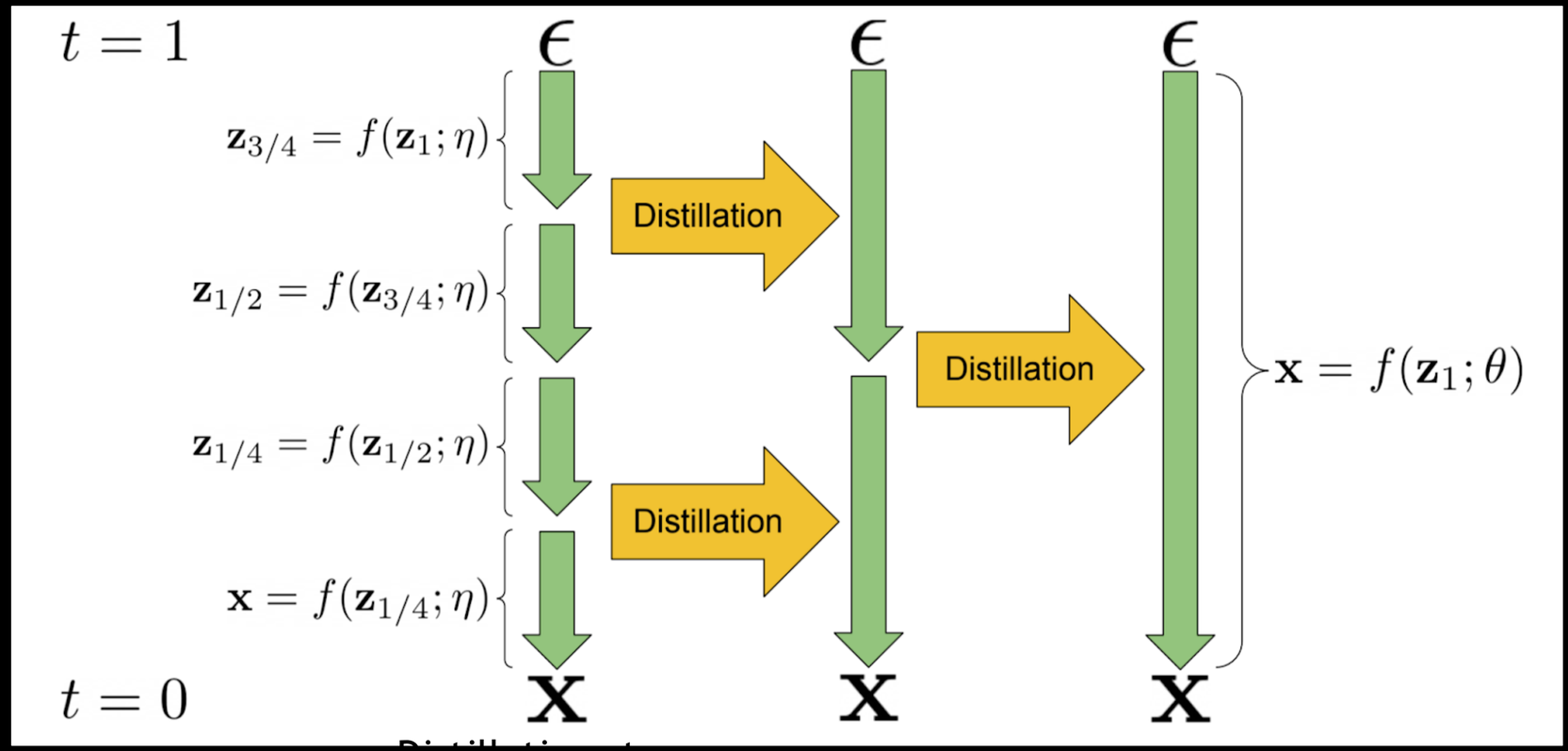
*On Distillation of Guided Diffusion Models, Meng et al., CVPR 2023*

# Progressive distillation

Distill a deterministic ODE sampler (i.e. DDIM sampler) to the same model architecture.

A “student” model is learned to distill two adjacent sampling steps of the “teacher” model to one sampling step.

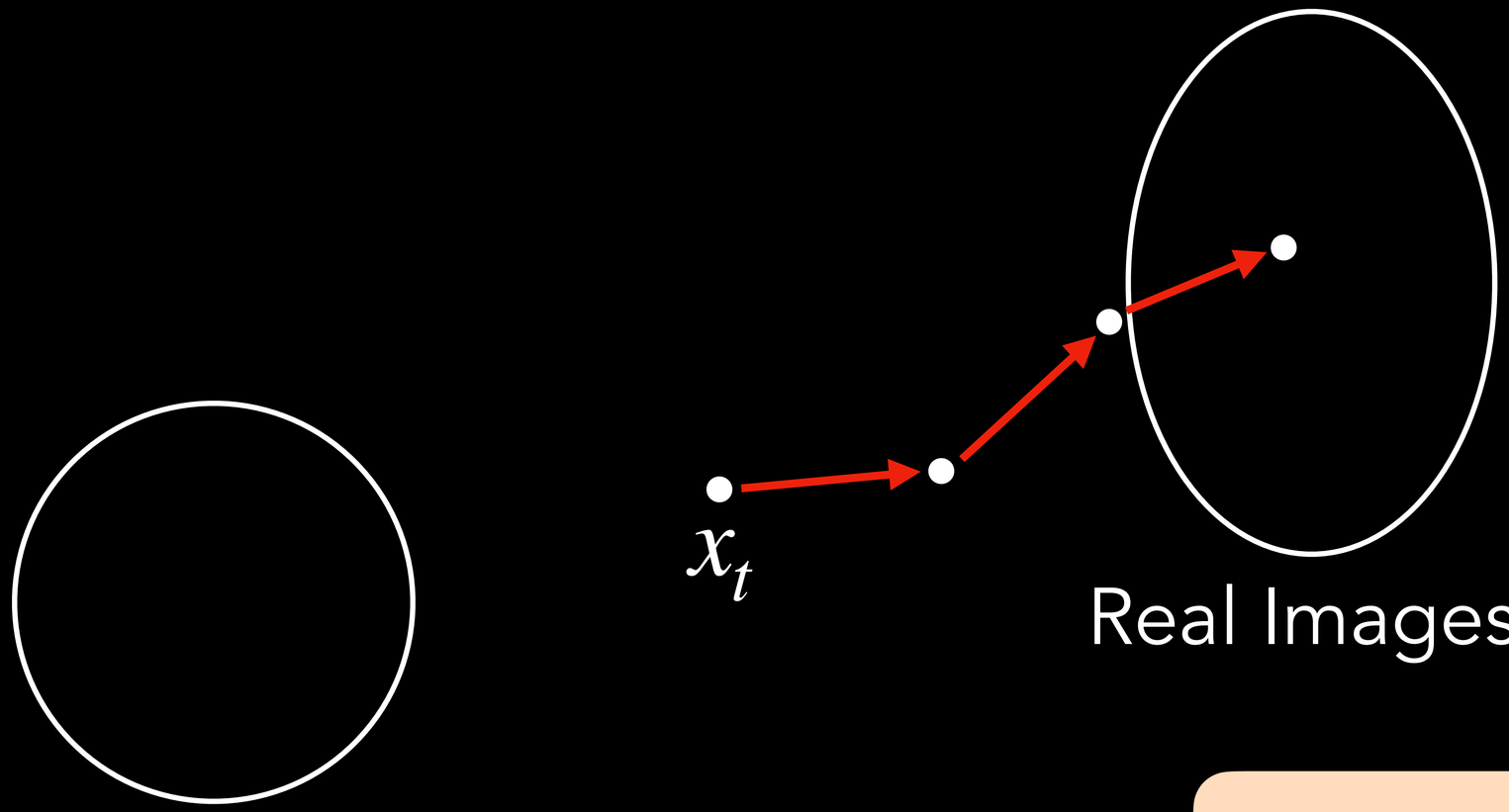
At next stage, the “student” model from previous stage will serve as the new “teacher” model.




Salimans & Ho, “Progressive distillation for fast sampling of diffusion models”, ICLR 2022.

# Some other sampling tricks

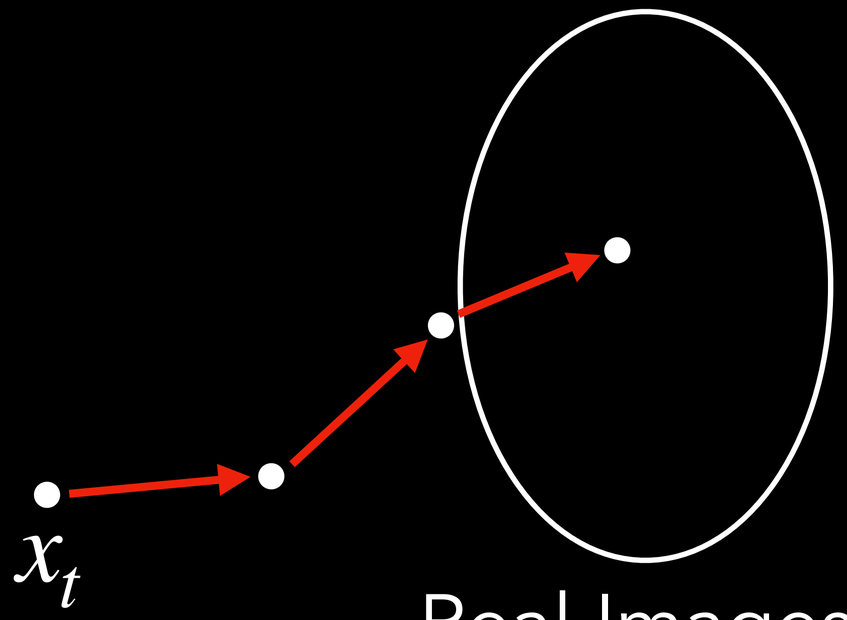
# SDEdit: don’t start from pure noise!




diffusion neural network

# SDEdit: don’t start from pure noise!

Pure Noise

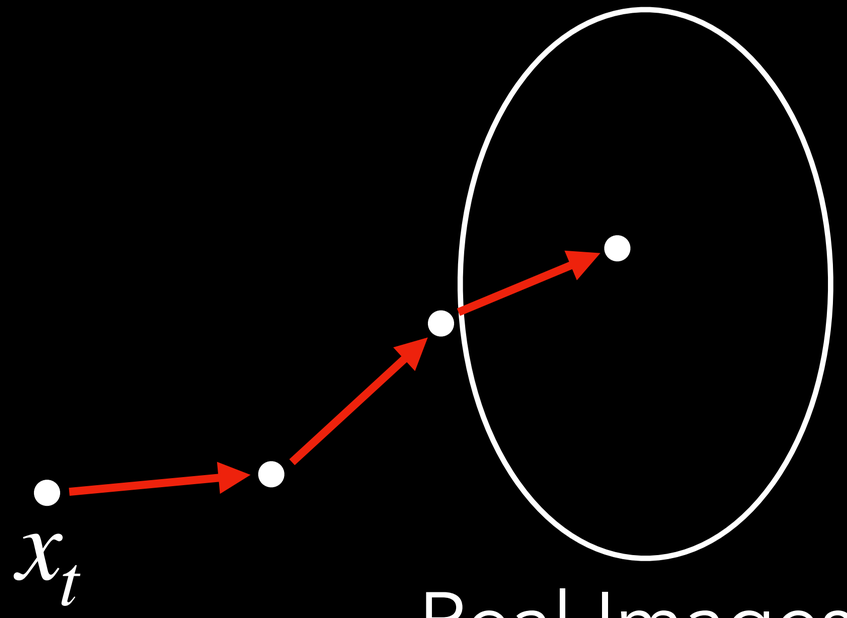



Real Images

diffusion neural network

# SDEdit: don’t start from pure noise!

e Noise




diffusion neural network

Steve Seitz, Graphics in 5 Minutes

# SDEdit: don’t start from pure noise!

$x_t$

Real Images

**“Guy with beard”**

**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# SDEdit: don’t start from pure noise!

$x_t$

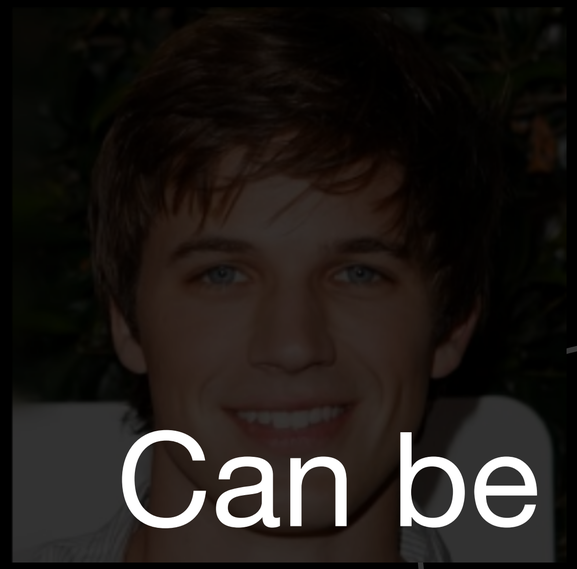
Real Images

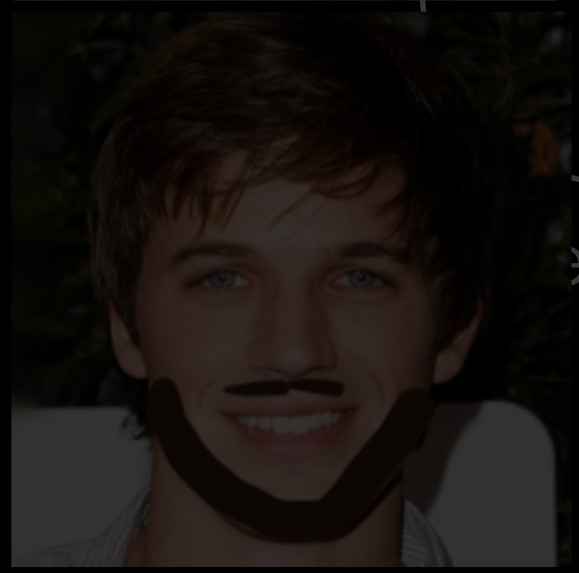
**“Guy with beard”**

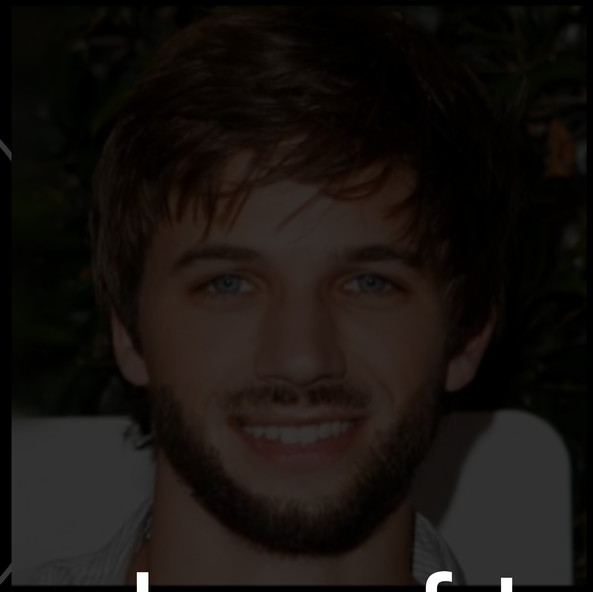
**diffusion neural network**

Steve Seitz, Graphics in 5 Minutes

# SDEdit: don’t start from pure noise!







Real Images

“Guy with beard”

diffusion neural network

Can be hard to pick the right value of t

Steve Seitz, Graphics in 5 Minutes

# Prompt-to-Prompt: Generating two similar images

Photo of a cat riding a bicycle

Photo of a cat riding a car

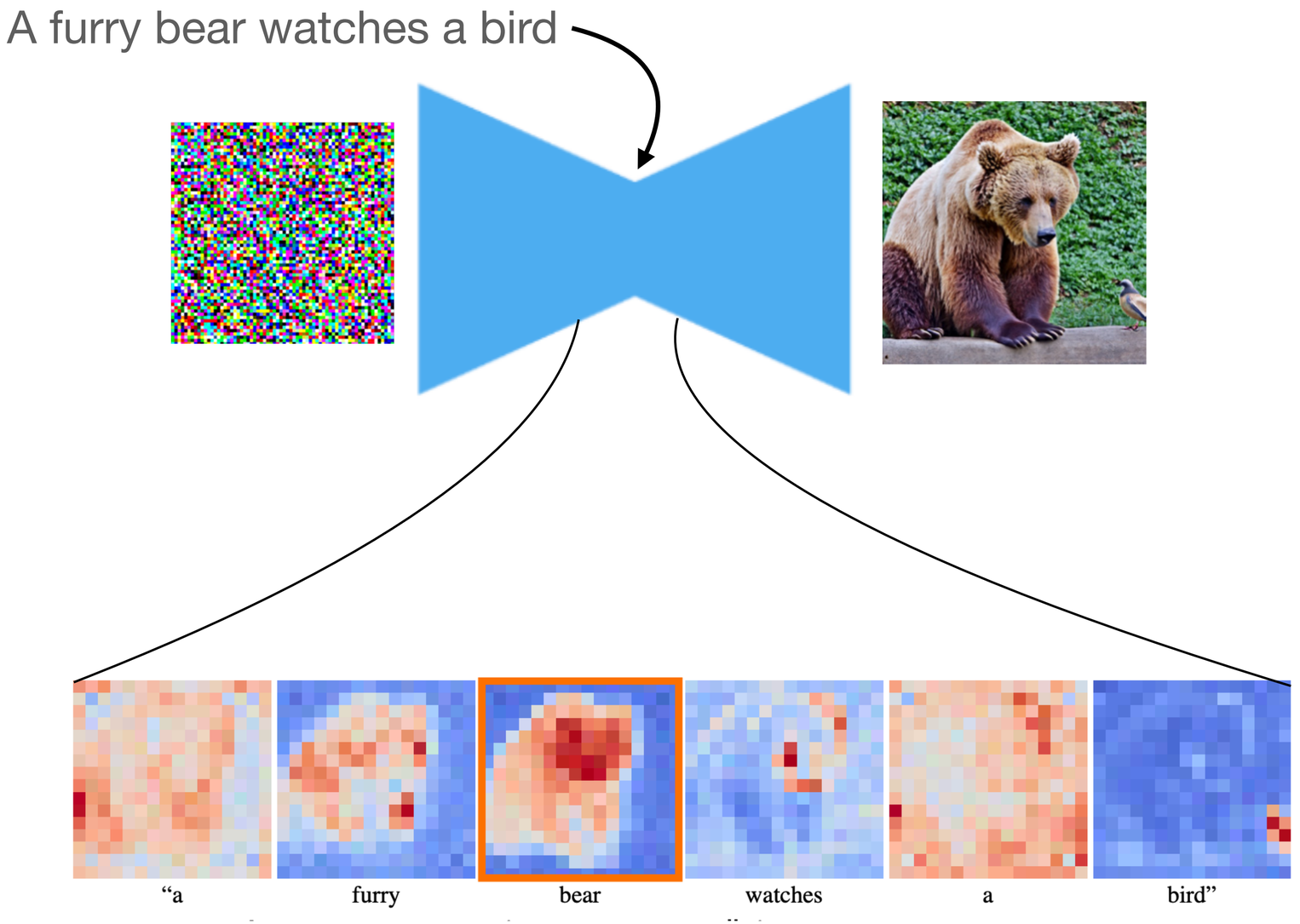
Prompt-to-Prompt Image Editing with Cross-Attention Control, Hertz et al.

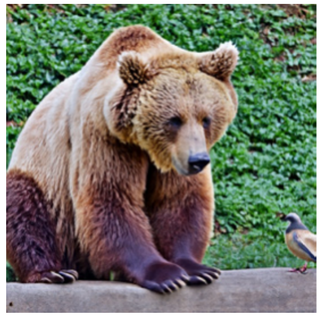
# Prompt-to-Prompt: Generating two similar images

Photo of a cat riding a bicycle

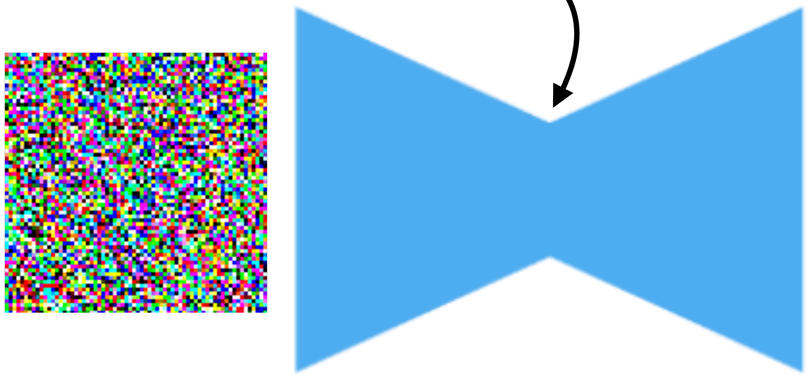

Photo of a cat riding a car

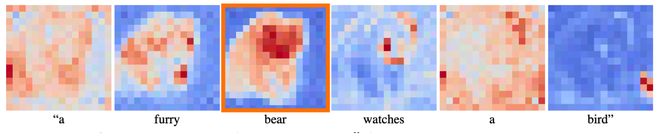

Prompt-to-Prompt Image Editing with Cross-Attention Control, Hertz et al.


A furry bear watches a bird
“a
furry
bear
watches
a
bird”

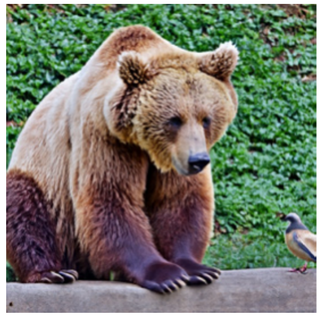


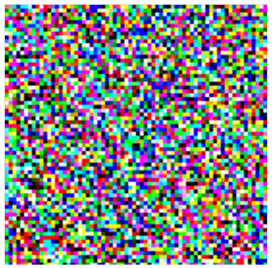
A furry bear watches a ***CAT***

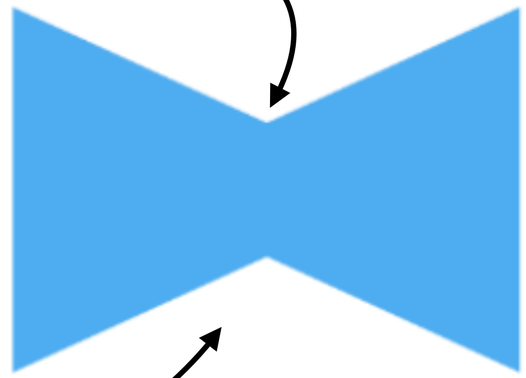


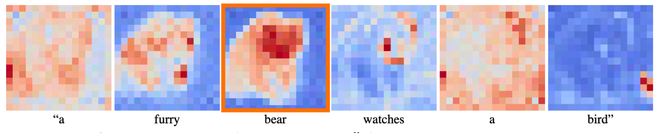


A furry bear watches a ***CAT***

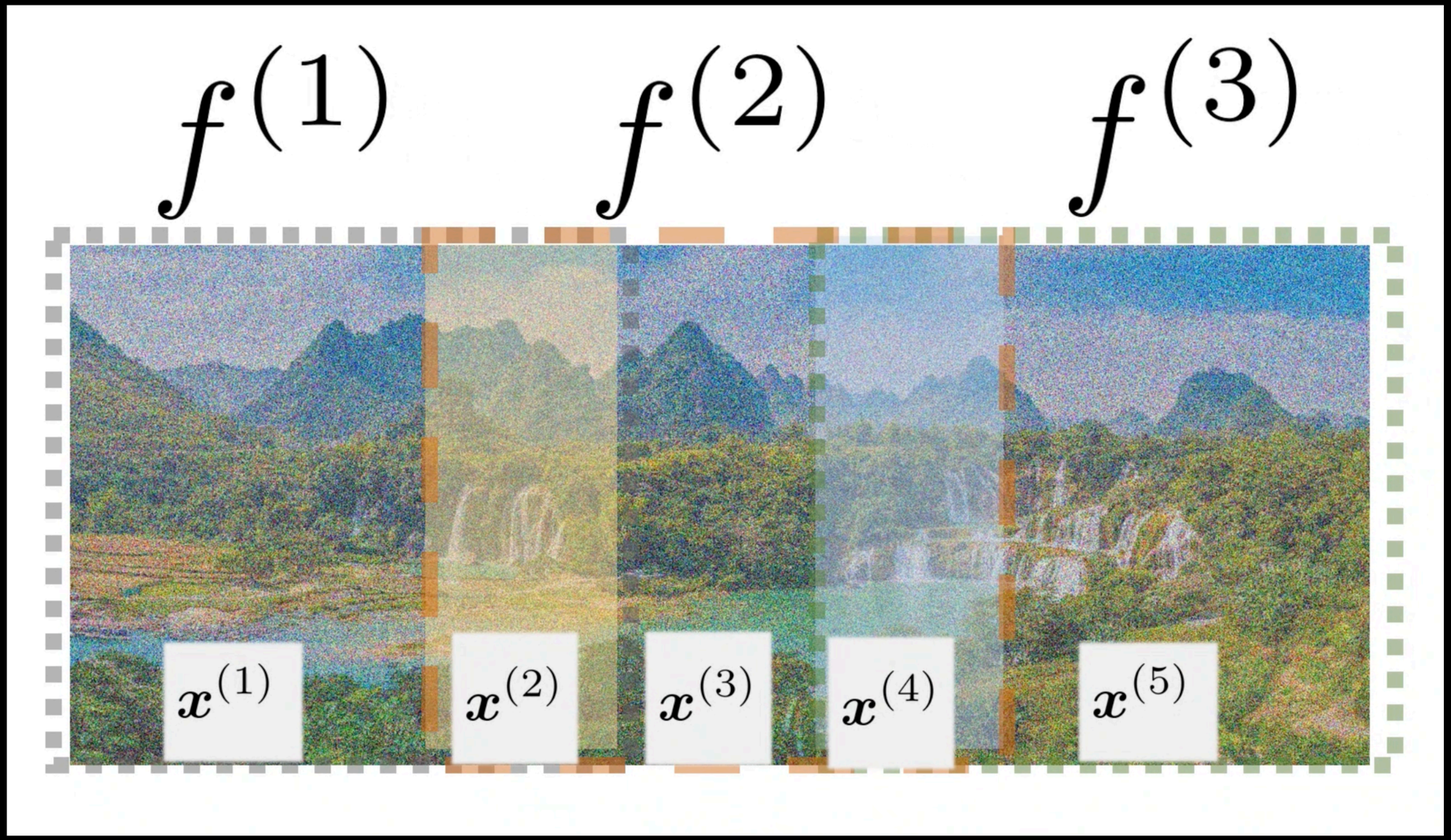




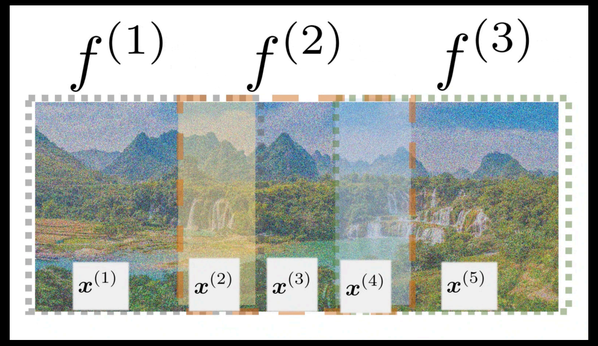




# Mixing scores



# Mixing scores




# Can’t do this with latent models!

Encode

Latent Image

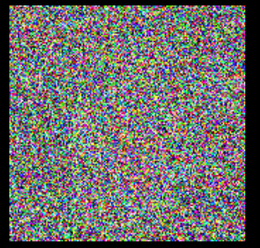
$\mathbf{x}_t$

$t$

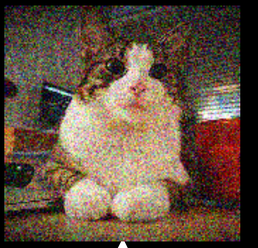
Time Representation

Fully-connected

$\epsilon_\theta(\mathbf{x}_t, t)$



$\epsilon_\theta(\mathbf{x}_t, t)$



Decode

Latent Image