Generative Models II

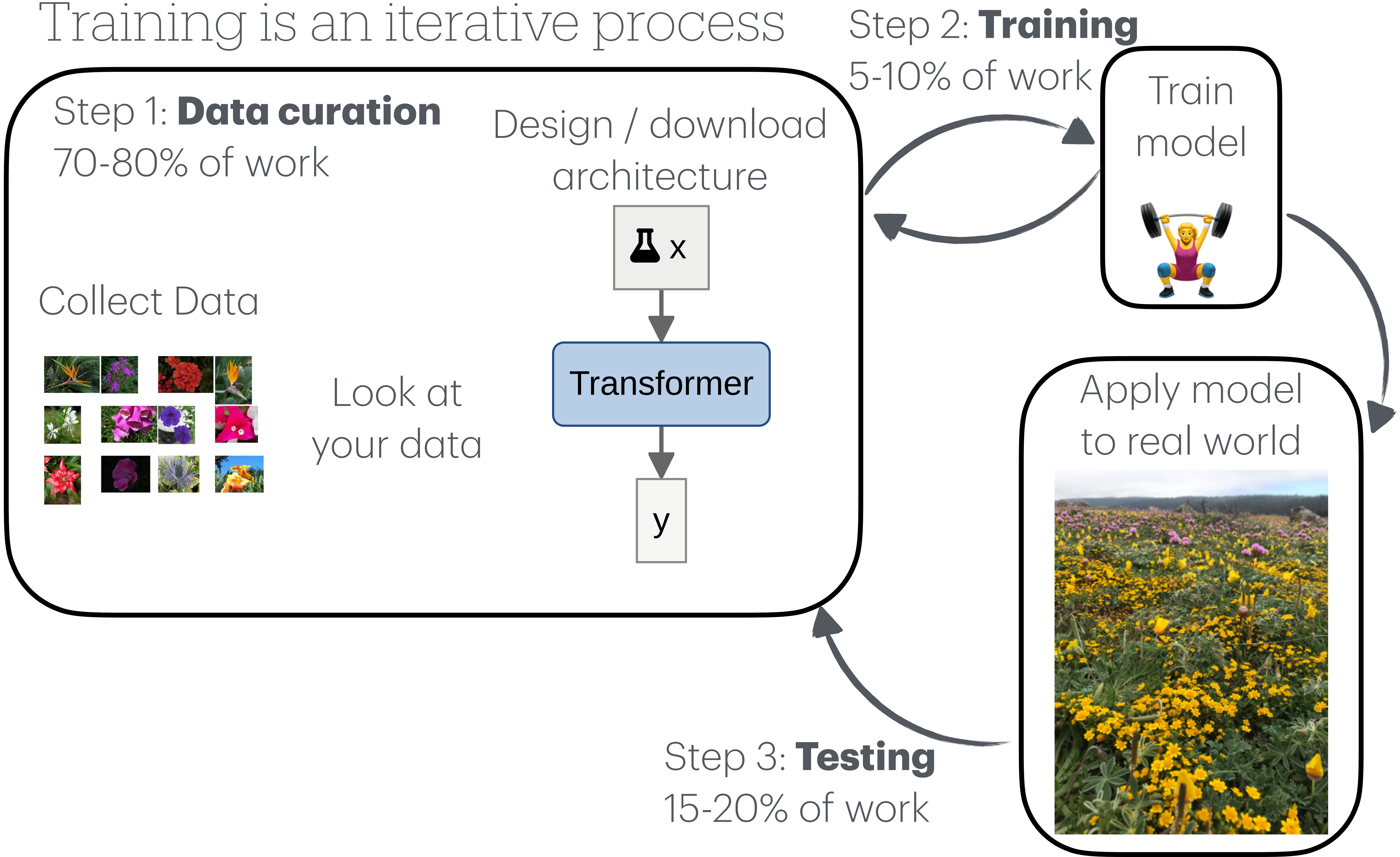
Homework 4

Discussion

- Did Claude / Codex zero-shot the homework?
  - What was your prompt?
- How did you get to your solution?
  - Iterative process. How many models did you train in total?
  - Logging / Visualizations
  - Other tips / hints?

How to train a network?

Training is an iterative process

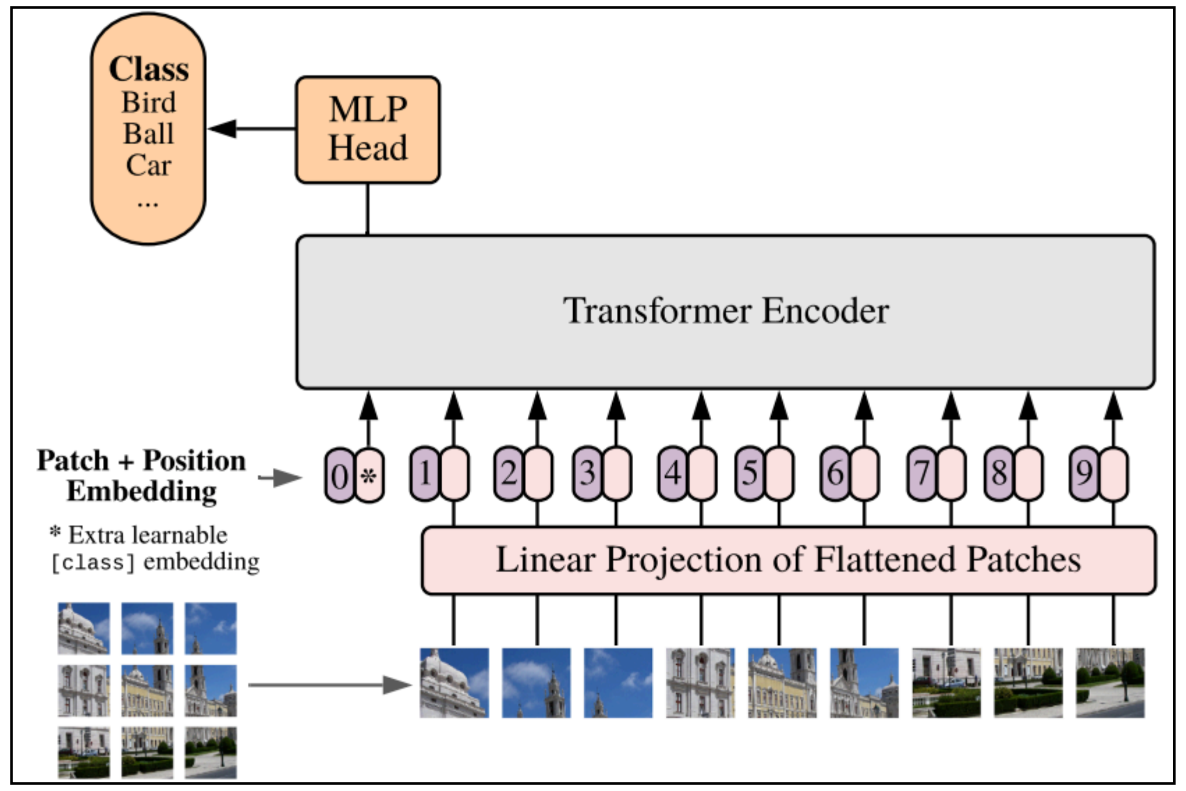


Generative Models

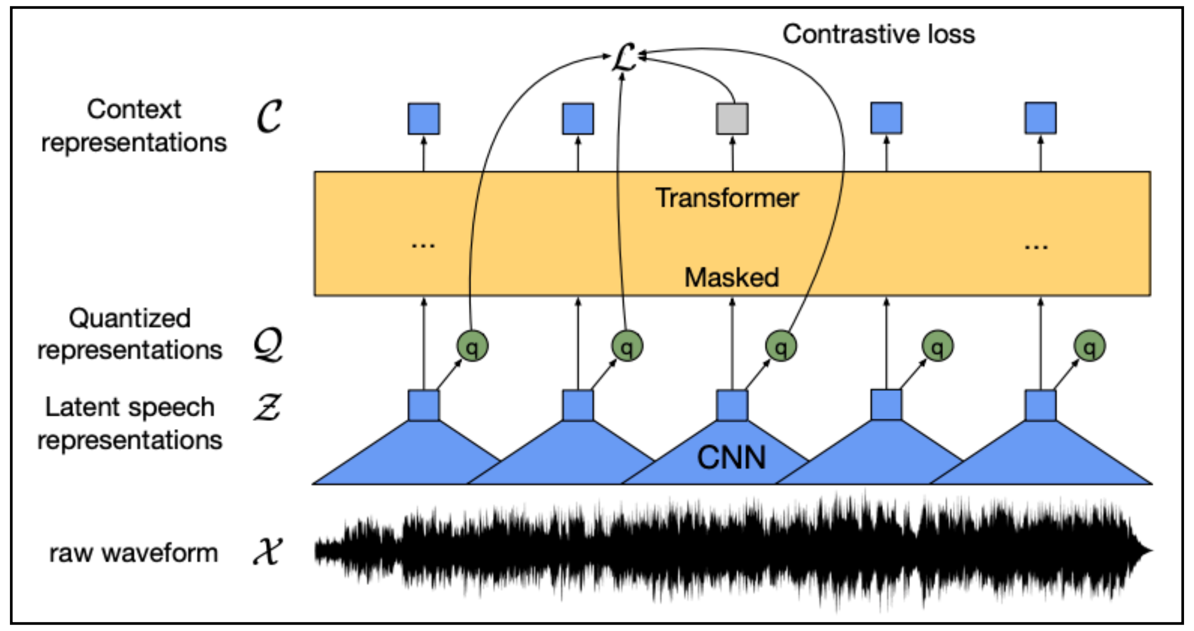
Philipp Krähenbühl, UT Austin

Recap: Discriminative models

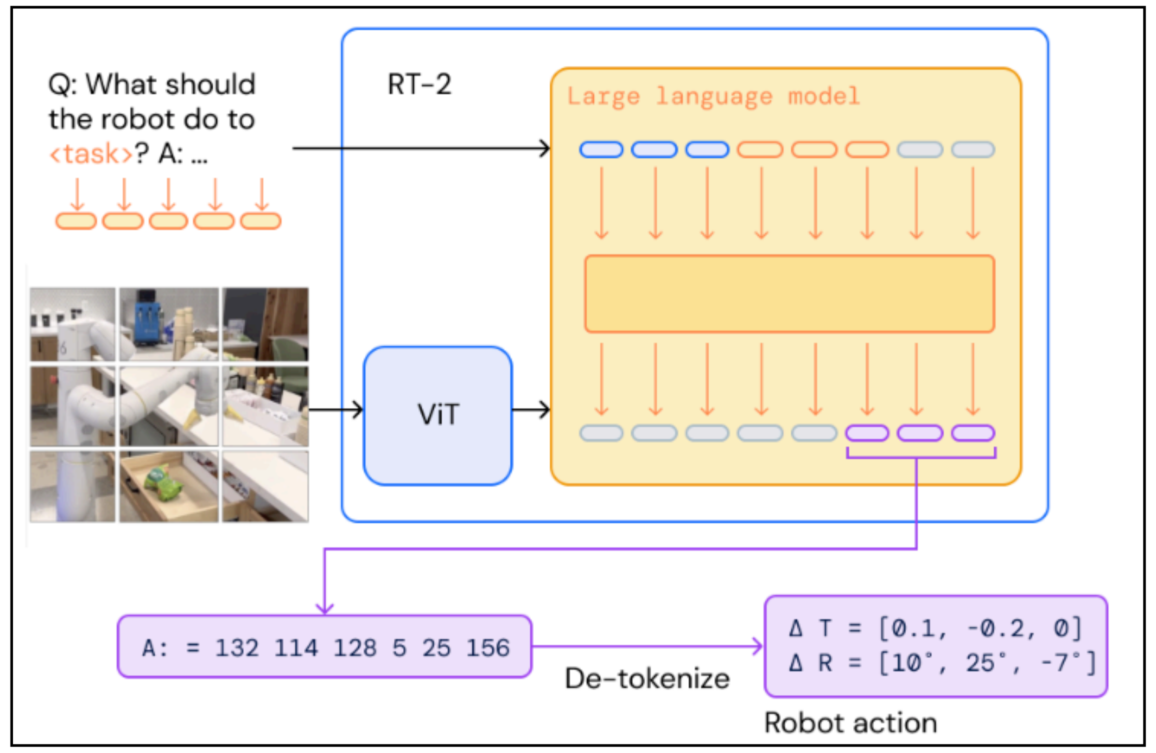
- Discriminative model: $P(Y|X)$
- Examples:
  - Image/video recognition
  - Speech recognition
  - Control policies
  - Weather prediction
  - …



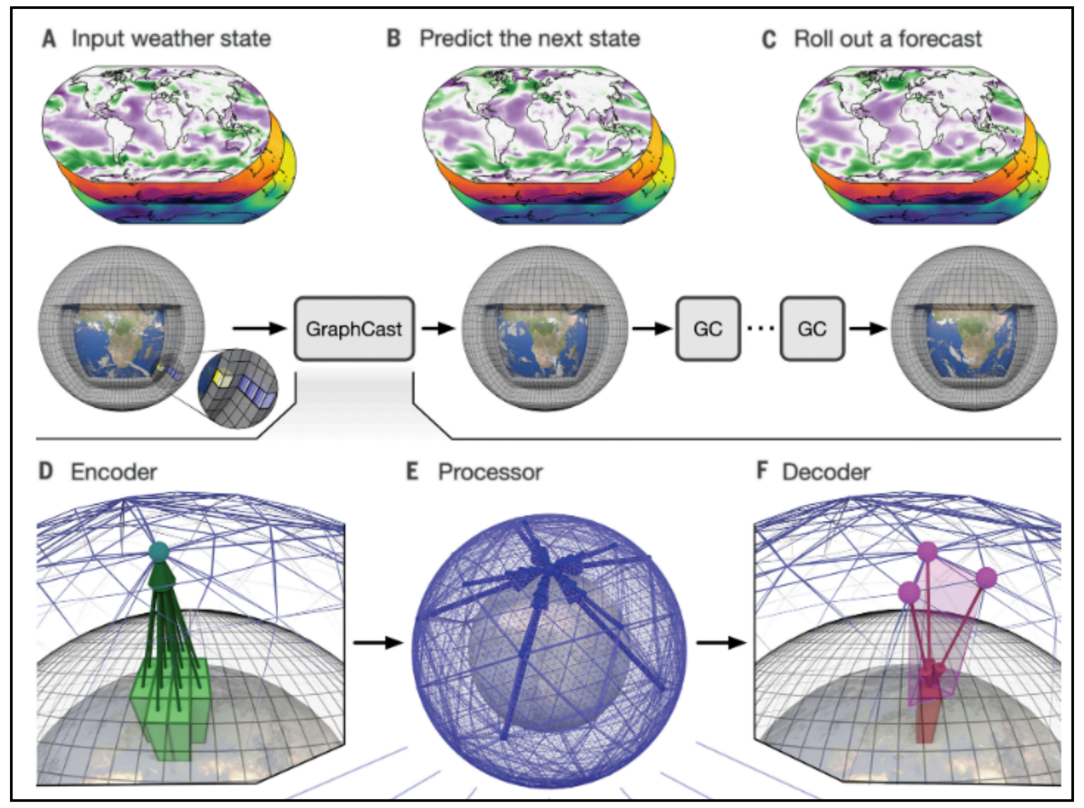
[1] Vision Transformer



[2] Wave2vec 2.0



[3] RT-2



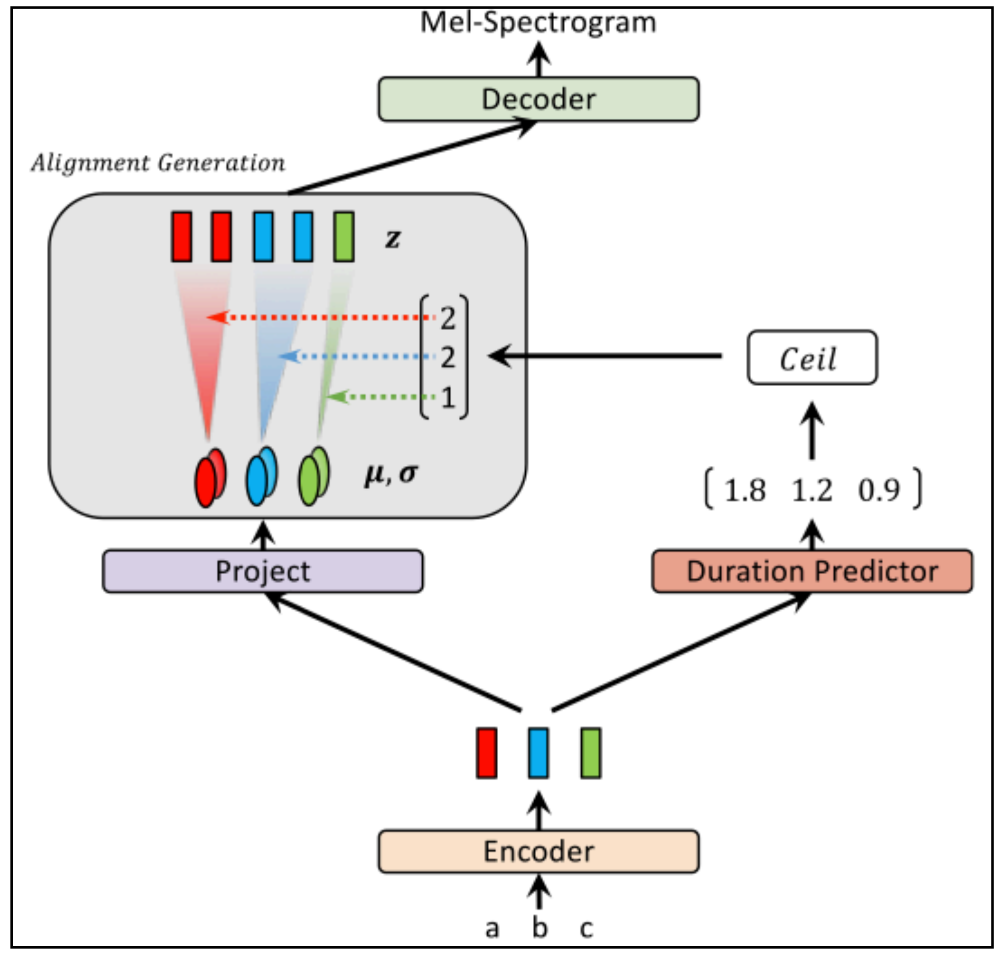
[4] GraphCast

[1] Dosovitskiy, Alexey, et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." International Conference on Learning Representations. 2020.
[2] Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations." Advances in neural information processing systems 33 (2020): 12449-12460.
[3] Brohan, Anthony, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control." arXiv preprint arXiv:2307.15818 (2023).
[4] Remi Lam et al. ,Learning skillful medium-range global weather forecasting.Science382,1416-1421(2023).

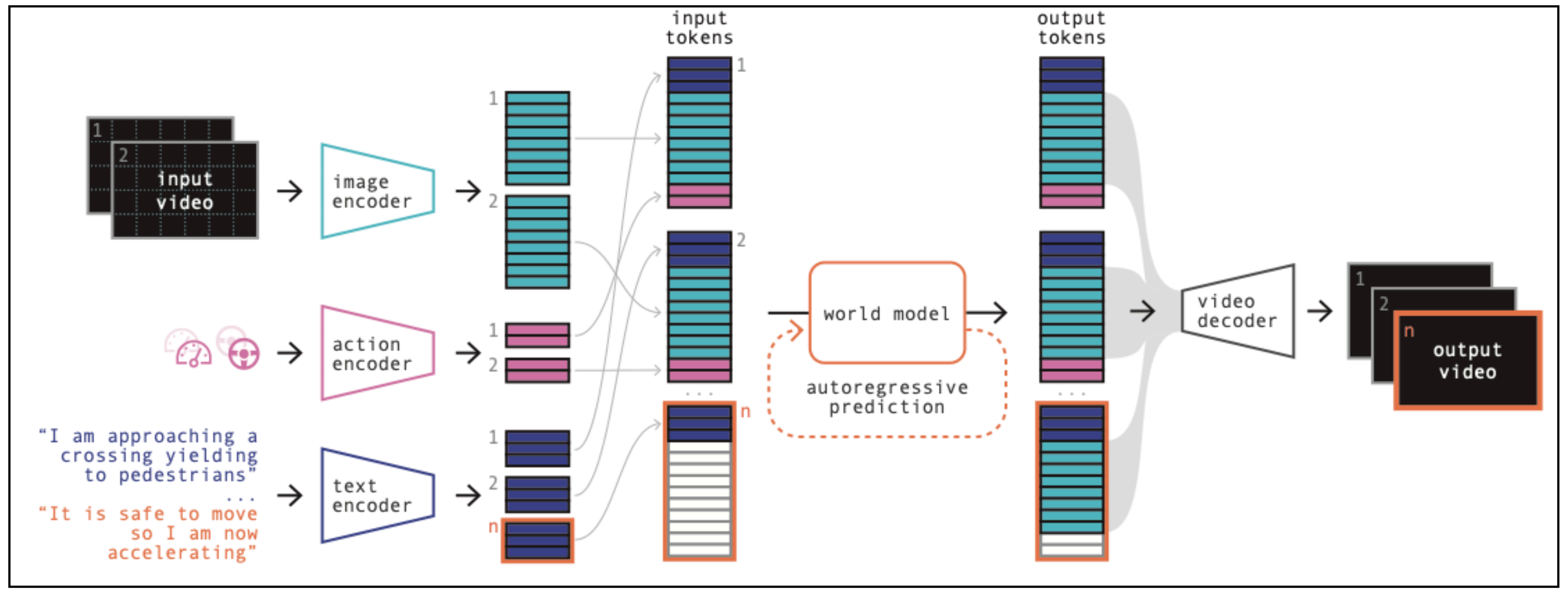
Recap: Generative models

- Generative model: $P(X)$
- Examples:
  - Image/video generation
  - Speech synthesis
  - Physics simulation / world modeling
  - Weather simulation (gaming)
  - ...

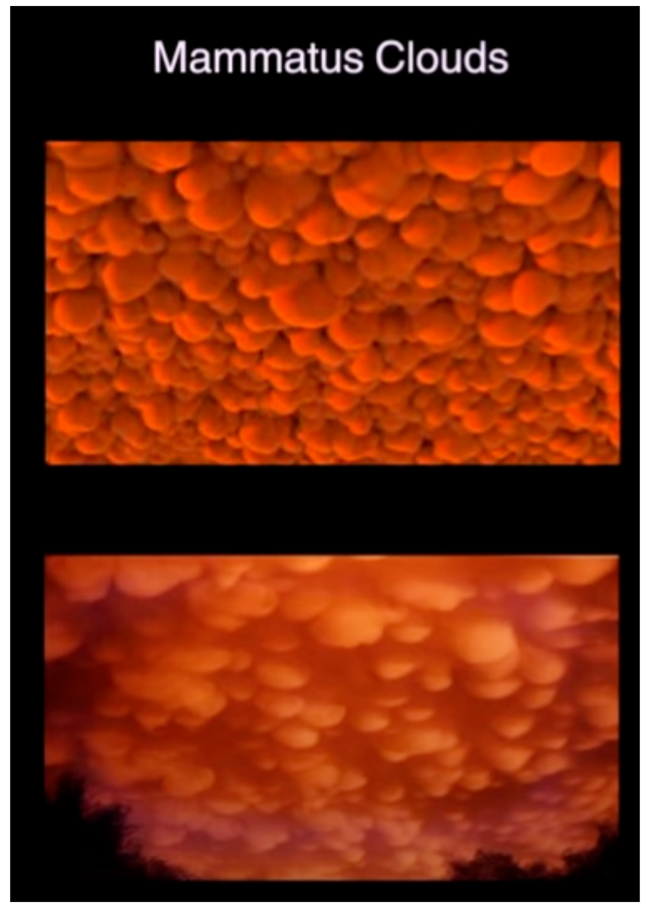
[1] Sora



[2] Glow-TTS



[3] GAIA-1



[4] Weatherscapes

[1] Brook, Tim, et al. "Video generation models as world simulators" OpenAI Blog (2024)
[2] Kim, Jaehyeon, et al. "Glow-tts: A generative flow for text-to-speech via monotonic alignment search." Advances in Neural Information Processing Systems 33 (2020): 8067-8077..
[3] Hu, Anthony, et al. "Gaia-1: A generative world model for autonomous driving." arXiv preprint arXiv:2309.17080 (2023).
[4]J. A. Amador Herrera, et al. "Weatherscapes: Nowcasting Heat Transfer and Water Continuity." ACM Transactions on Graphics (SIGGRAPH Asia 2021), Vol. 40, No. 6, Article 204..

Recap: Generative models

Two kinds of models

Sampling based $x \sim P(X)$

- Sample $z \sim P(Z)$
- Learn transformation
  - $P(x \mid z)$ or $f : z \to x$

z

Deep Network

Density estimation based $P(X)$

- Learn special form of $P(X)$
- Model specific sampling / generation

Deep Network

$P(X)$

Recap: Auto-regressive models

$$P(x) = P(x_1)P(x_2 | x_1)P(x_3 | x_1, x_2)P(x_4 | x_1 \ldots x_3)\ldots$$

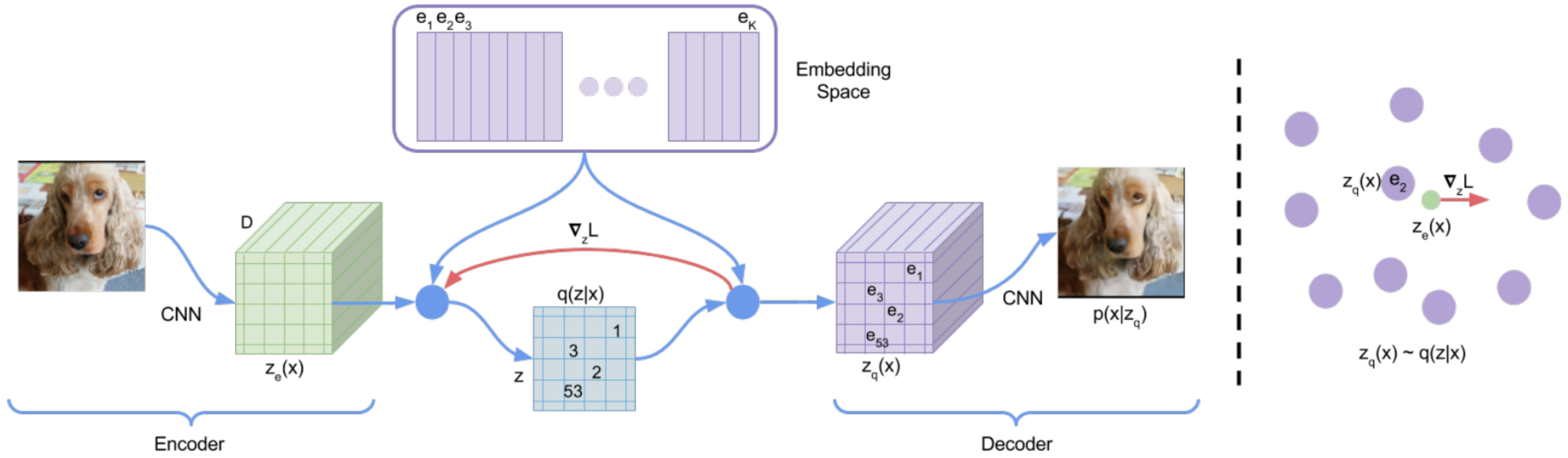
- $P(x_i | x_1 \ldots x_{i-1}) = \text{softmax}(f(x_1 \ldots x_{i-1}))$
- Basis of most LLM models
- Easy estimation of $P(x)$
- Easy sampling
  $x_1 \sim P(X_1); x_2 \sim P(X_2 | x_1)$
  - Slow sampling

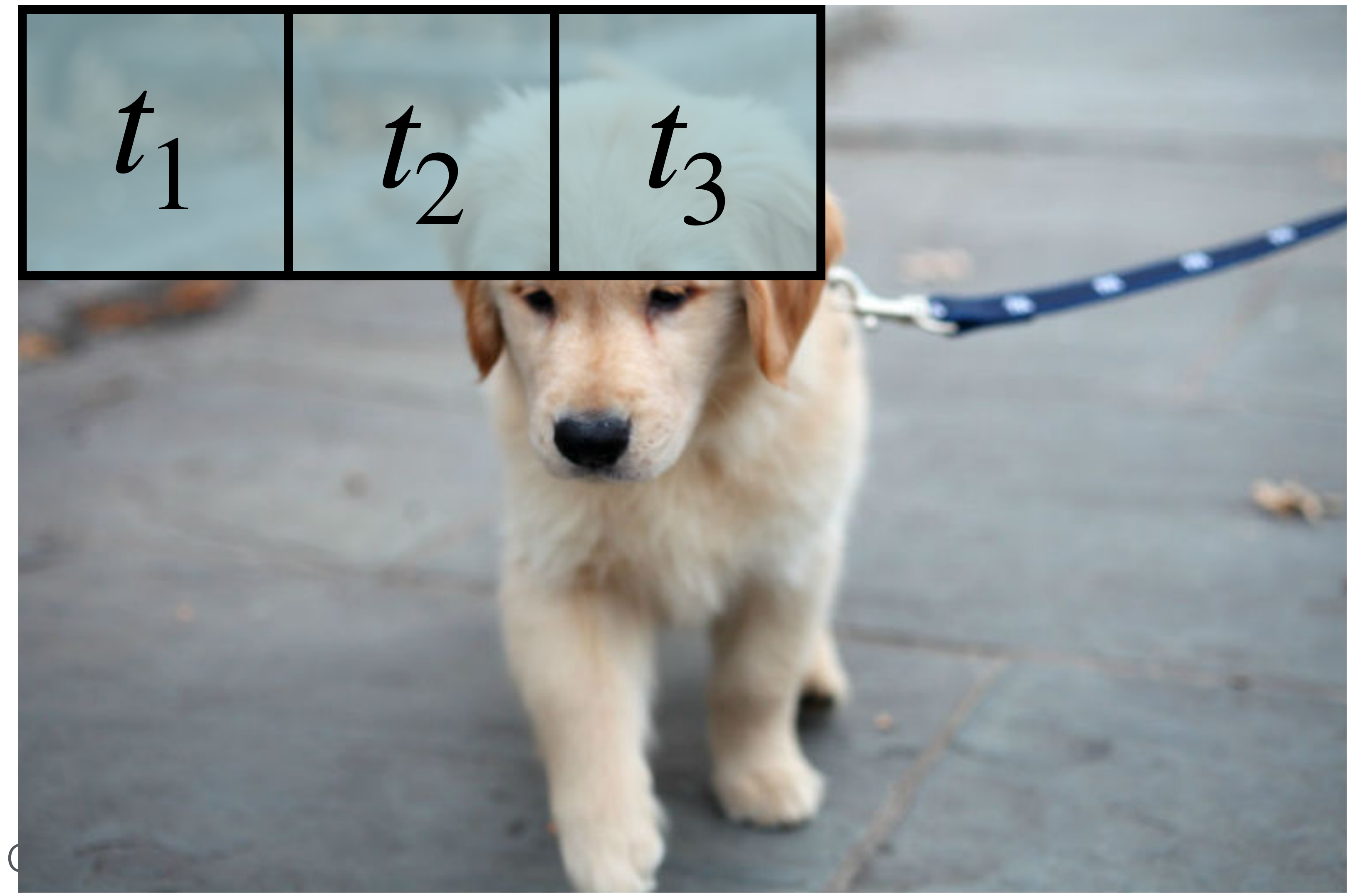
[1] WaveNet: A Generative Model for Raw Audio. Aaron van den Oord, et al. 2016
[2] Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. Songwei Ge, et al. 2022

Recap: Vector Quantization

- Variational Auto-Encoder
  - Decoder $P_D(x|z)$ Encoder $Q(z|x)$
- Vector Quantizer
  - $q(z) = \arg\min_{e_k} ||z - e_k||$
  - Learn codebook $\{e_1 \dots e_K\}$
  - What is $\nabla q(z)$?

Vanilla auto-regressive model





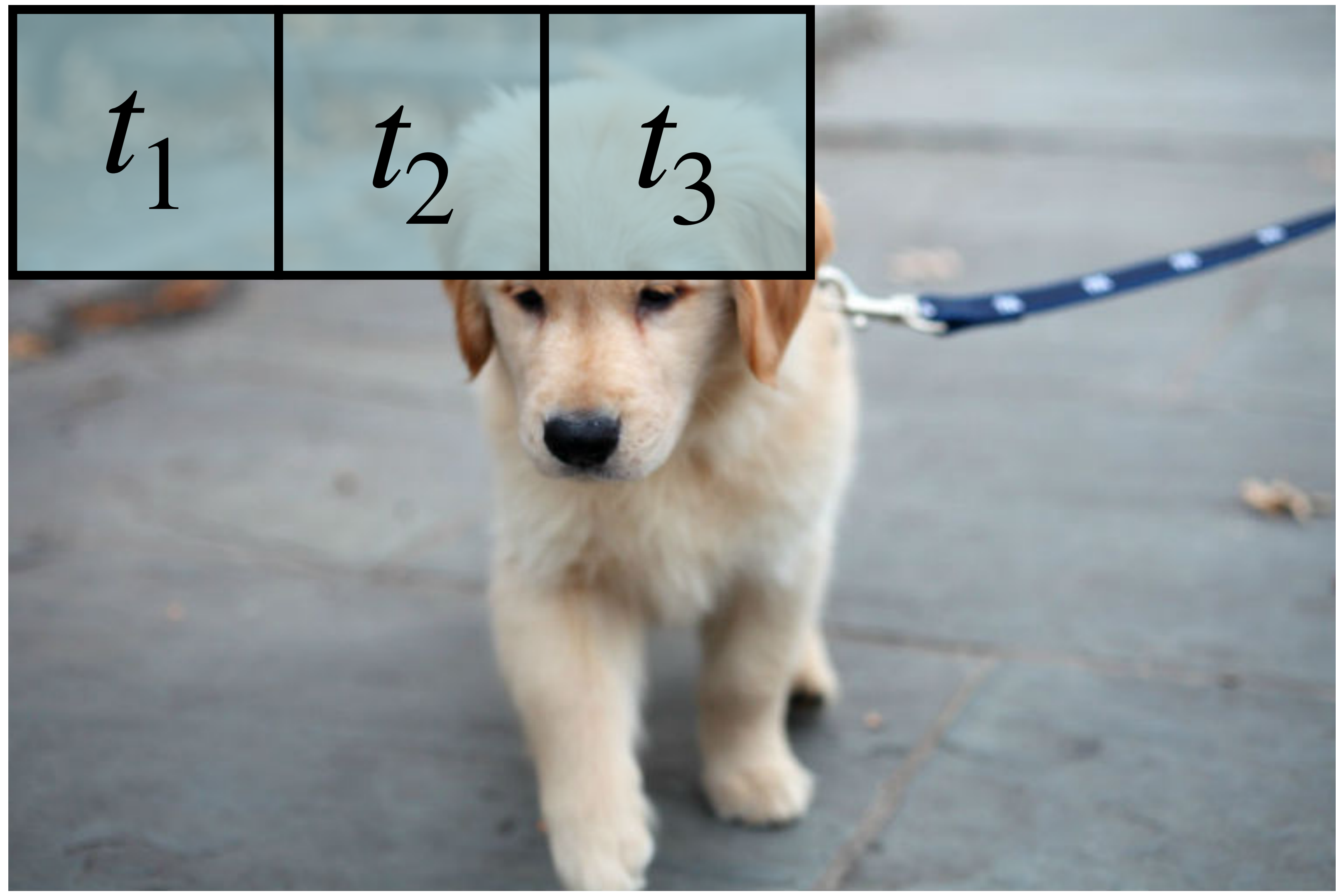
Tokenized auto-regressive model

[1] Neural Discrete Representation Learning. Aaron van den Oord, et al. 2017
[2] Language models are unsupervised multitask learners. Alec Radford, et al. 2019
[3] Simulating 500 million years of evolution with a language model. Thomas Hayes, et

Recap: Tokenization

- Image [1]
  - Convert patch p_i of pixels into token $t_i \in \{1, \ldots, K\}$
- Text [2]
  - Convert set of characters into token
- Protein-sequence [3]
  - Convert local protein structure to token

Vanilla auto-regressive model



Tokenized auto-regressive model

[1] Neural Discrete Representation Learning. Aaron van den Oord, et al. 2017
[2] Language models are unsupervised multitask learners. Alec Radford, et al. 2019
[3] Simulating 500 million years of evolution with a language model. Thomas Hayes, et al. 2024

Tokenization

A different view

- Convert
  - images ↔ streams of tokens
  - text ↔ streams of tokens
    - More in next section

$\leftrightarrow \boxed{t_1}\boxed{t_2}\boxed{t_3}$

A cute little dog fully focused on walking $\leftrightarrow \boxed{\hat{t}_1}\boxed{\hat{t}_2}\boxed{\hat{t}_3}$

DALL-E

$\leftrightarrow$

t_1	t_2	t_3

- **Let's learn a generative model over text and image tokens**
  - $P(\mathbf{t} \mid \hat{\mathbf{t}}) = P(t_1 \mid \hat{\mathbf{t}})P(t_2 \mid t_1, \hat{\mathbf{t}})\ldots P(t_L \mid t_1, \ldots, t_{L-1}, \hat{\mathbf{t}})$
- Q: What would we need to get this to work?

A cute little dog fully focused on walking

$\leftrightarrow$

$\hat{t}_1$	$\hat{t}_2$	$\hat{t}_3$

Zero-Shot Text-to-Image Generation, Ramesh et al. 2021

DALL-E

- Let's learn a generative model over text and image tokens
  - $P(\mathbf{t} \mid \hat{\mathbf{t}}) = P(t_1 \mid \hat{\mathbf{t}})P(t_2 \mid t_1, \hat{\mathbf{t}})\ldots P(t_L \mid t_1, \ldots, t_{L-1}, \hat{\mathbf{t}})$
- Where do we get image-text data from?
- What architecture do we use?

$\leftrightarrow$ $\boxed{t_1}\boxed{t_2}\boxed{t_3}$

A cute little dog fully focused on walking $\leftrightarrow$ $\boxed{\hat{t}_1}\boxed{\hat{t}_2}\boxed{\hat{t}_3}$

Zero-Shot Text-to-Image Generation, Ramesh et al. 2021

DALL-E

Dataset

- Image captioning dataset
  - Conceptual Captions [1]
    - 3.3 million text-image
  - OpenAI Internal data (the internet)
    - 250 million text-images pairs
    - YFCC100M [2]
  - Lots of cleanup

IMG_9793: Streetcar (Toronto Transit) by Andy Nystrom

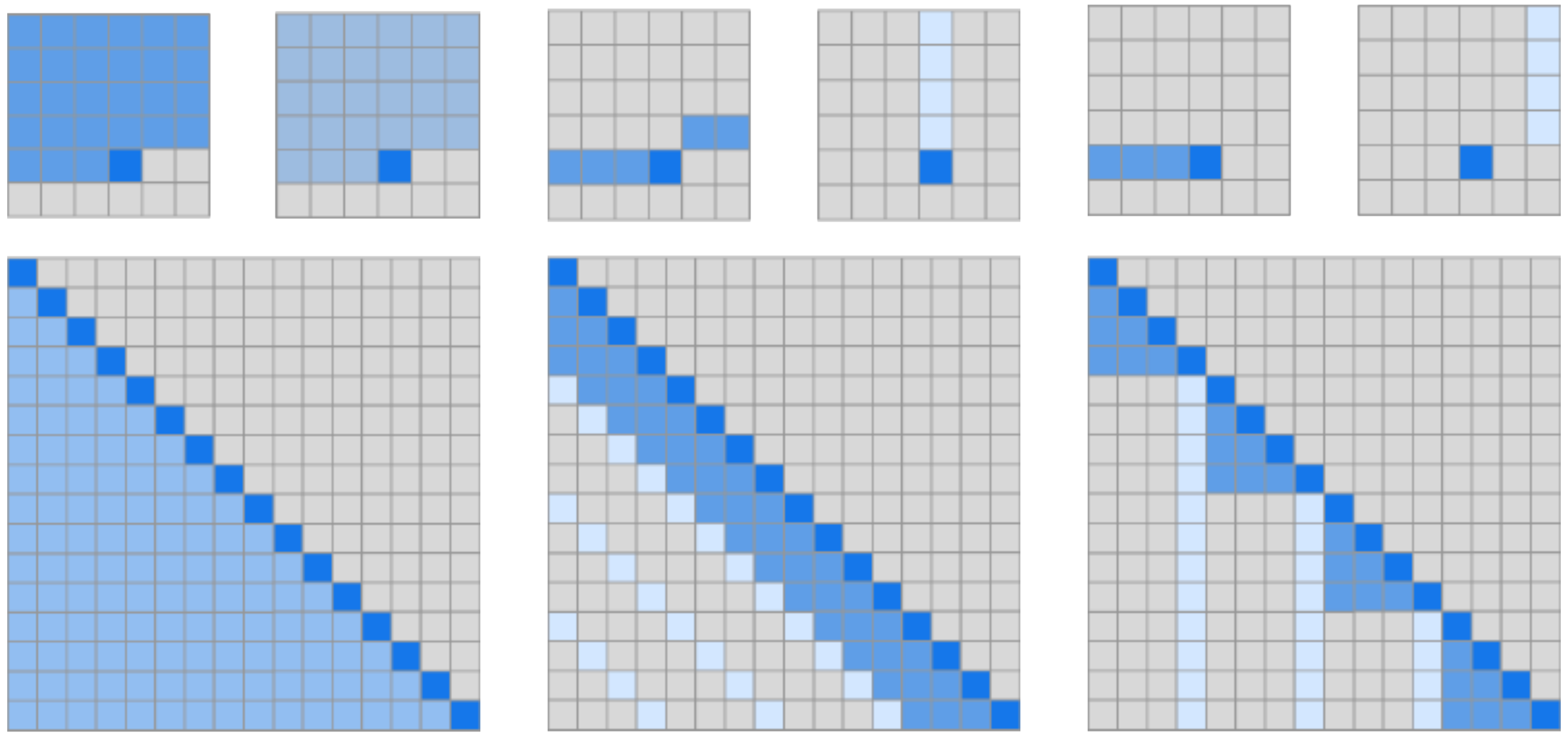
Celebrating our 6th wedding anniversary in Villa Mary by Rita & Tomek

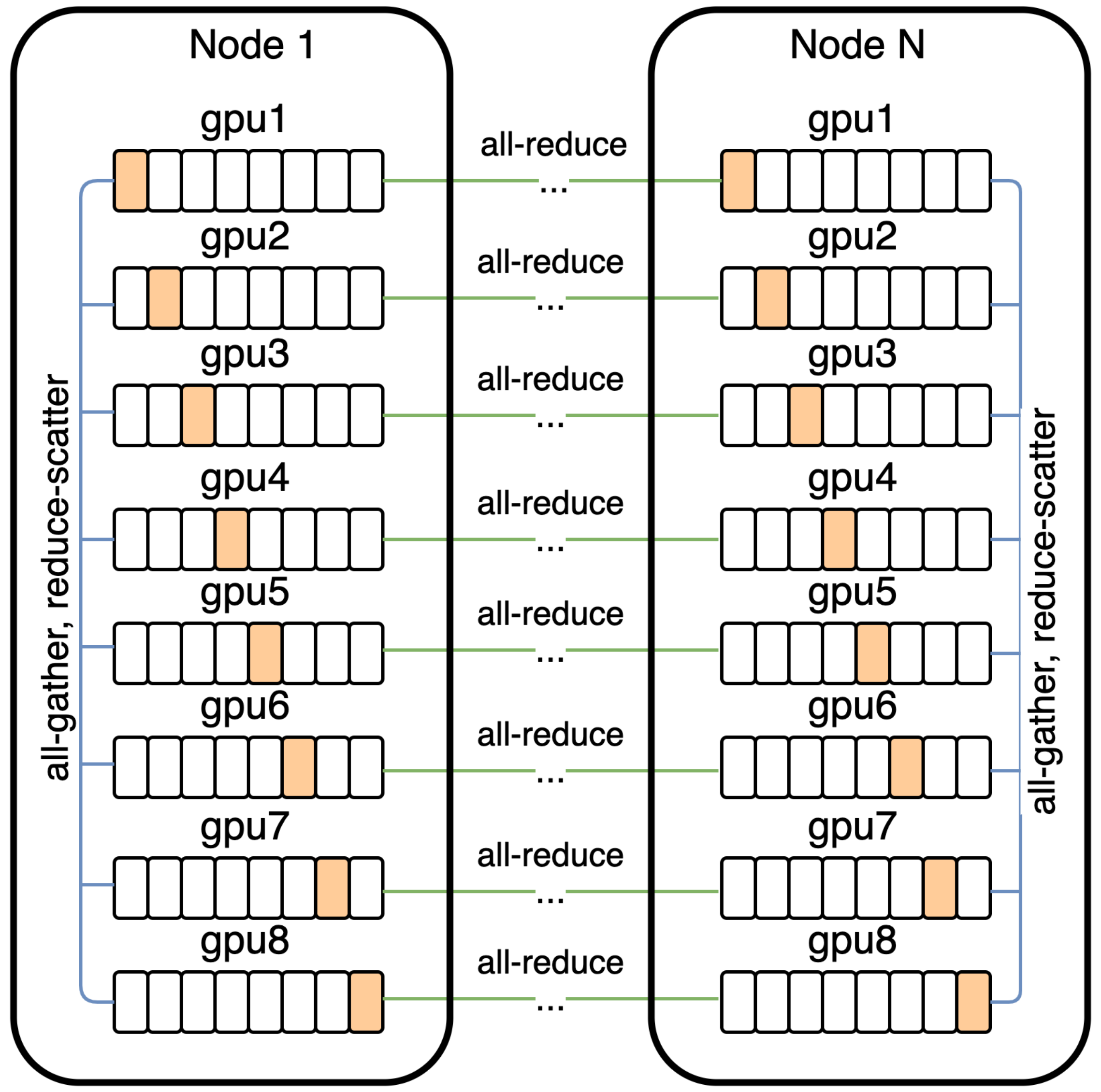
[1] Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning, Sharma et al. 2018
[2] YFCC100M: The New Data in Multimedia Research, Thomee et al. 2015

DALL-E

Architecture

- Sparse transformer [1]
- Mixed-precision training
- Sharded Multi-GPU training
    - Pre-cursor to FSDP





[1] Generating Long Sequences with Sparse Transformers, Child et al. 2019

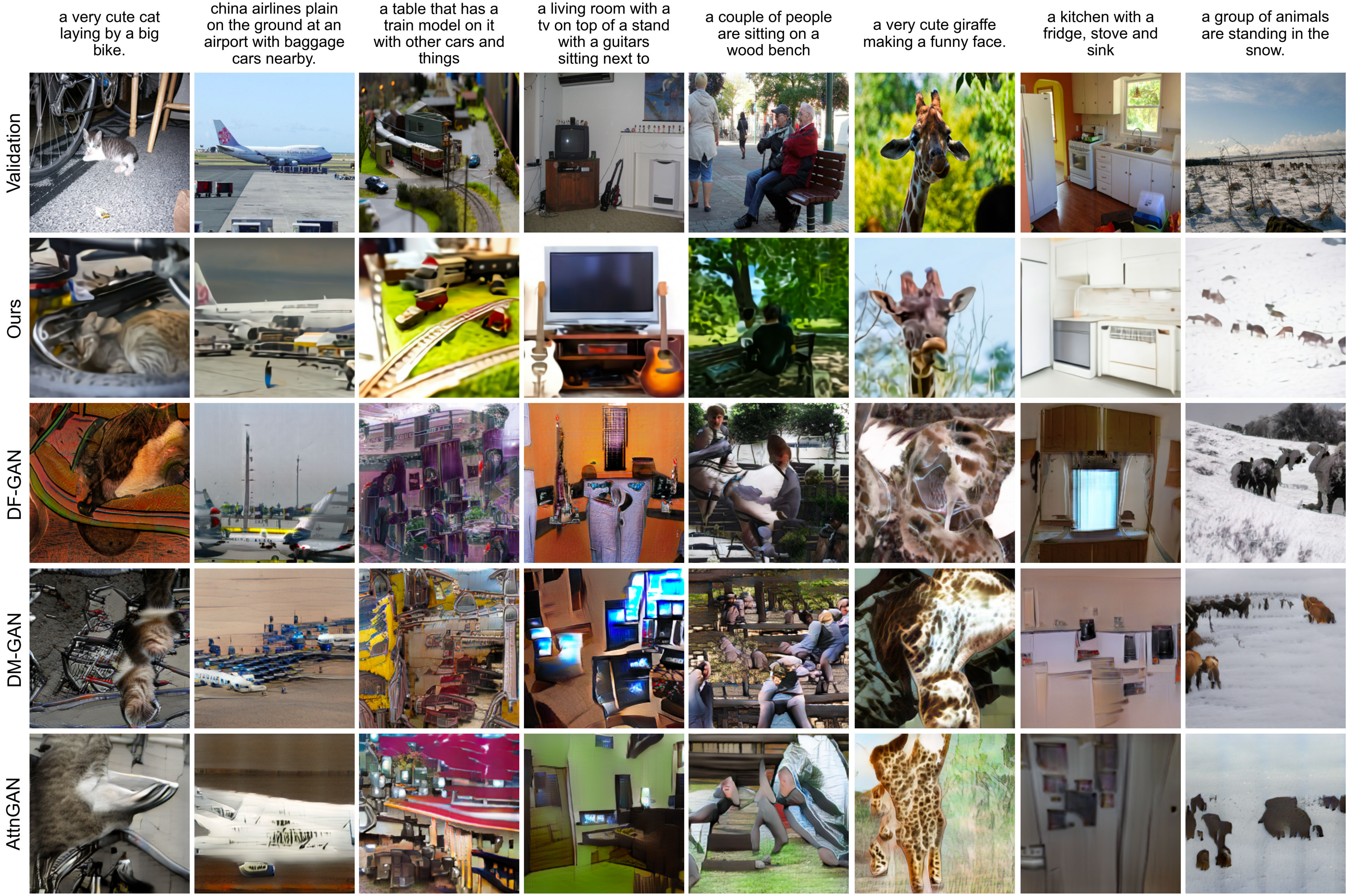
DALL-E

Results

a tapir made of accordion. a tapir with the texture of an accordion.

an illustration of a baby hedgehog in a christmas sweater walking a dog

a neon sign that reads “backprop”. a neon sign that reads “backprop”. backprop neon sign


a very cute cat laying by a big bike.
china airlines plain on the ground at an airport with baggage cars nearby.
a table that has a train model on it with other cars and things
a living room with a tv on top of a stand with a guitars sitting next to
a couple of people are sitting on a wood bench
a very cute giraffe making a funny face.
a kitchen with a fridge, stove and sink
a group of animals are standing in the snow.
Validation
Ours
DF-GAN
DM-GAN
AttnGAN

DALL-E

Lessons learned

- Data is king
- Scaling matters
- Models can be simple

$\leftrightarrow$ $\boxed{t_1}\boxed{t_2}\boxed{t_3}$

A cute little dog fully focused on walking $\leftrightarrow$ $\boxed{\hat{t}_1}\boxed{\hat{t}_2}\boxed{\hat{t}_3}$

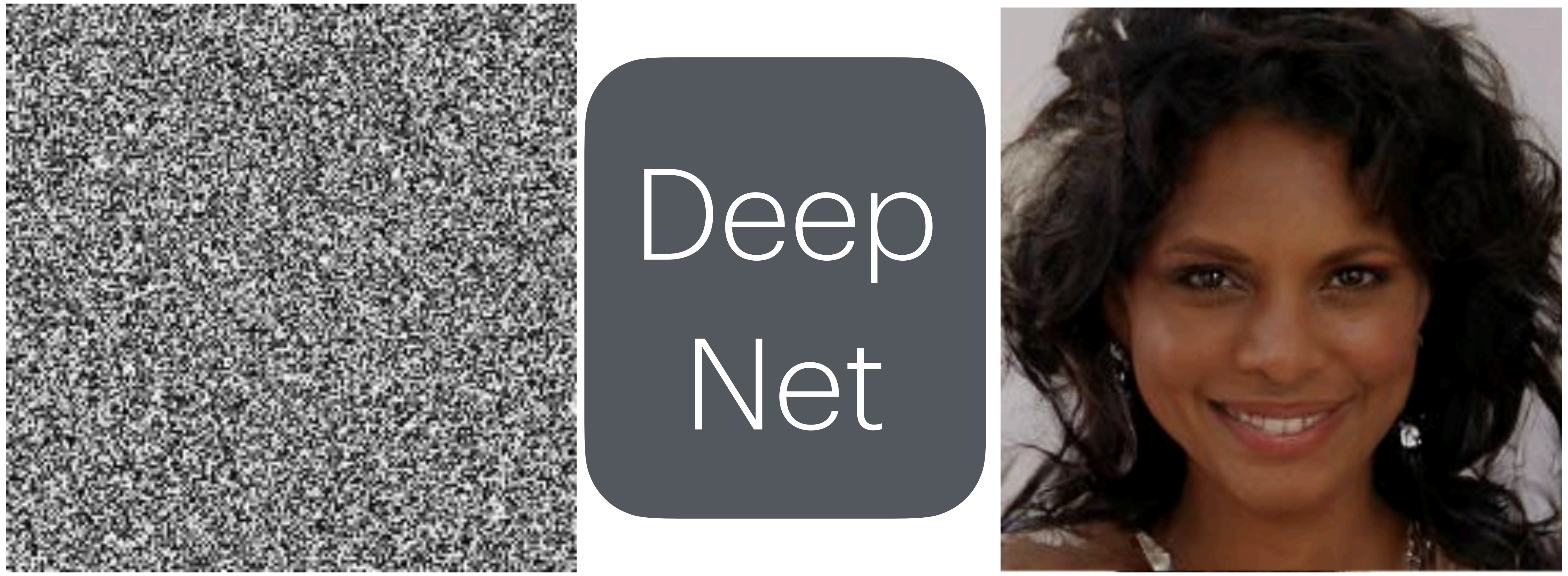
Zero-Shot Text-to-Image Generation, Ramesh et al. 2021

Diffusion

Philipp Krähenbühl, UT Austin

Main idea

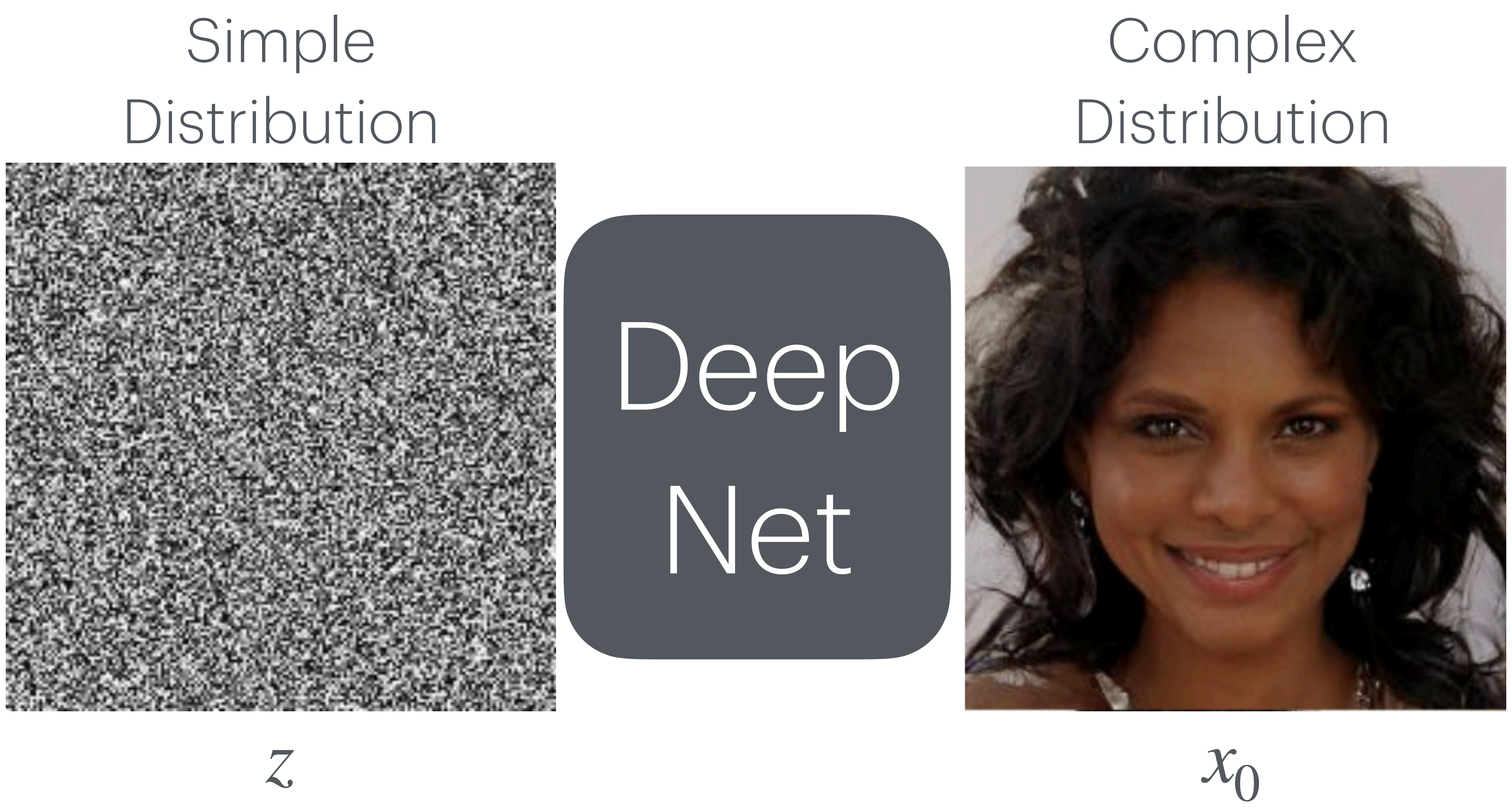
- Learn to transform random noise to images
- Why?



z x_0

Main idea

- Learn to transform random noise to images
  - Random noise: Simple distribution, easy to sample from
  - Images: Complex distribution

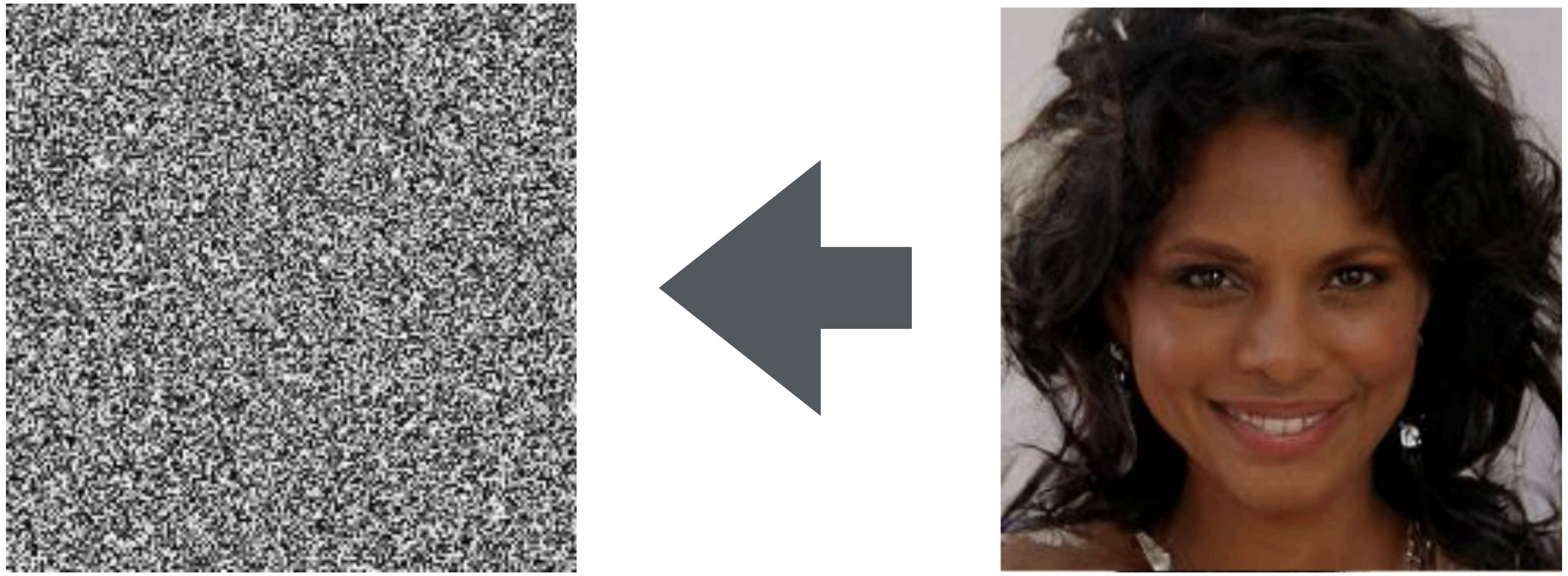


Diffusion Process

- Can we build a model from real images to noise?

Simple Distribution

Complex Distribution



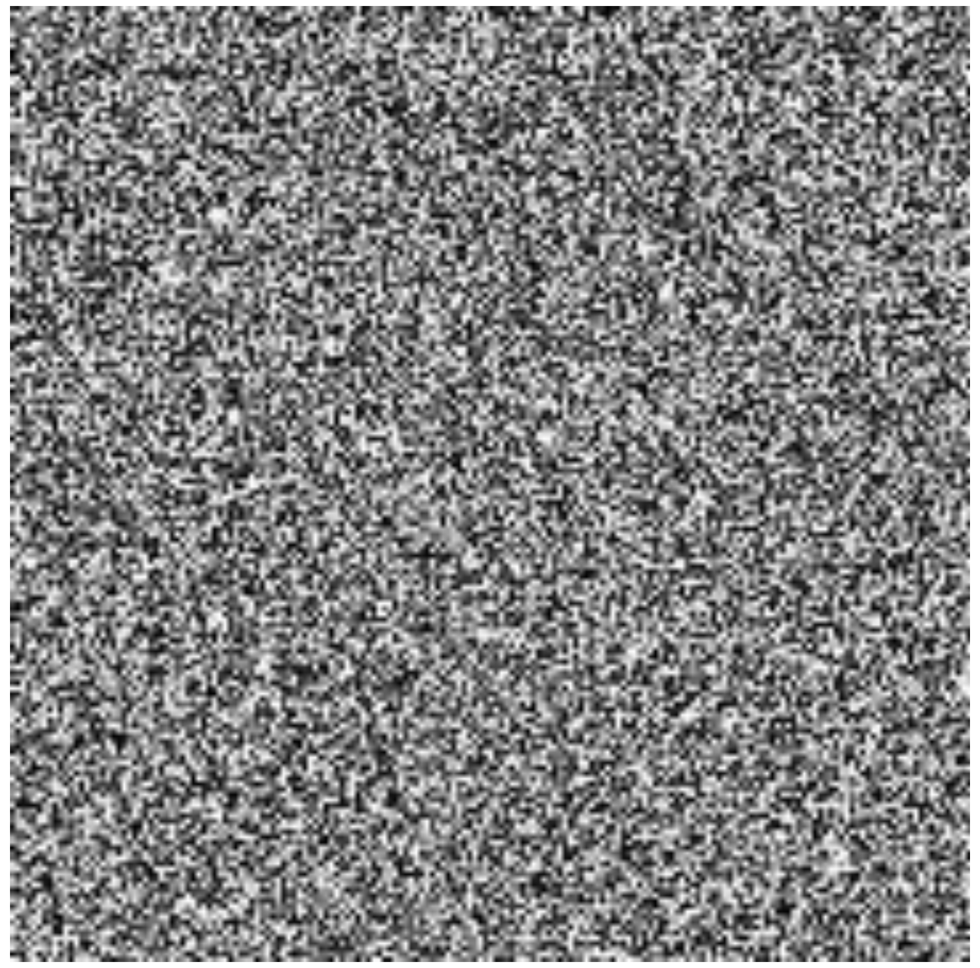
z

x_0

Diffusion Process

- Can we build a model from real images to noise?

Simple Distribution



z

Complex Distribution

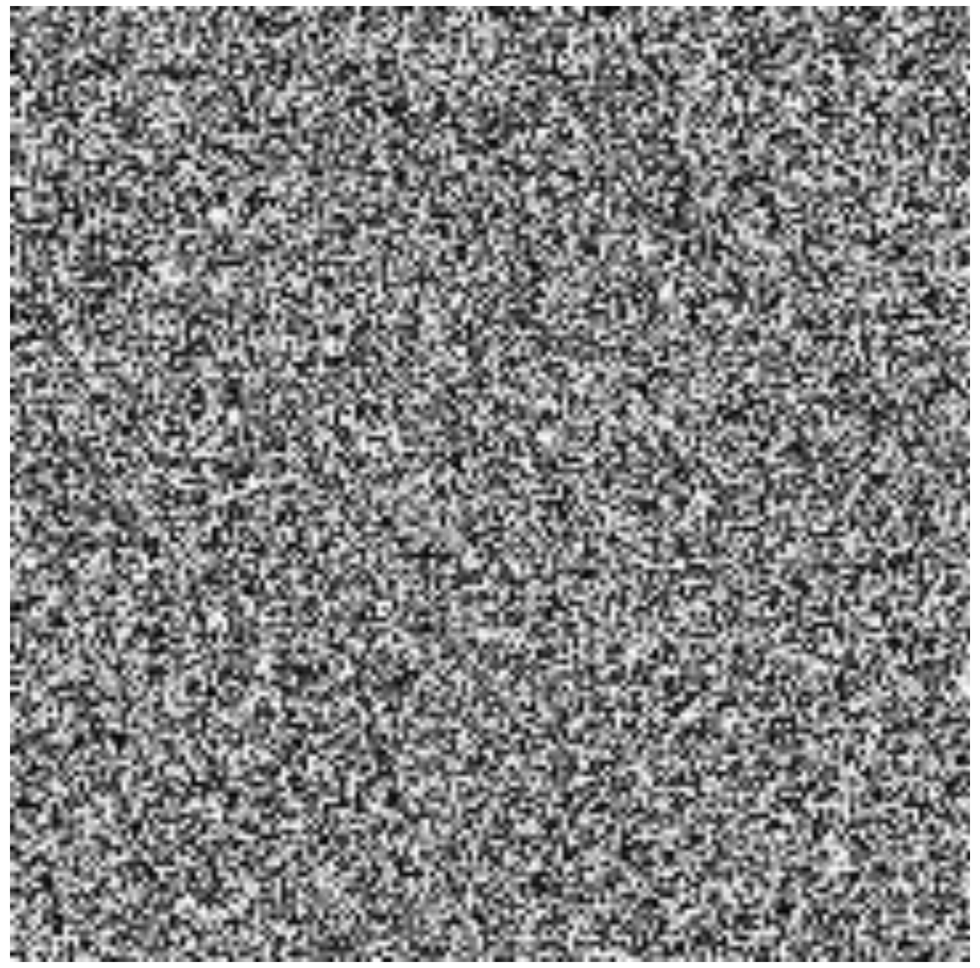
x_0

```
def f(x):
    return torch.randn(x.shape)
```

Diffusion Process

- Lets revert this

Simple Distribution



z

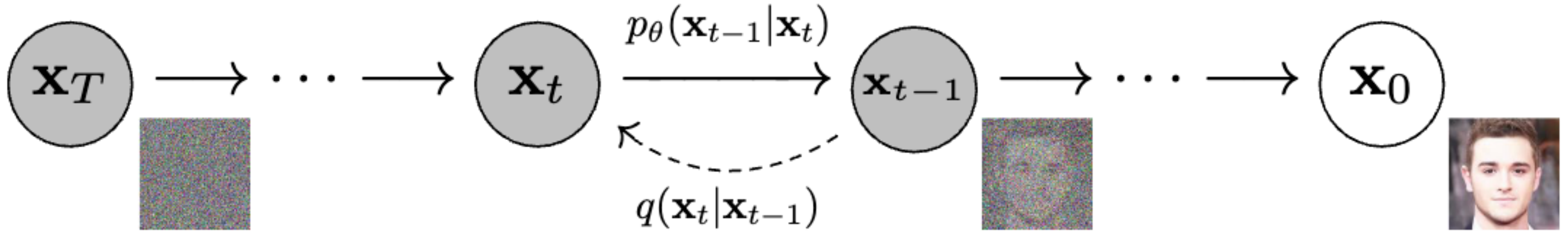
Complex Distribution

x_0

```
def f(x):
    return torch.randn(x.shape)
```

Diffusion Process

Forward process



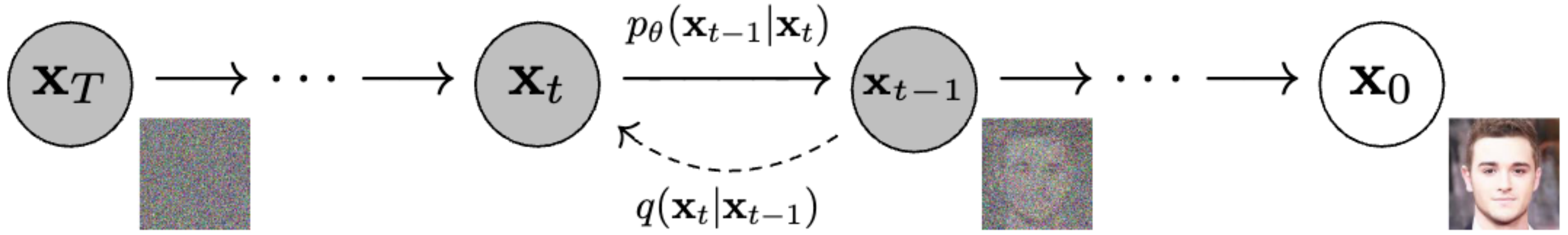
- Make an image noisy
  - Start with an image x_0
  - Add noise $q(x_t | x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} x_t, \beta_t I)$
  - β_t increases linearly with t
- $$\begin{aligned} q(x_t | x_0) &= \int \prod_{i=1}^{t} q(x_i | x_{i-1}) dx_{1\ldots t-1} \\ &= \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I) \end{aligned}$$

  where $\bar{\alpha}_t = \prod_{i=1}^{t} (1-\beta_i)$

Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Process

Reverse process



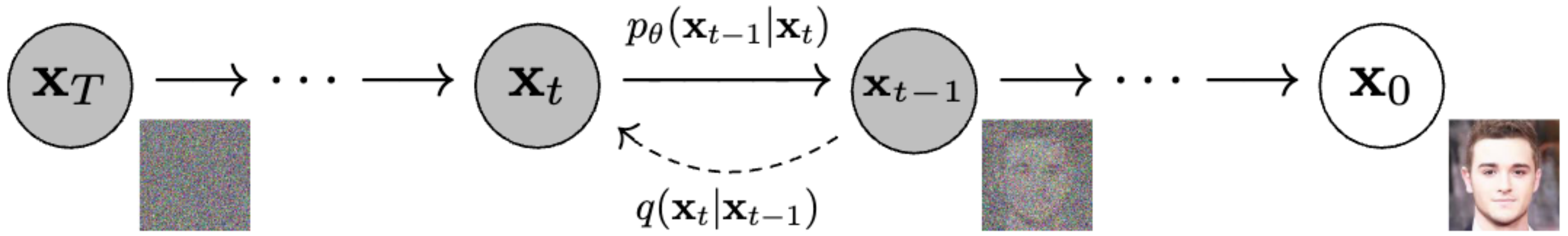
- Learn to predict image progressively
  - Start $P(x_T) = \mathcal{N}(0, I)$
  - Denoise $P(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$
- Reverse process

$$P(x_{0\ldots T}) = P(x_T) \prod_{t=1}^{T} P(x_{t-1} \mid x_t)$$

- $P(x_0) = \int P(x_{0\ldots T}) dx_{1\ldots T}$

Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Process



- Maximize Evidence lower bound (ELBO)

$$\log P(x_0) \geq E_q \left[\log \frac{P(x_{0\ldots T})}{q(x_{1\ldots T} | x_0)} \right]$$

- (Lot's of math later)

- Relatively simply training and sampling algorithms
  - $\epsilon(x_t, t)$ is a noise-prediction network

Algorithm 1 Training

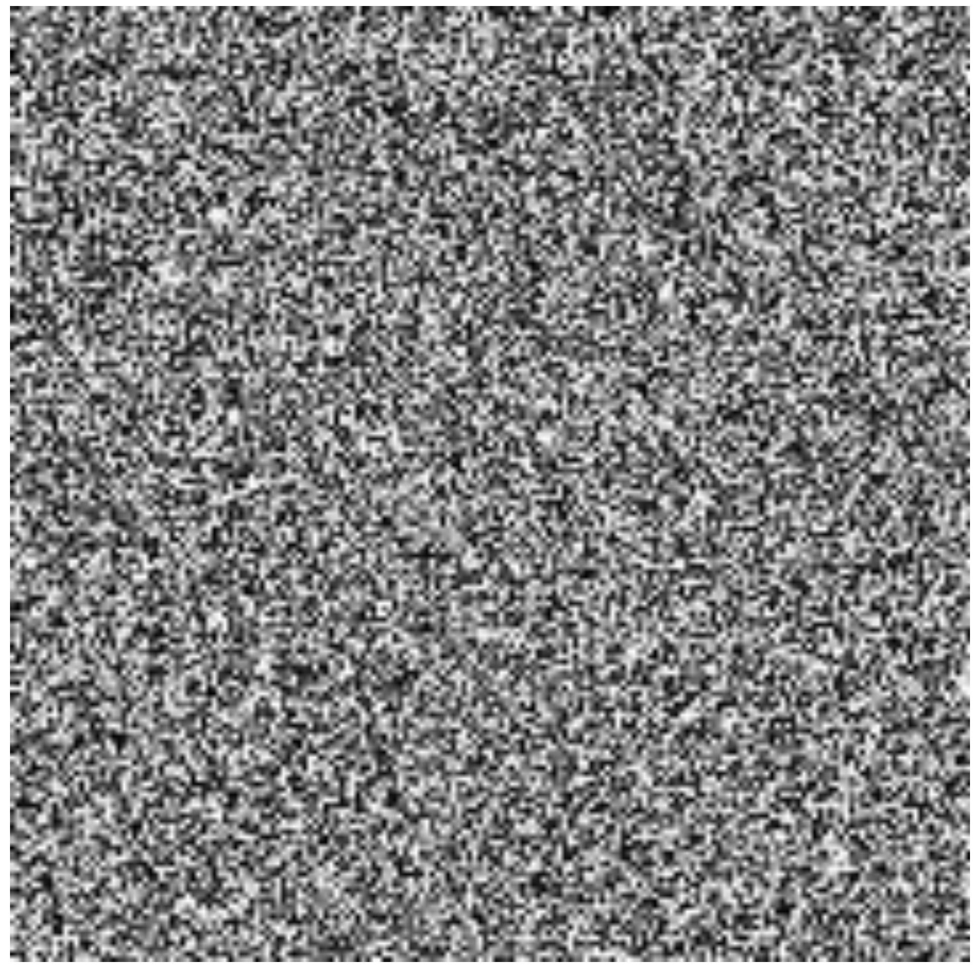
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Model

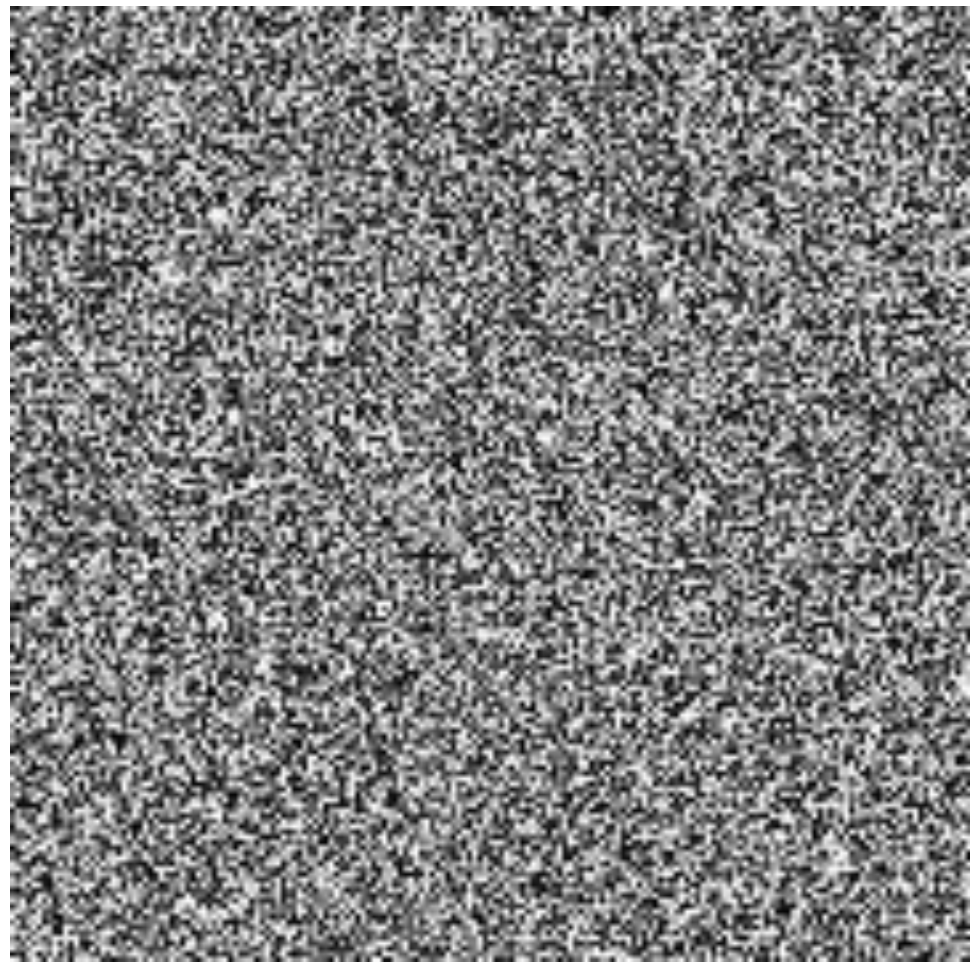


U-Net

Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Model

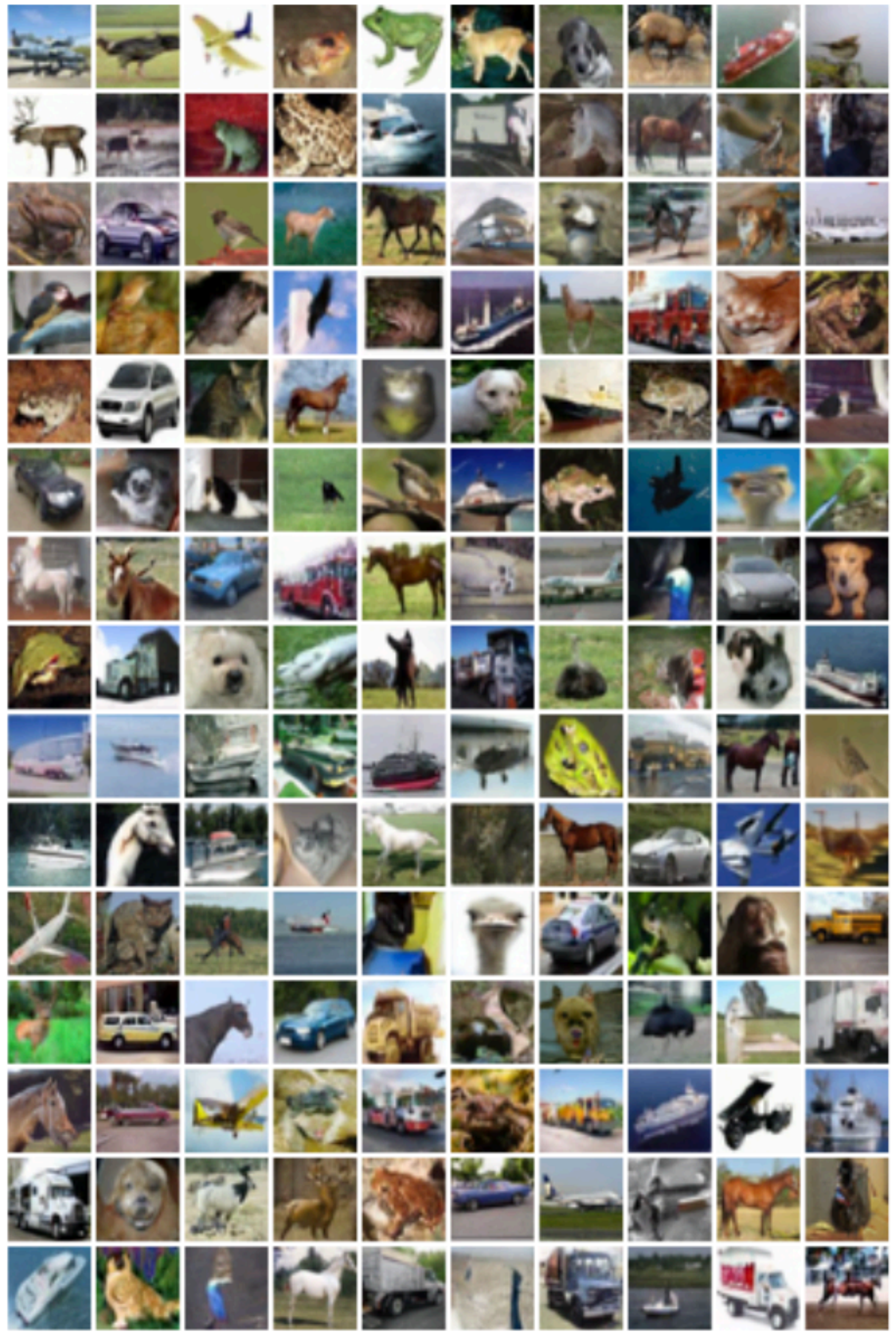
Nowadays



Transformer
(DiT)

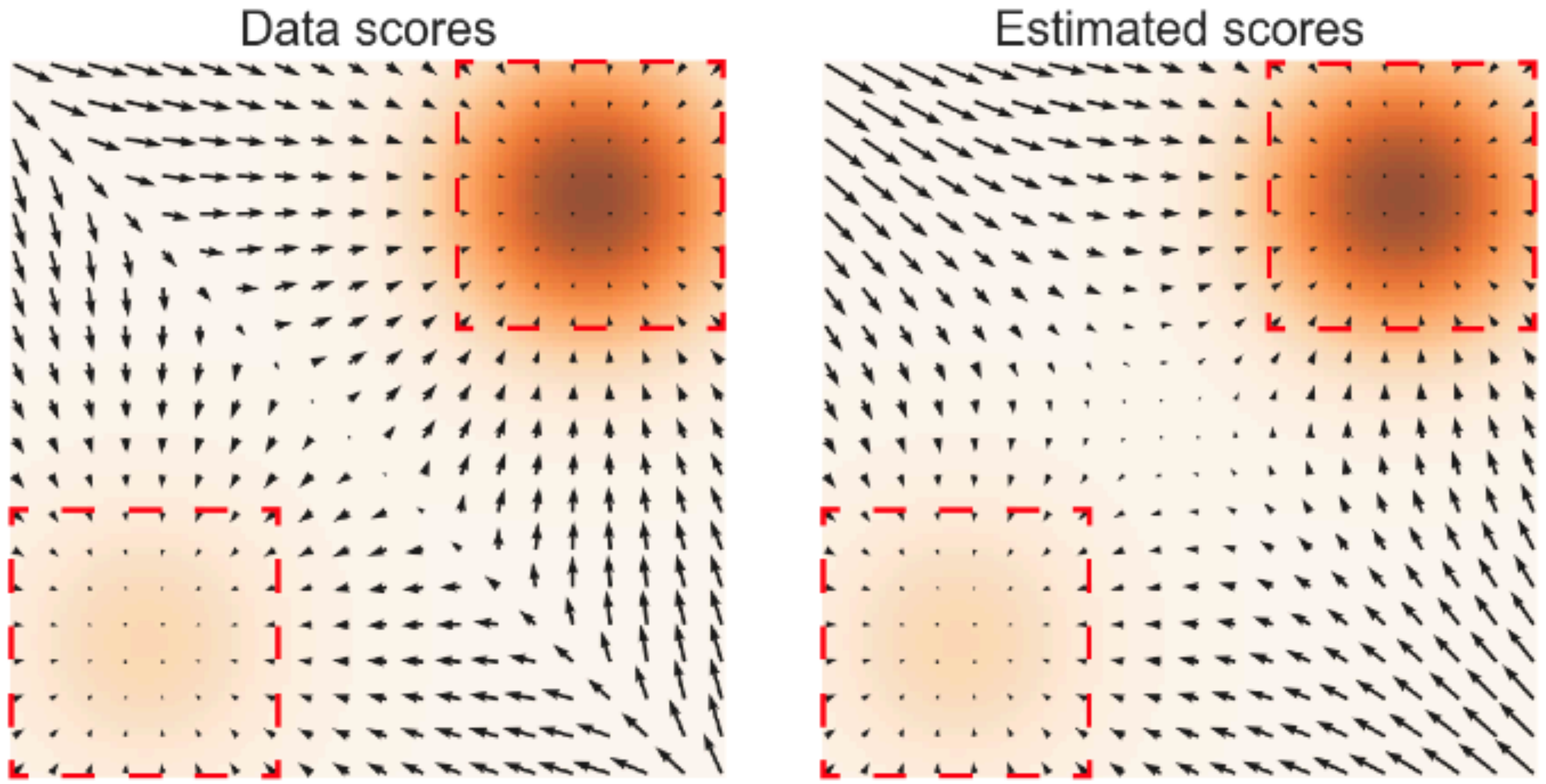
Scalable Diffusion Models with Transformers, Peebles and Xie 2023

Diffusion - Results



Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Score-based models



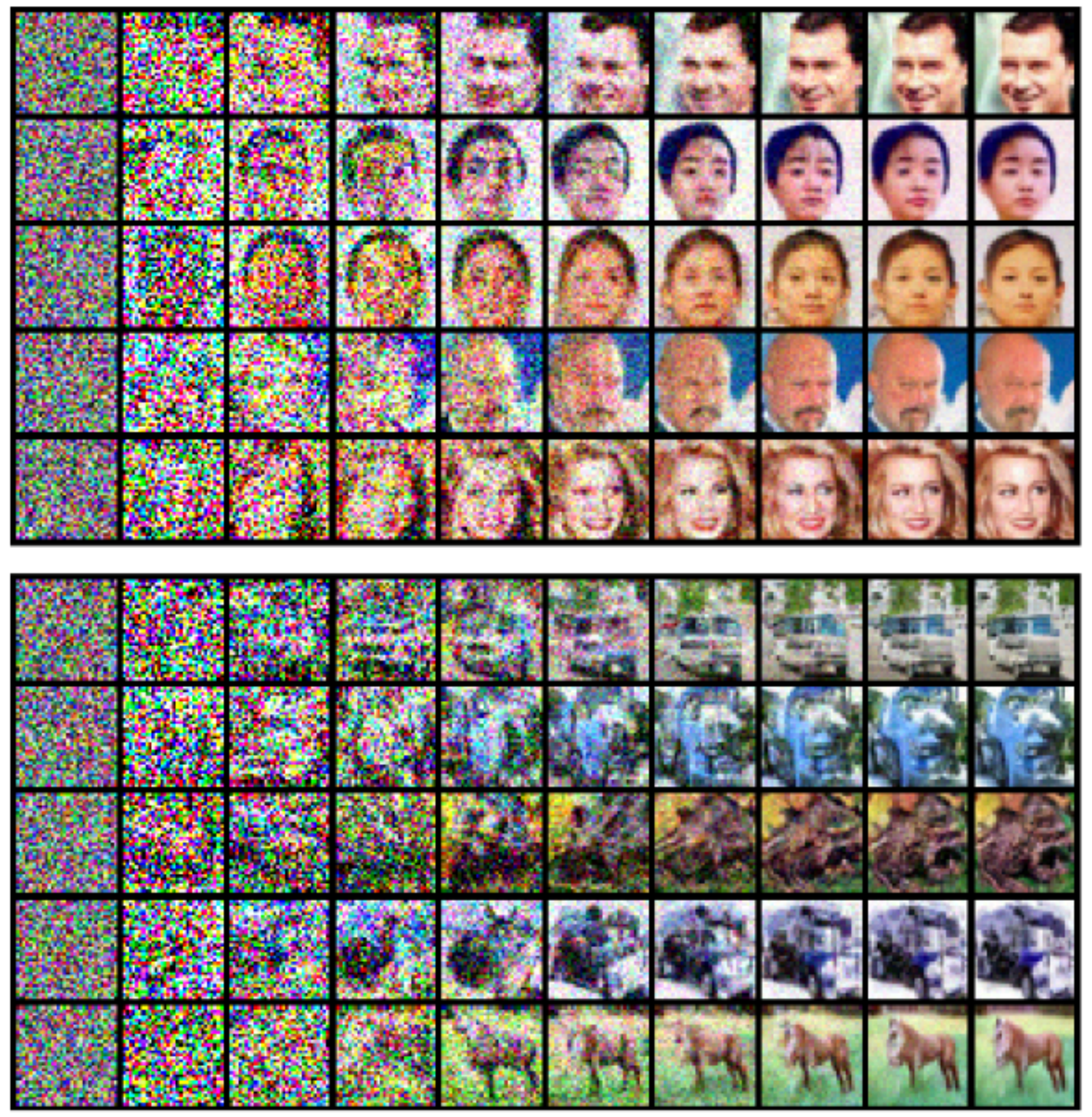
- $P(x)$ / $\log P(x)$
  - is hard to learn
- $\nabla \log P(x)$ (score function)
  - is easier to learn/estimate

$$E_{x \sim P}\left[\left|s(x) - \nabla P(x)\right|^2\right] =$$

- $$E_{x \sim P}\left[\mathrm{tr}(\nabla_x s(x)) + \frac{1}{2}\left|s(x)\right|^2\right] + \mathrm{const}$$

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.
Initialize $\tilde{\mathbf{x}}_0$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ ▷ α_i is the step size.
  for $t \leftarrow 1$ to T **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
  end for
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
end for
return $\tilde{\mathbf{x}}_T$



Generative Modeling by Estimating Gradients of the Data Distribution. Yang Song, et al. 2019.

Guided Diffusion

- Learn variance $\Sigma(x_t)$
- Better architecture
  - Deeper, more attention heads, attention on multiple blocks, …
- Classifier guidance (conditioning)

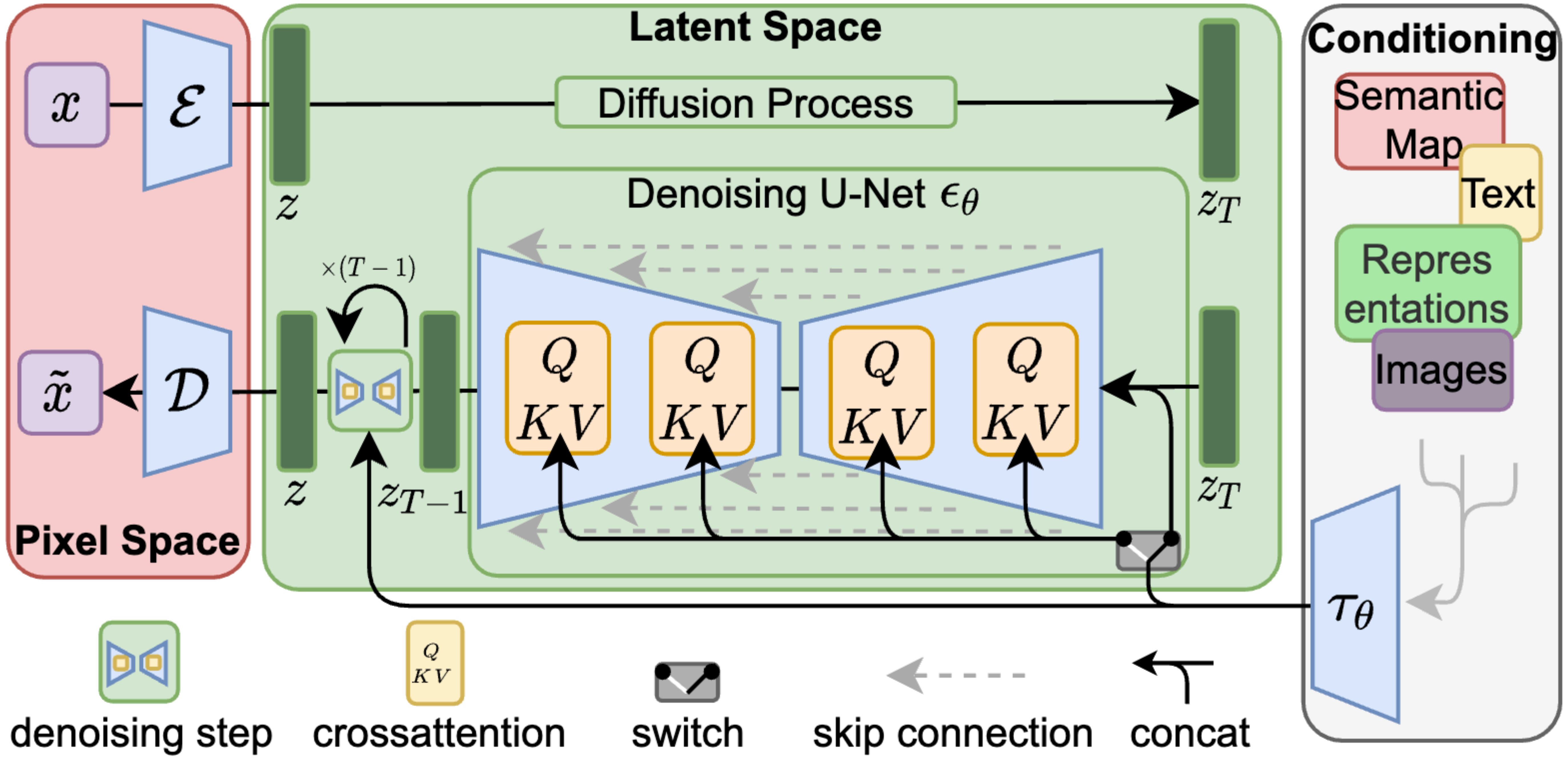
Diffusion Models Beat GANs on Image Synthesis. Prafulla Dhariwal, et al. 2021.

Diffusion

- Very good image quality
- Not easily controllable
- Computationally quite expensive
  - Multiple sampling steps
  - Fairly high resolution inputs and outputs required (original image size)

Latent Diffusion

- Auto-encoder + Diffusion
  - Similar to VQVAE + Auto-regressive
- Speeds up training and generation
  - Lower resolution diffusion
  - Auto-encoders are fast
- Higher resolution outputs



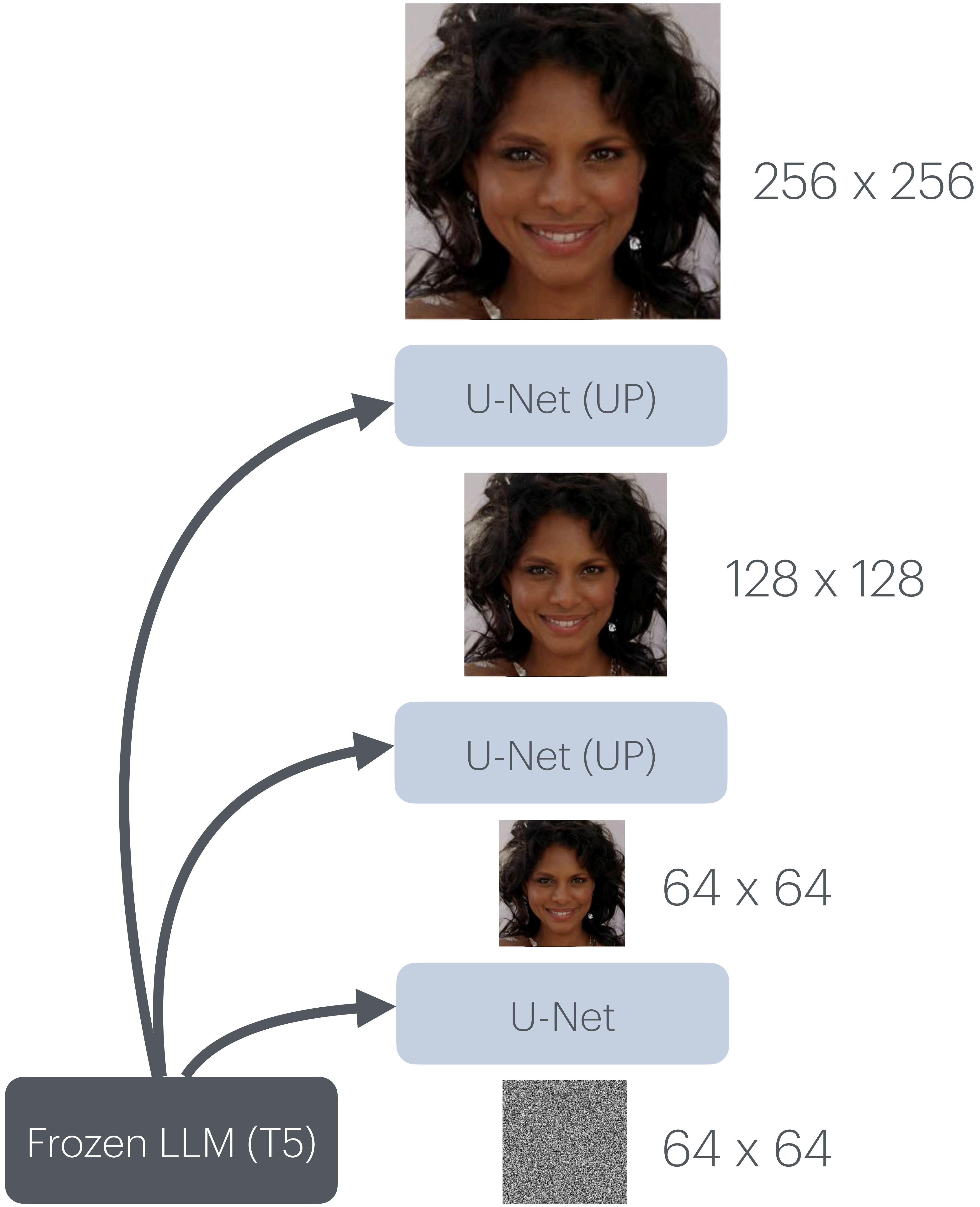
High-Resolution Image Synthesis with Latent Diffusion Models. Robin Rombach, et al. 2021.

Latent Diffusion

High-Resolution Image Synthesis with Latent Diffusion Models. Robin Rombach,, et al. 2021.

Imagen

- First really large scale diffusion model
  - 800M+ image-text pairs
- Frozen LLM
- Lower resolution diffusion 64x64
  - Upsampling to 1024



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

Imagen

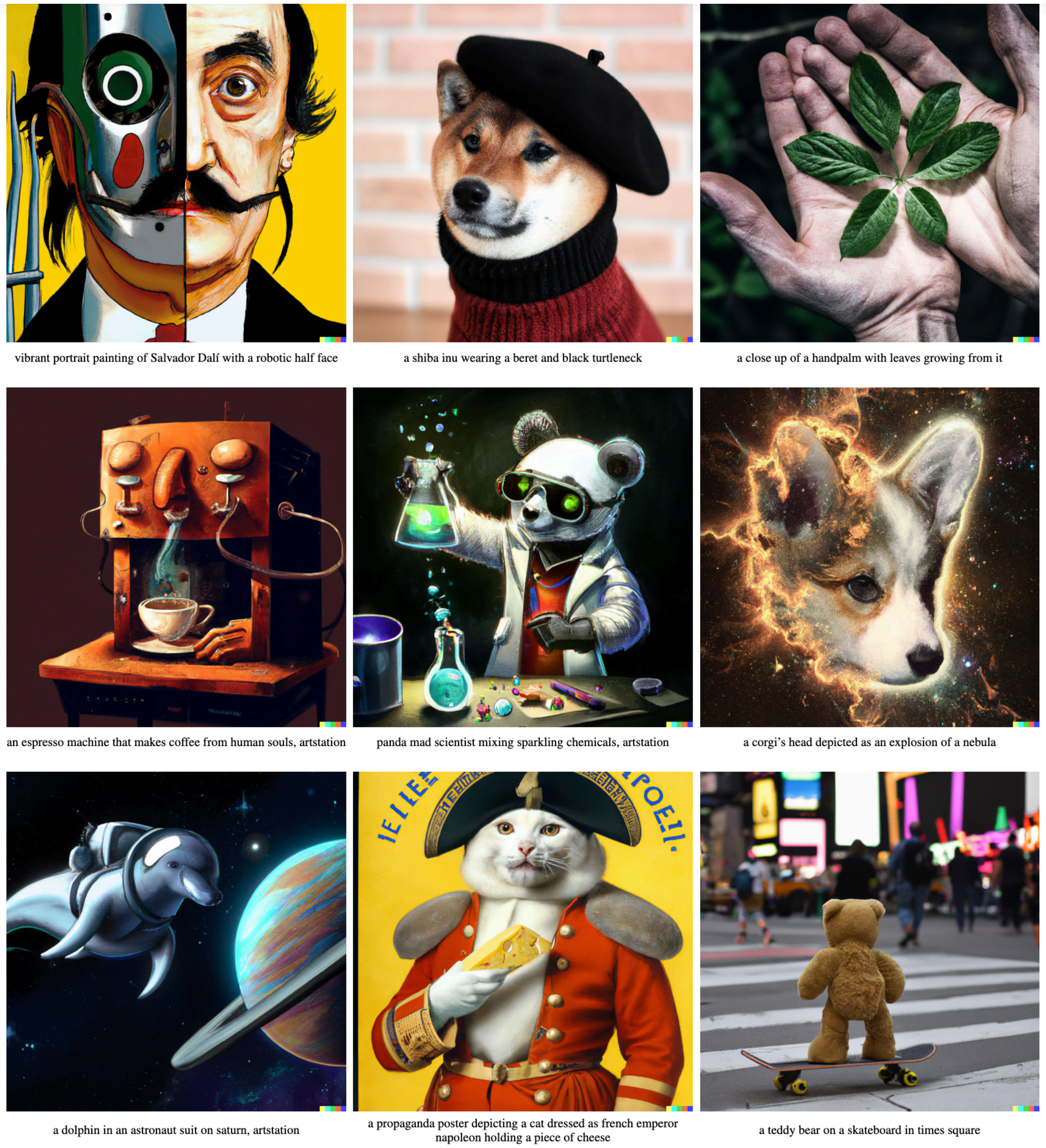
Results

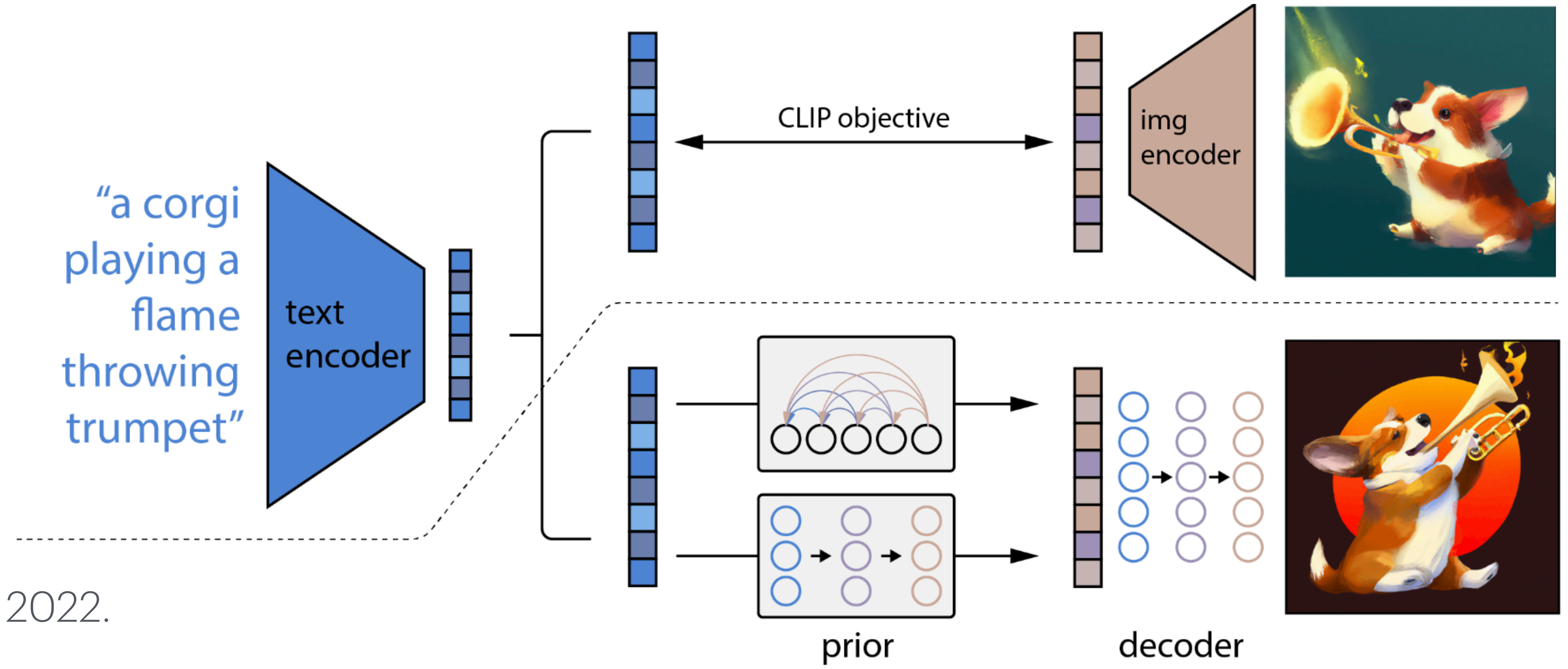
A chrome-plated duck with a golden beak arguing with an angry turtle in a forest

The Toronto skyline with Google brain logo written in fireworks.

DALL-E 2

- CLIP-LM conditioned diffusion
  - 64x64 results
  - Upsampling
    $64 \times 64 \rightarrow 256 \times 256 \rightarrow 1024 \times 1024$





Hierarchical Text-Conditional Image Generation with CLIP Latents. Aditya Ramesh, et al. 2022.

DALL-E 3

- Better data
  - Recaptioned dataset

A fierce garden gnome warrior, clad in armor crafted from leaves and bark, brandishes a tiny sword and shield. He stands valiantly on a rock amidst a blooming garden, surrounded by colorful flowers and towering plants. A determined expression is painted on his face, ready to defend his garden kingdom.

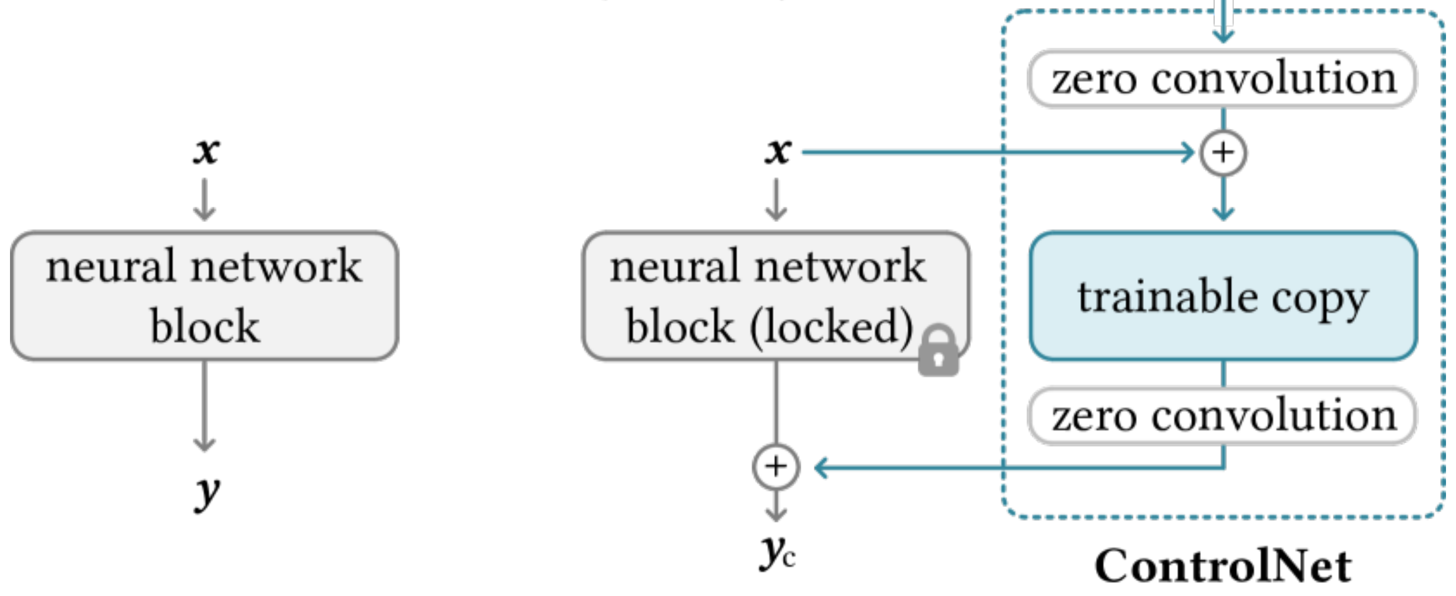
An icy landscape under a starlit sky, where a magnificent frozen waterfall flows over a cliff. In the center of the scene, a fire burns bright, its flames seemingly frozen in place, casting a shimmering glow on the surrounding ice and snow.

A swirling, multicolored portal emerges from the depths of an ocean of coffee, with waves of the rich liquid gently rippling outward. The portal engulfs a coffee cup, which serves as a gateway to a fantastical dimension. The surrounding digital art landscape reflects the colors of the portal, creating an alluring scene of endless possibilities.

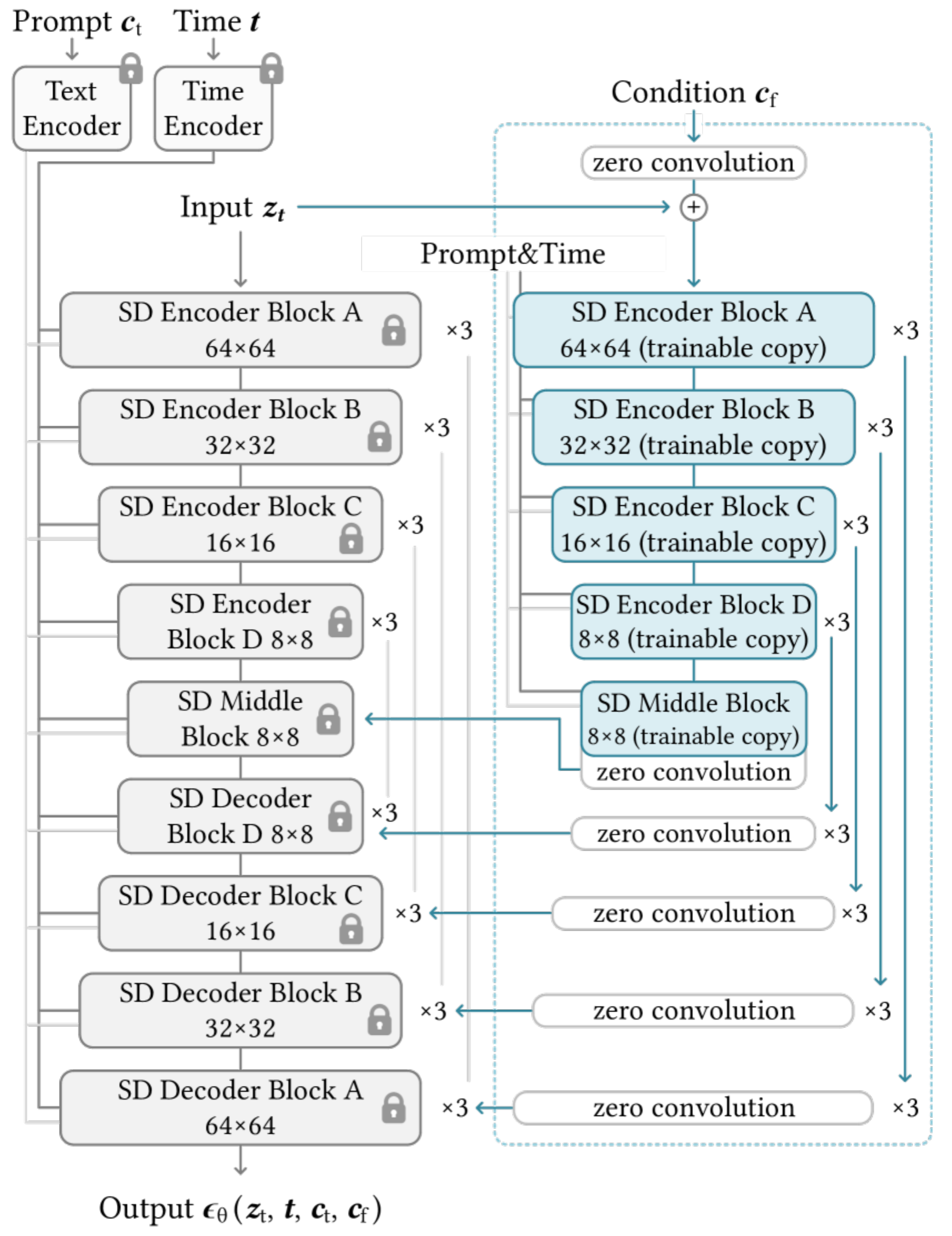
Improving Image Generation with Better Captions. James Betker, et al. 2023.

ControlNet

- "Condition" on more than just text
- Start from pre-trained model
- Add copy of encoder
    - For additional input
    - Fuse with zero-initialized convolution



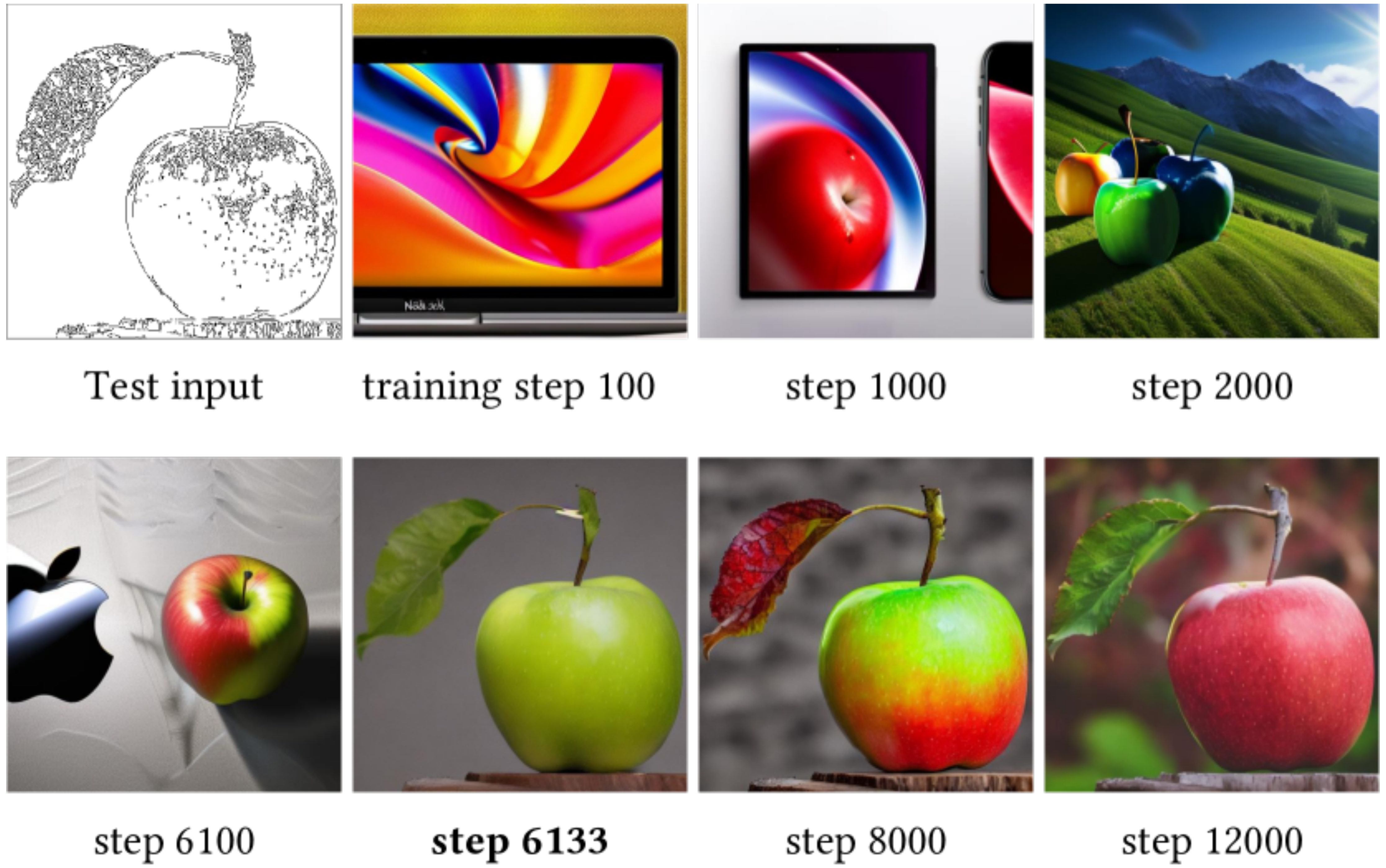
(a) Before (b) After



(a) Stable Diffusion (b) ControlNet

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

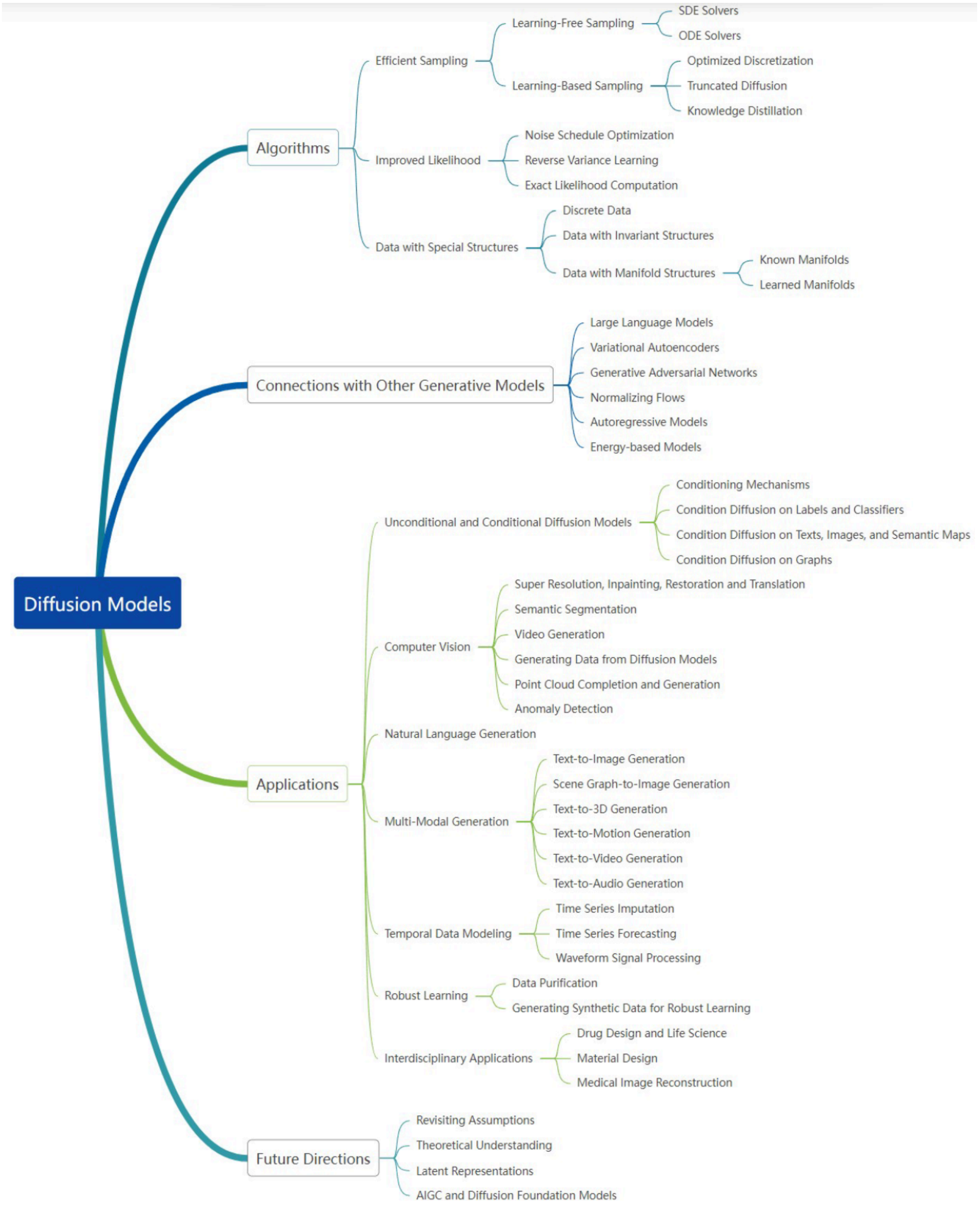
ControlNet



- Training objective: Denoise
  - Original image + noise
  - Contintioned on auto-generated edge detections, pose tracks, ...
- Trains quite quickly

Diffusion is a large field

- More efficient sampling
  - One step diffusion, ...
- More efficient architectures
- More efficient training
  - Noise schedules, variance learning, ..
- ...



One-step Diffusion with Distribution Matching Distillation. Tianwei Yin, et al. 2023.
Diffusion Models: A Comprehensive Survey of Methods and Applications. Ling Yang, et al. 2022.

References

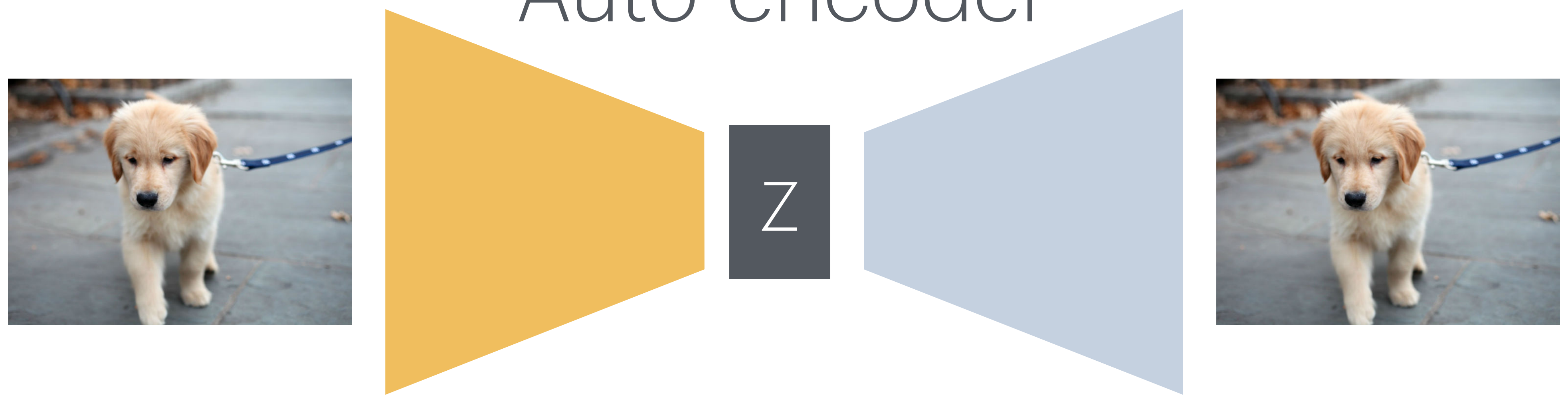
- Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.
- Generative Modeling by Estimating Gradients of the Data Distribution. Yang Song, et al. 2019.
- Diffusion Models Beat GANs on Image Synthesis. Prafulla Dhariwal, et al. 2021.
- High-Resolution Image Synthesis with Latent Diffusion Models. Robin Rombach, et al. 2021.
- Scalable Diffusion Models with Transformers, Peebles and Xie 2023
- Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. Chitwan Saharia, et al. 2022.
- Hierarchical Text-Conditional Image Generation with CLIP Latents. Aditya Ramesh, et al. 2022.
- Improving Image Generation with Better Captions. James Betker, et al. 2023.
- Adding Conditional Control to Text-to-Image Diffusion Models. Lvmin Zhang, et al. 2023.
- One-step Diffusion with Distribution Matching Distillation. Tianwei Yin, et al. 2023.
- Diffusion Models: A Comprehensive Survey of Methods and Applications. Ling Yang, et al. 2022.

Which Generative Model Should I Use?

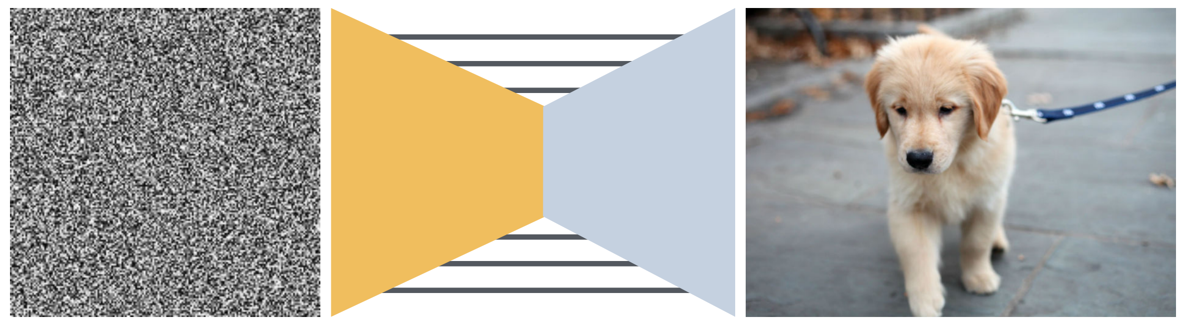
Philipp Krähenbühl, UT Austin

Recap

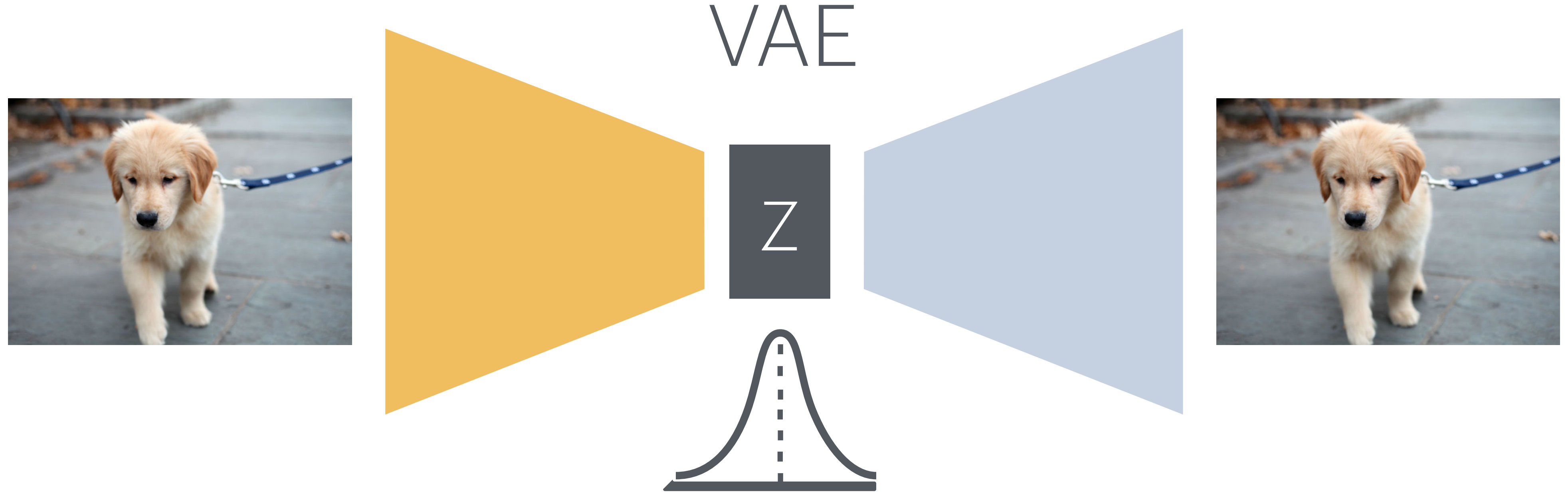
Auto-encoder



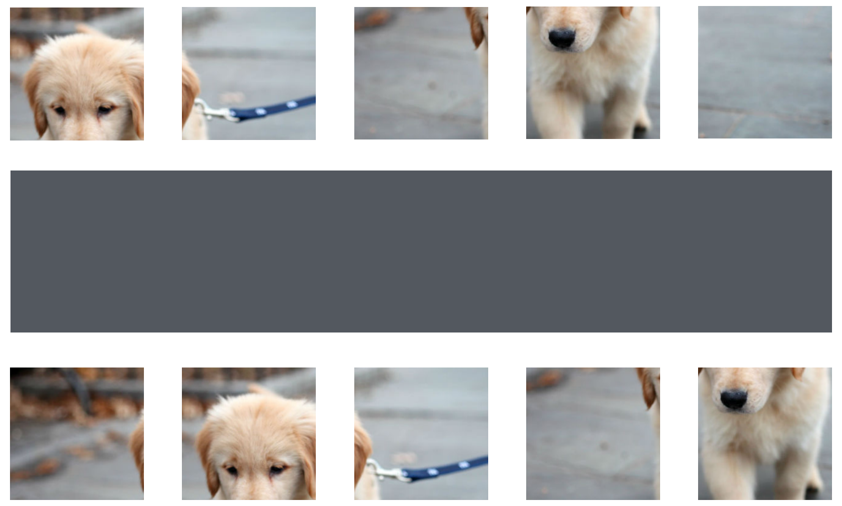
Diffusion



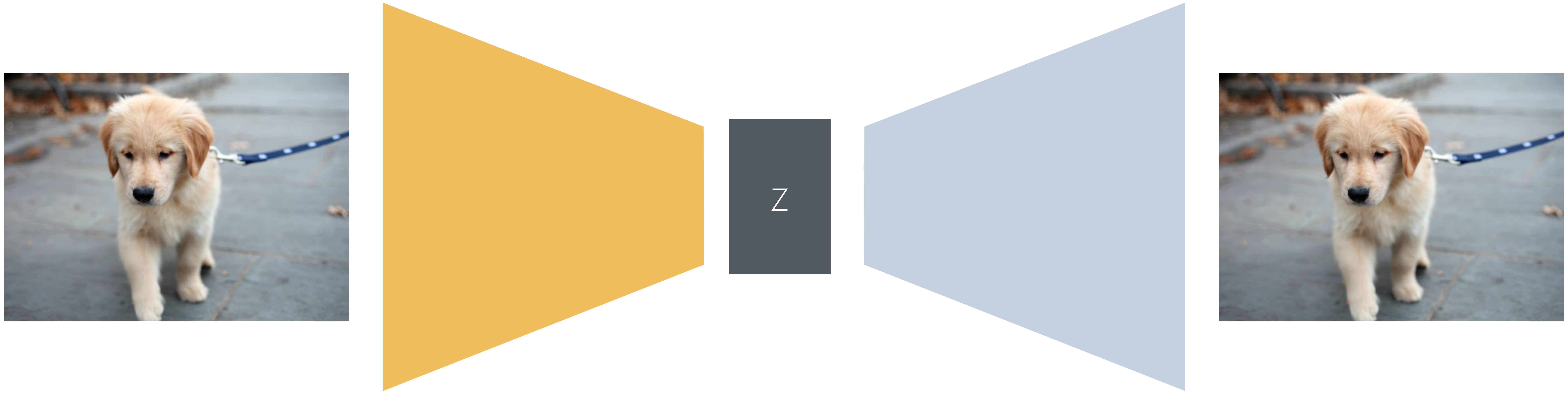
VAE



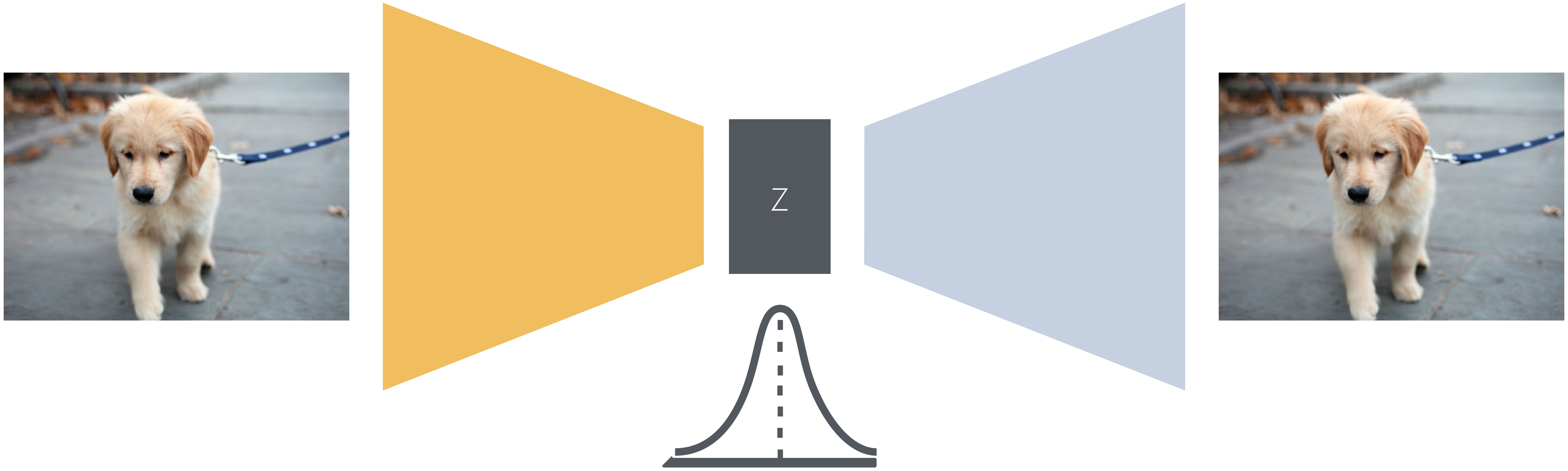
Auto-regressive



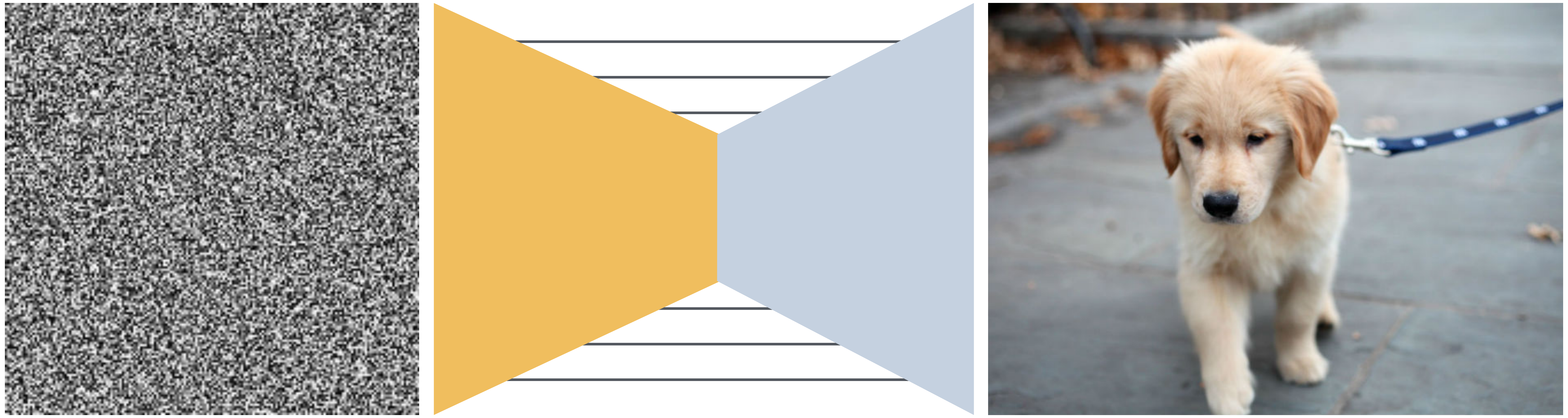
Auto-encoder



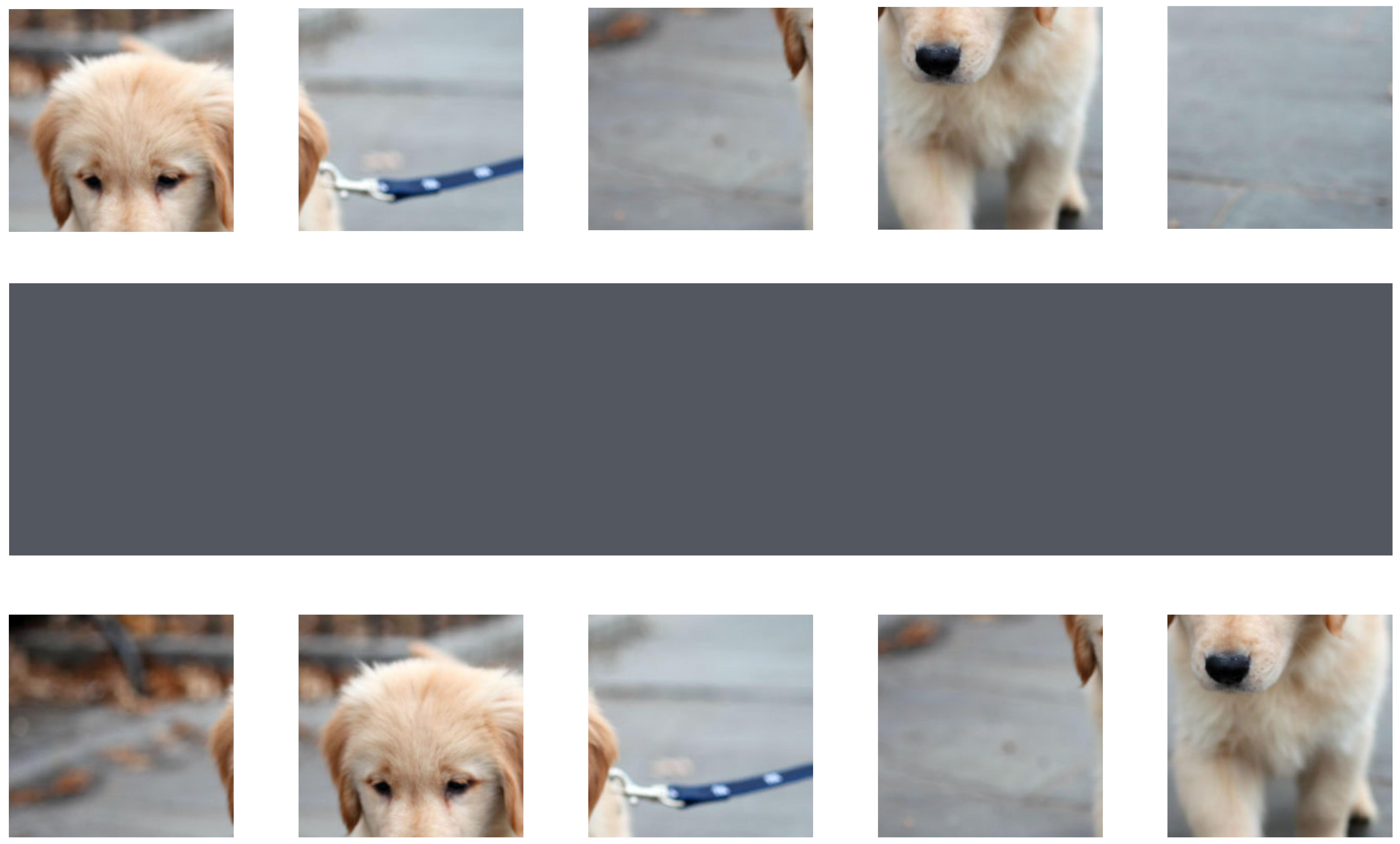
Variational Auto-Encoder



Diffusion Models



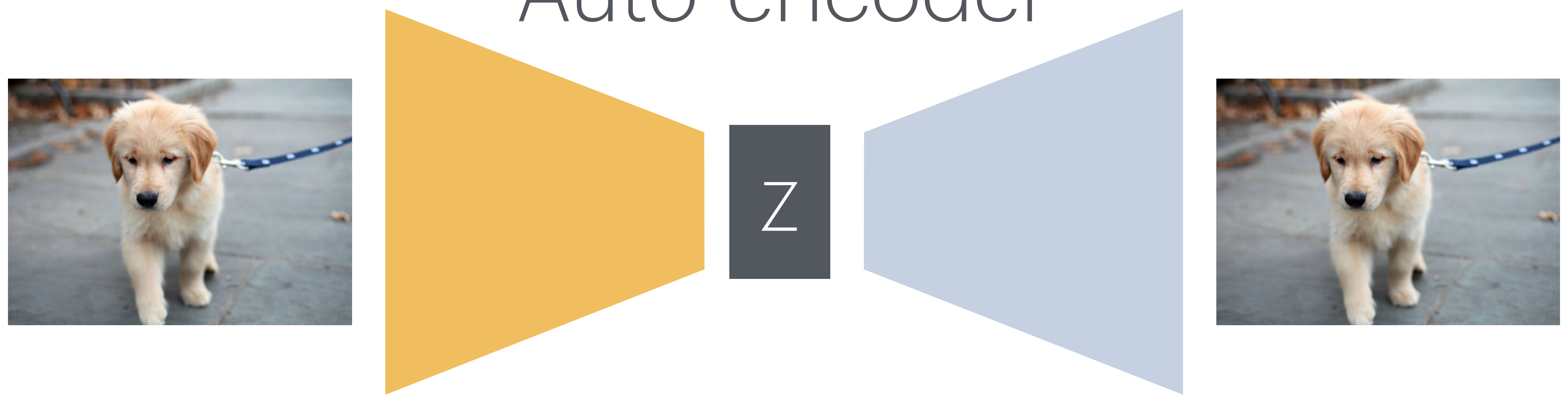
Auto-regressive Models



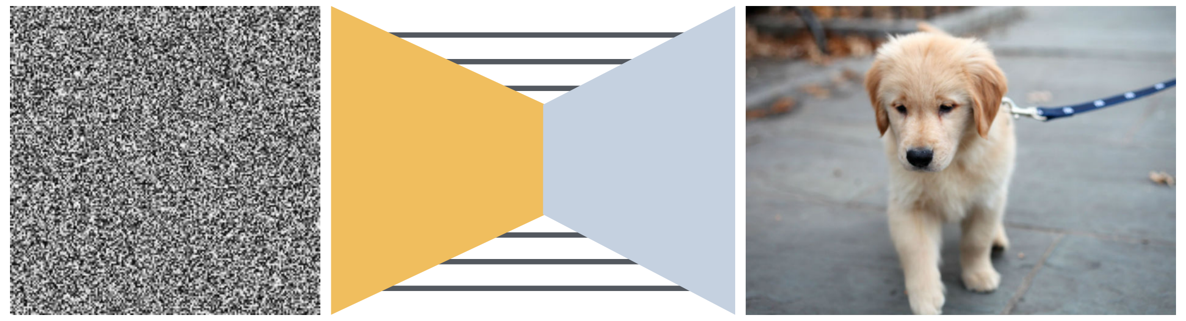
What model should I use?

As a generative model

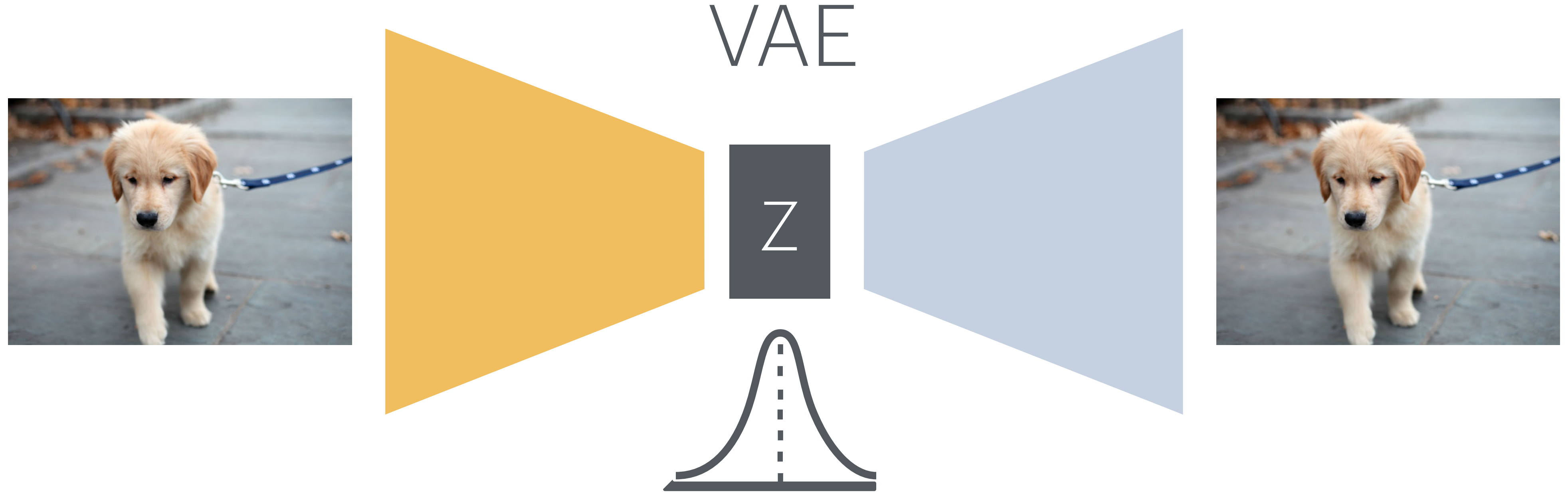
Auto-encoder



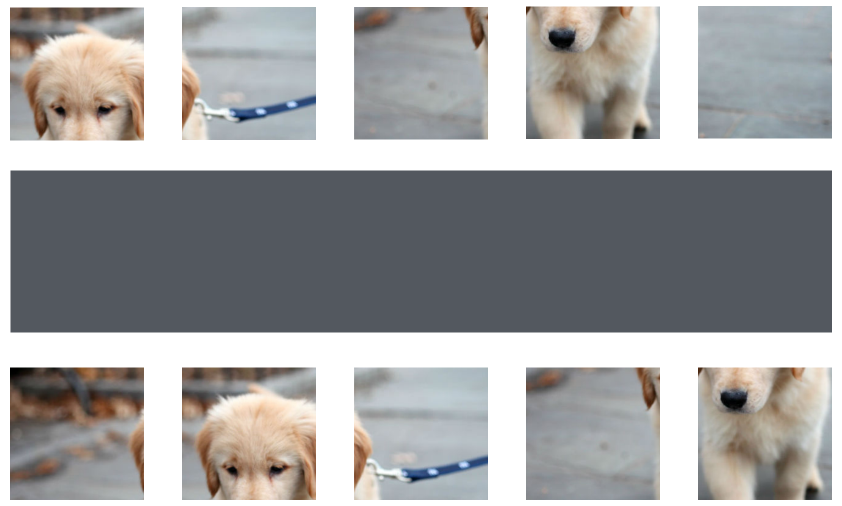
Diffusion



VAE



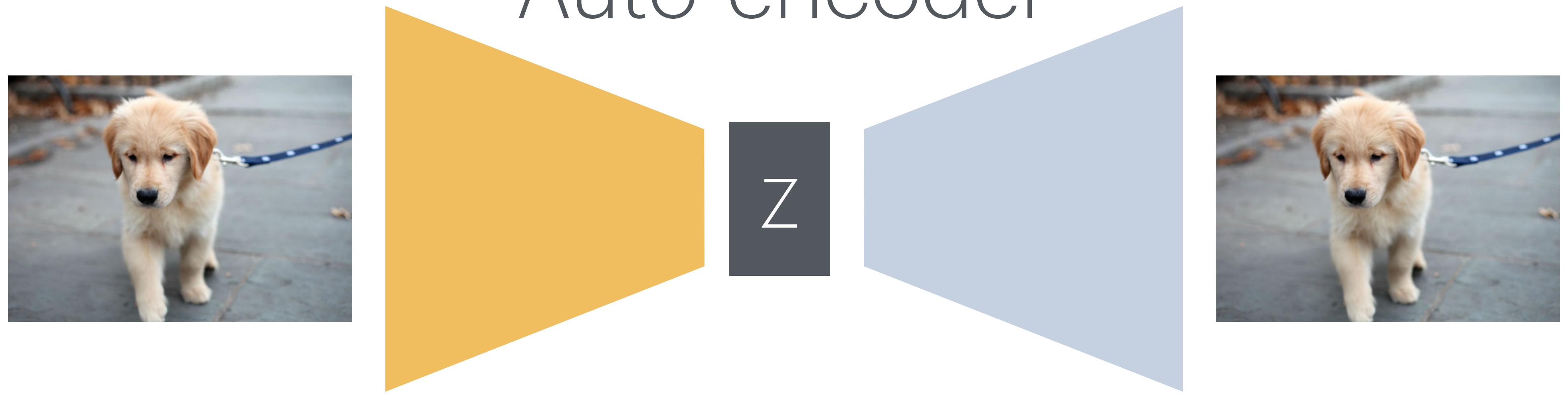
Auto-regressive



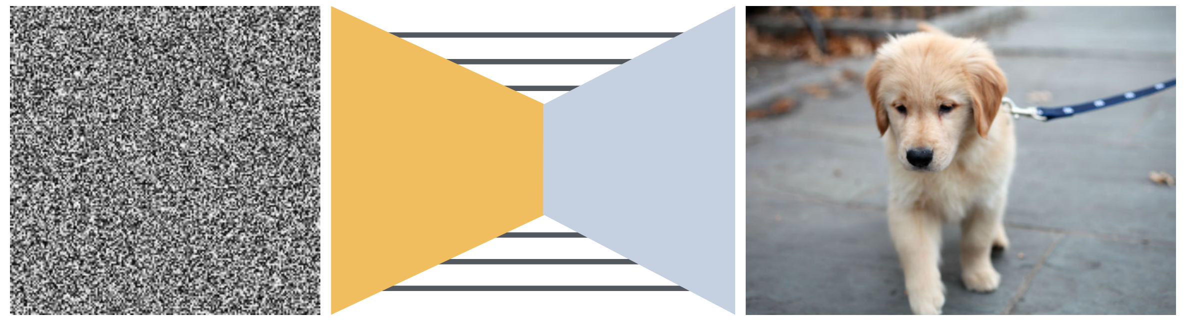
What model should I use?

Other uses

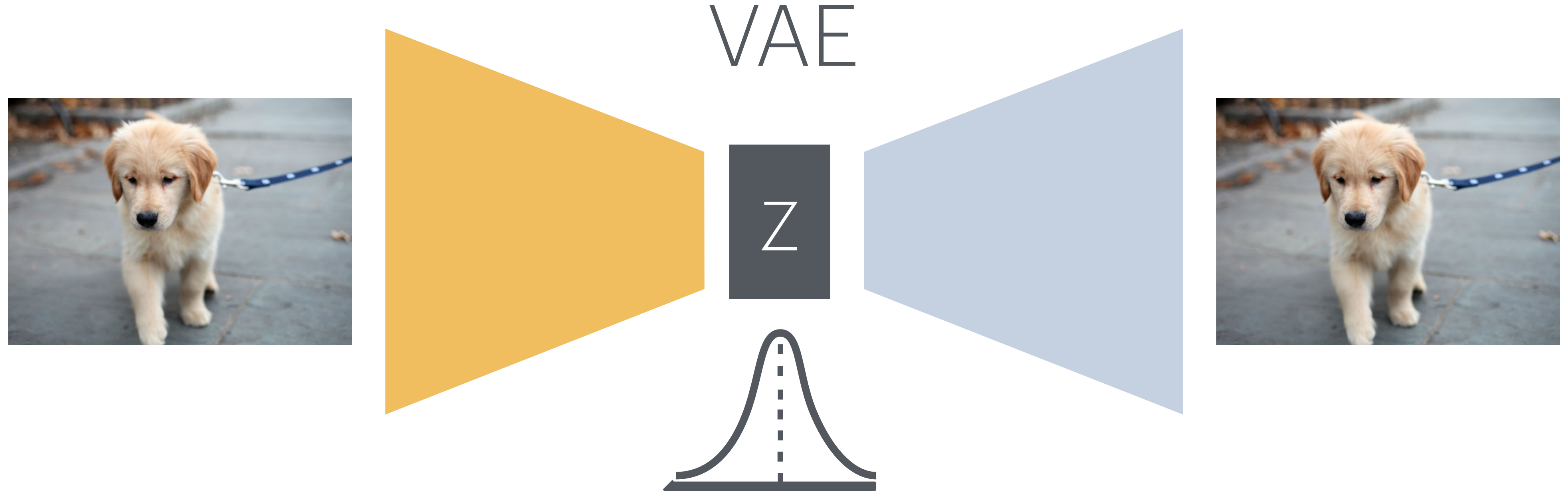
Auto-encoder



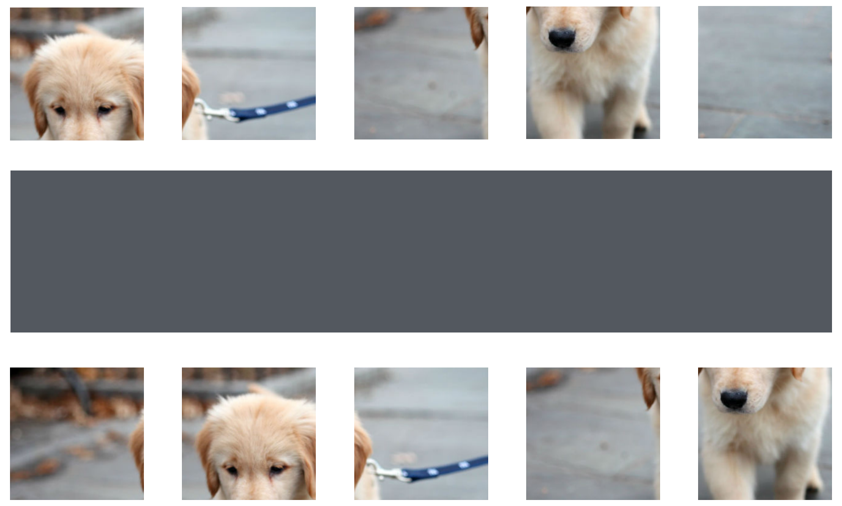
Diffusion



VAE



Auto-regressive



What model should I use?

Pre-trained models exist

Own domain
No prior models exist

What model should I use?

Pre-trained models exist		Own domain No prior models exist
Big compute	Small compute	

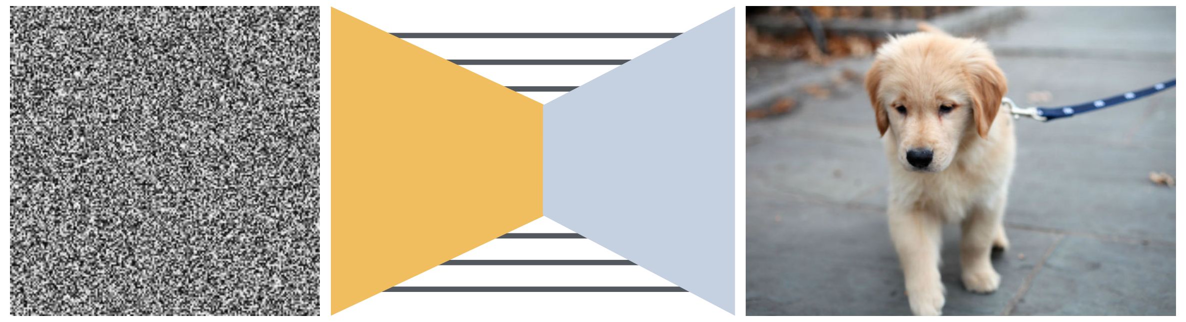
What model should I use?

Pre-trained models exist

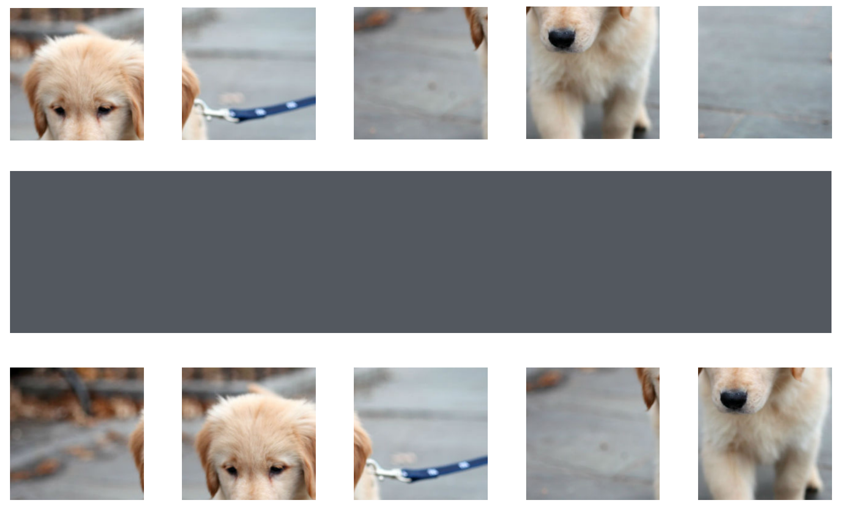
Big compute

Train

Diffusion



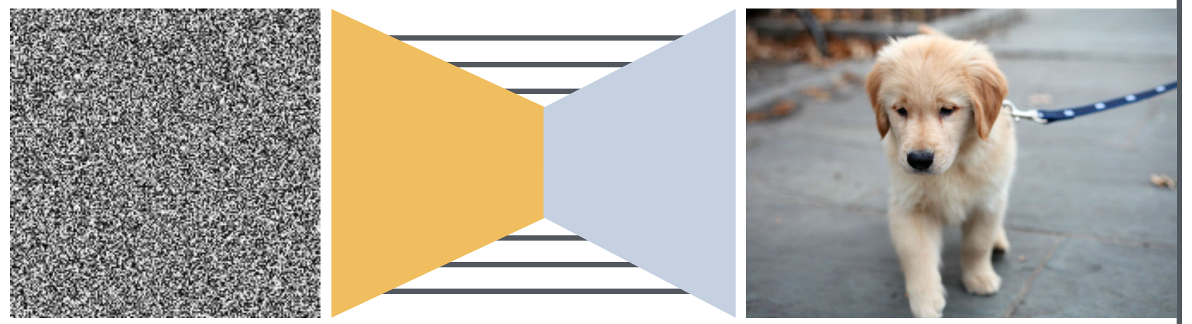
Auto-regressive



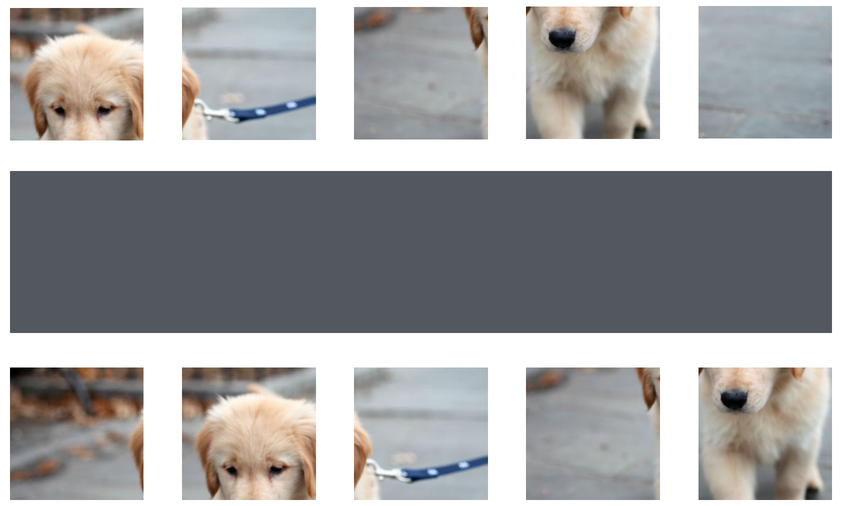
Small compute

Fine-tune

Diffusion



Auto-regressive



Own domain
No prior models exist

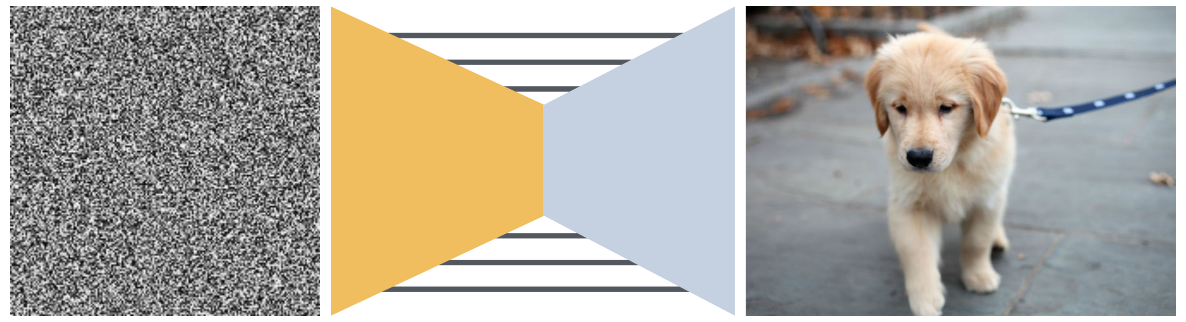
What model should I use?

Pre-trained models exist

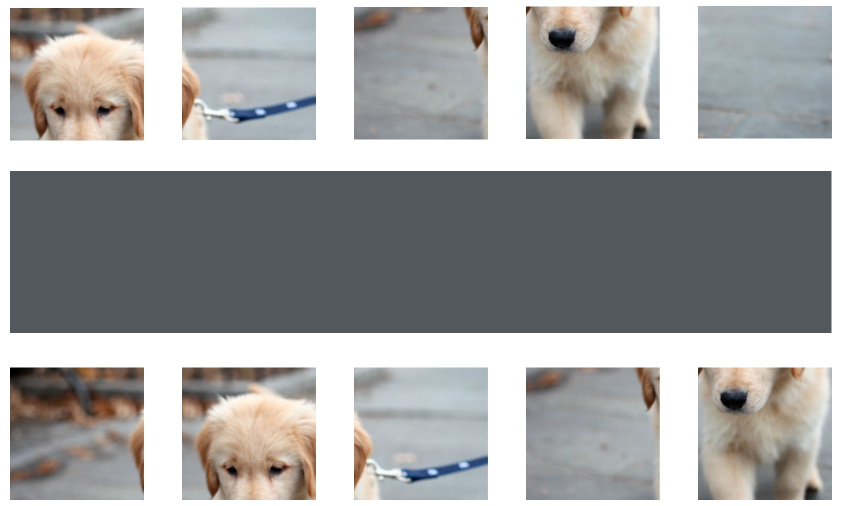
Big compute

Train

Diffusion



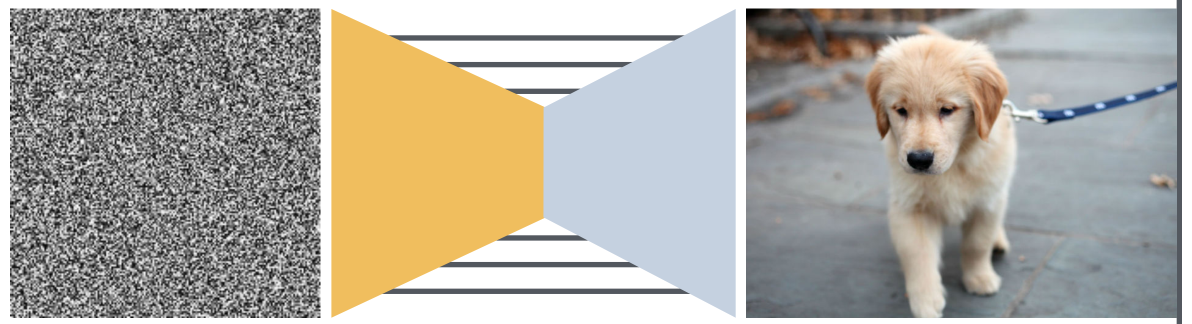
Auto-regressive



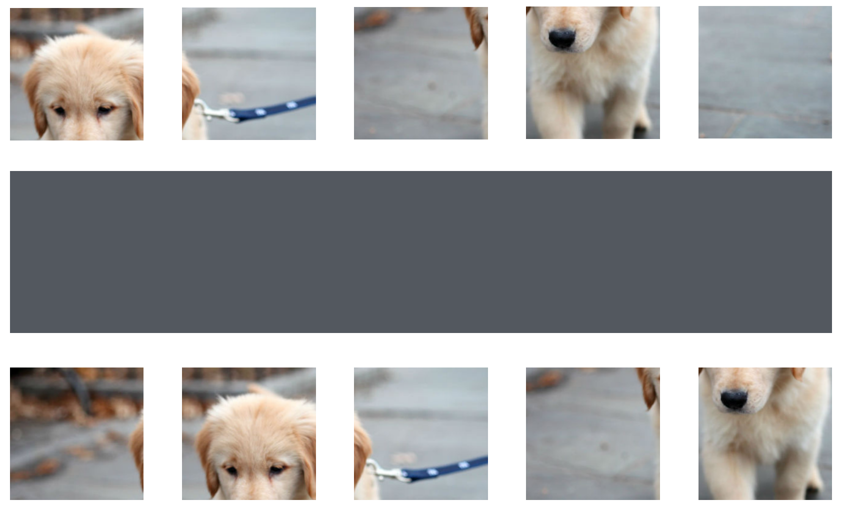
Small compute

Fine-tune

Diffusion



Auto-regressive



Own domain
No prior models exist

Big data, big compute

Small data
small compute

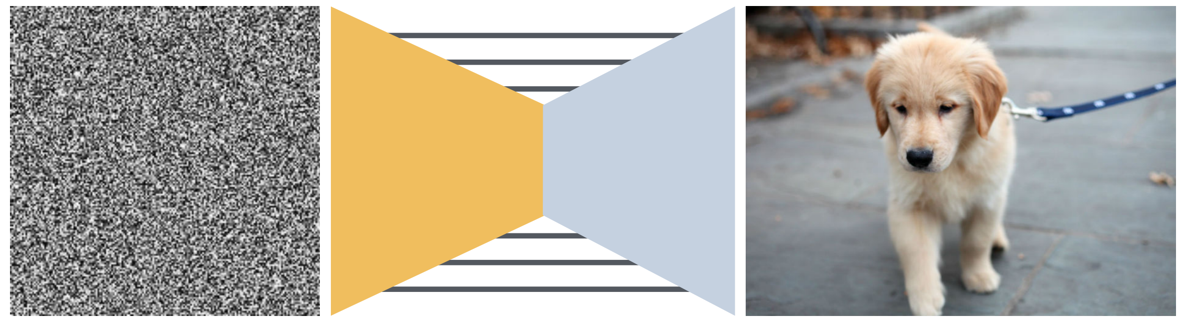
What model should I use?

Pre-trained models exist

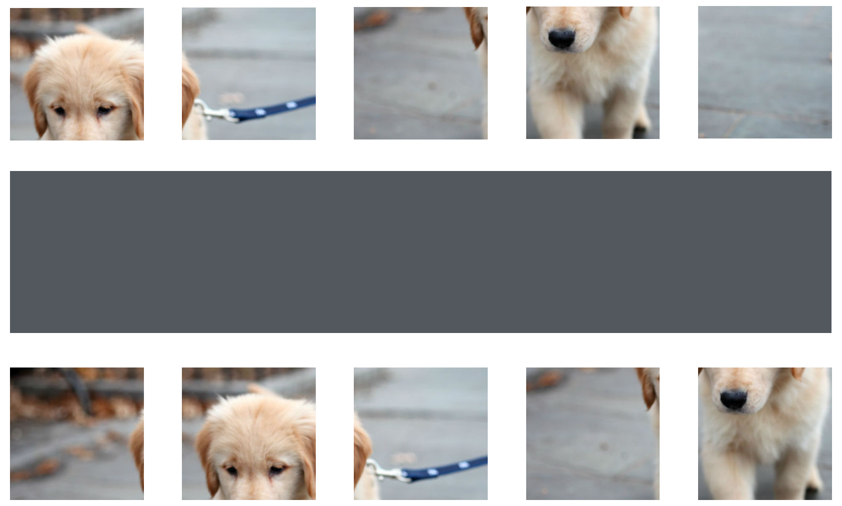
Big compute

Train

Diffusion



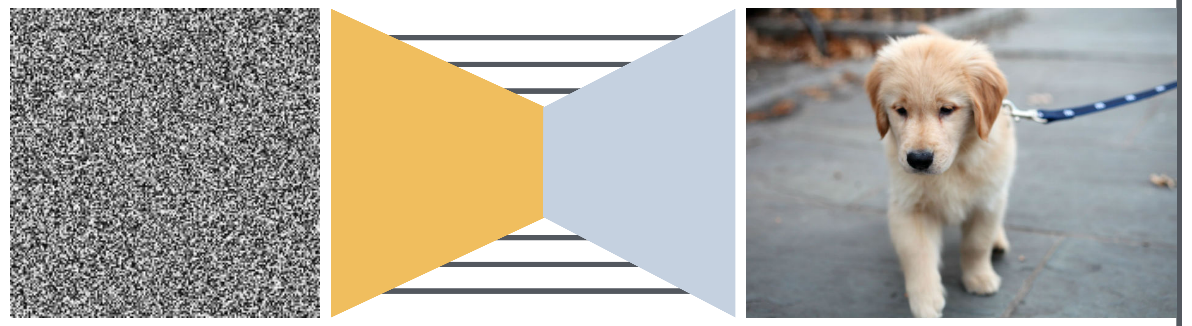
Auto-regressive



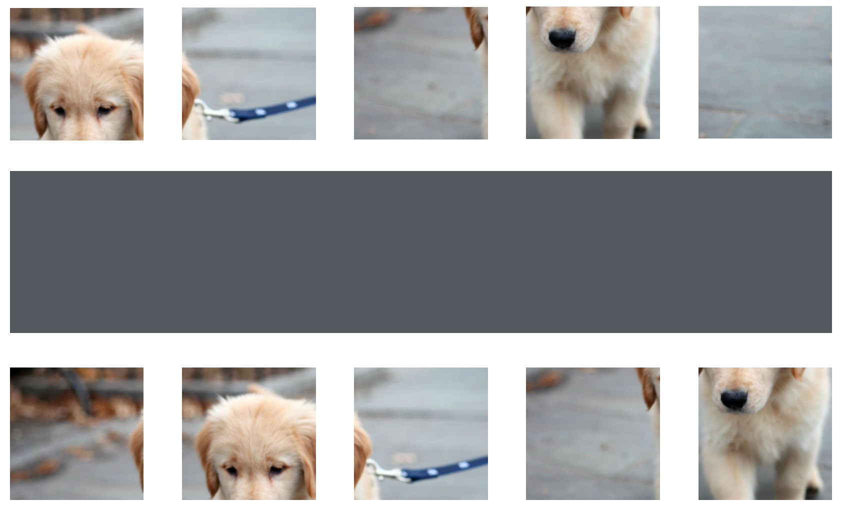
Small compute

Fine-tune

Diffusion



Auto-regressive

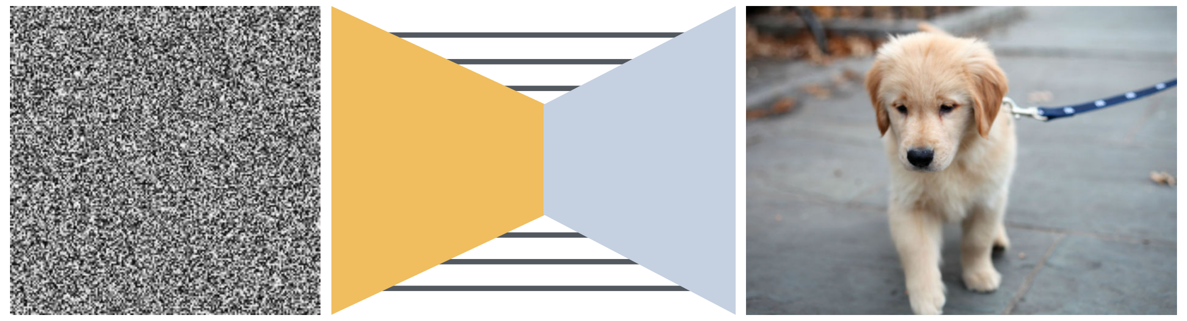


Own domain
No prior models exist

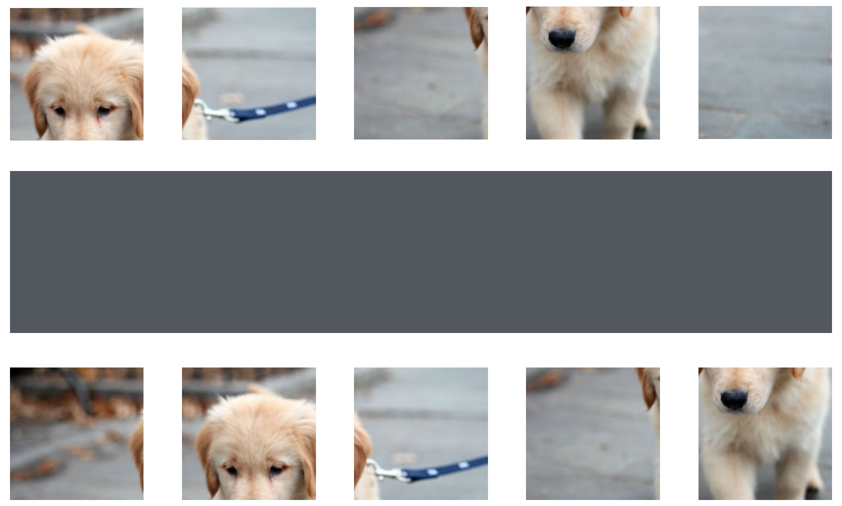
Big data, big compute

Train

Diffusion



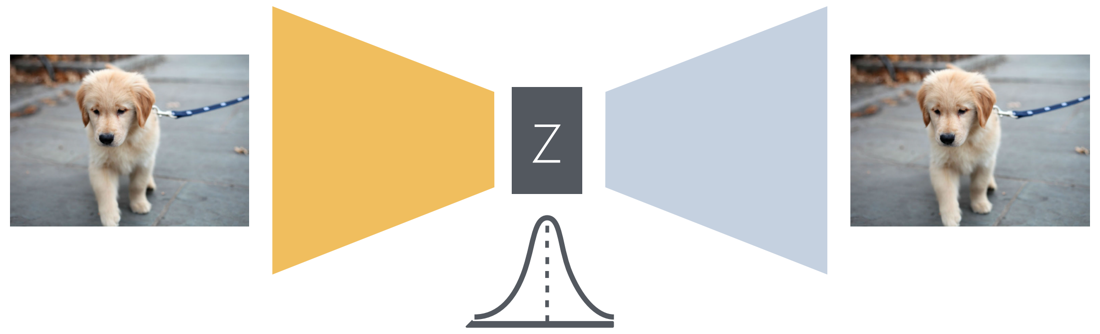
Auto-regressive



Small data
small compute

Train

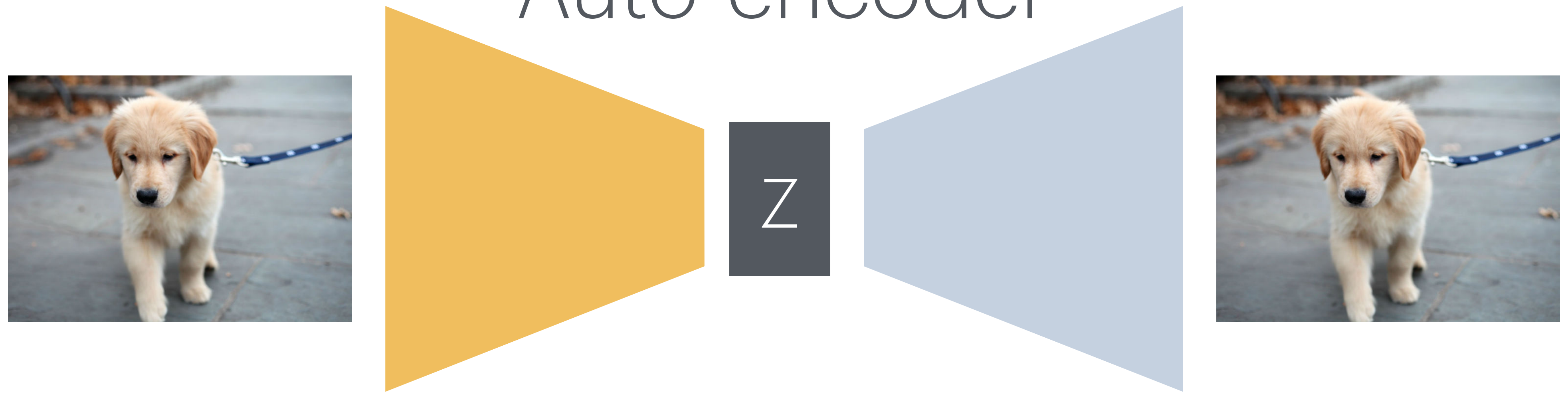
VAE



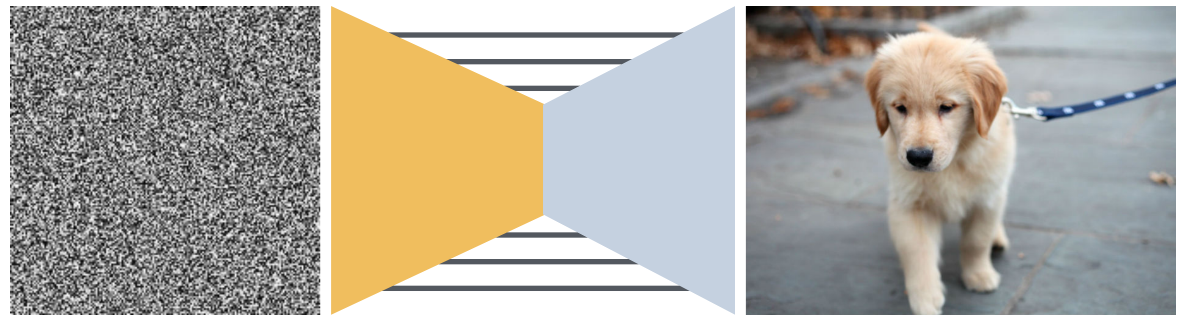
What model should I use?

Other uses

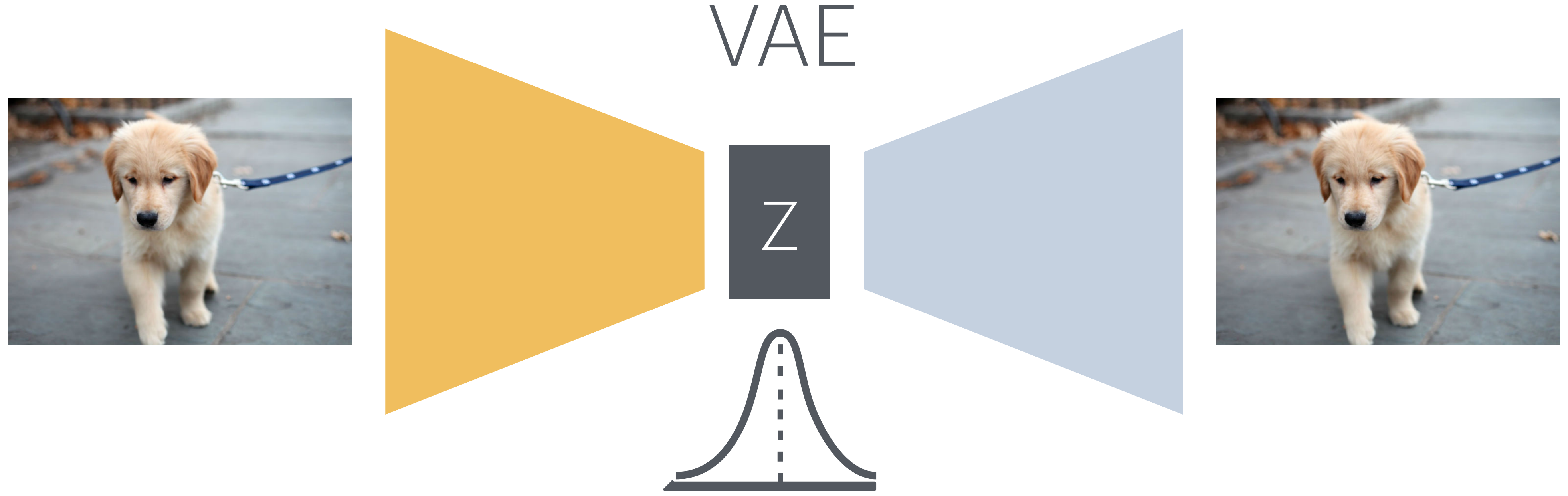
Auto-encoder



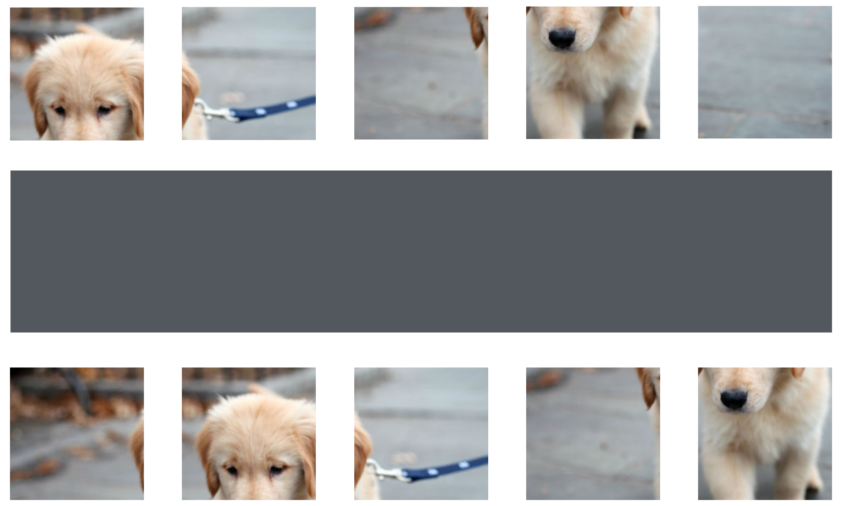
Diffusion



VAE



Auto-regressive



Diffusion - What have we learned about Deep Learning?

Philipp Krähenbühl, UT Austin

Data is King

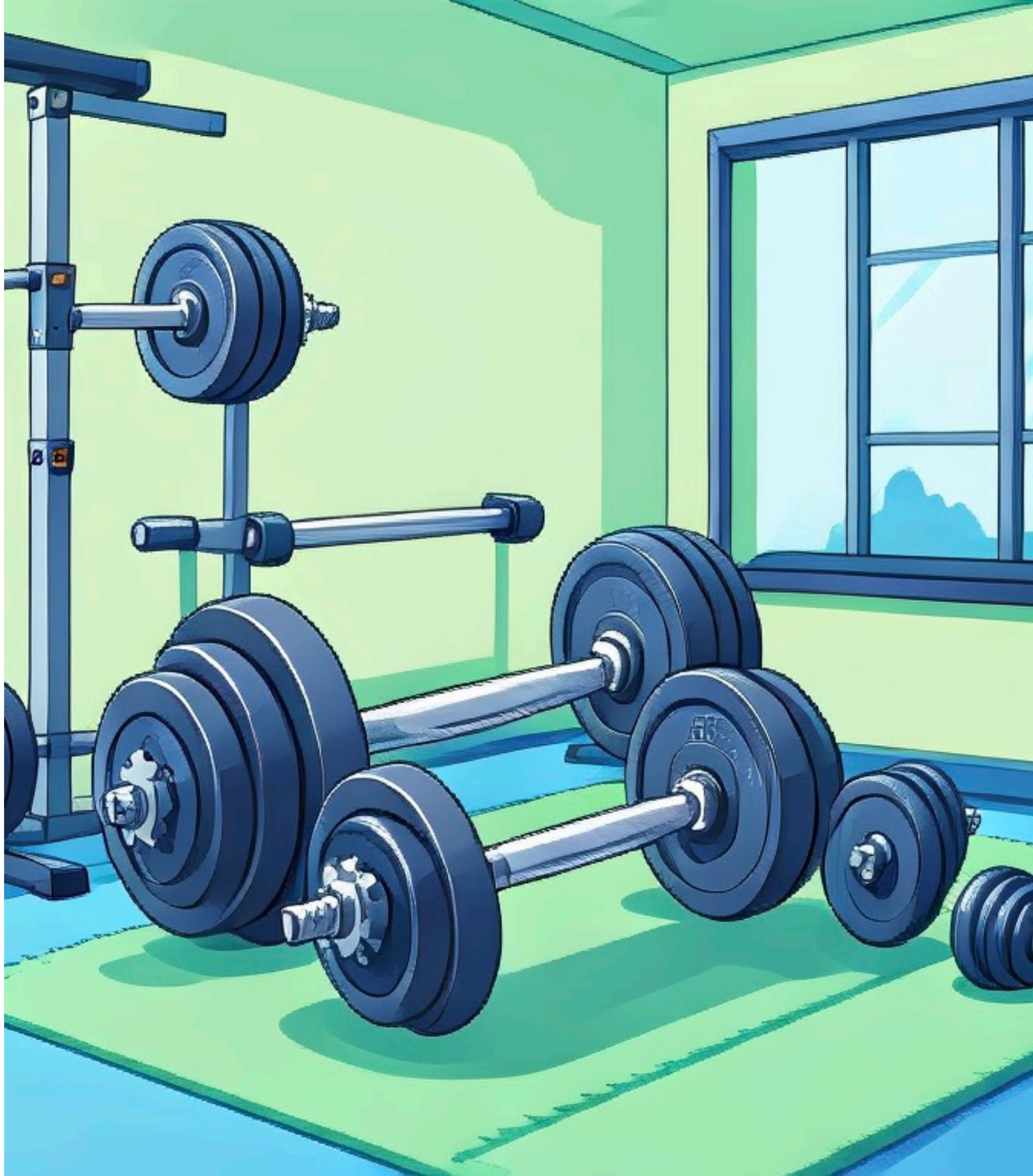
- Large dataset = Good generations
- High-quality dataset = Good generations

Models are secondary

- Tweaks on architecture give small improvements
- Systems and training setups matter
- Model = Capacity + Efficiency

Training is solved

- SGD / Adam just works



Processes matter

- Training is more complex than f(x)=y
- Inferences is more complex than f(x)=y