Lecture 7: Vision Diffusion and Generative Models

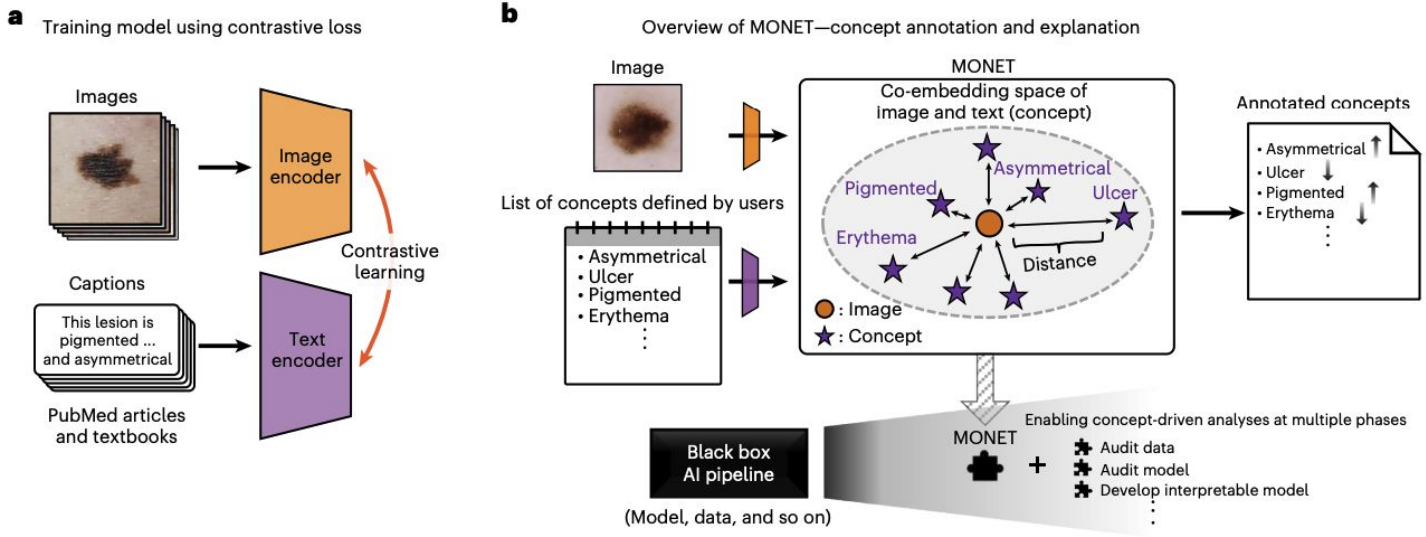
Announcements

- A1 is due Wednesday 11:59pm. Max 2 late days.
- Discussion presentation instructions and sign-up will be released shortly
  - Pay attention to Ed for announcement on submitting preferences, deadline will be Wednesday.
  - ~ 4 papers will be discussed per class; each student will be responsible for presenting 1 paper during the discussion class slots, and preparing questions for 2 others.
  - Worth 15% of grade.

Finishing up from last time… Vision-Language Representation Learners in Biomedicine

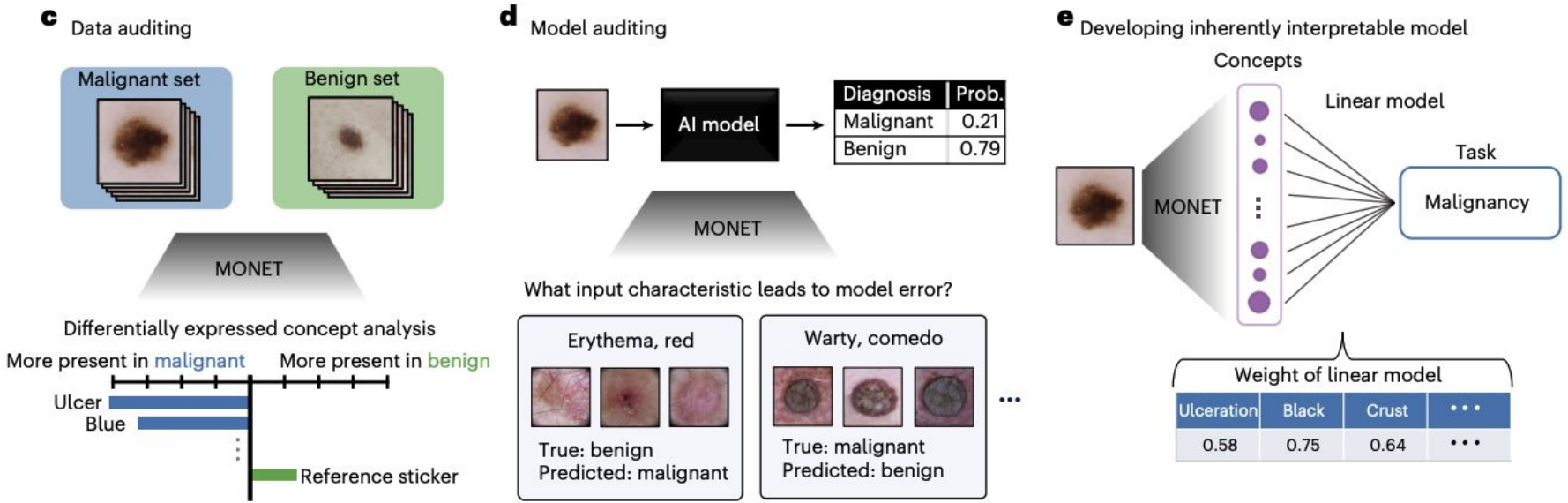
MONET: Leveraging a contrastively trained model to perform dataset and model auditing

- Trained on 105,550 dermatology image-text pairs from PubMed articles and medical textbooks



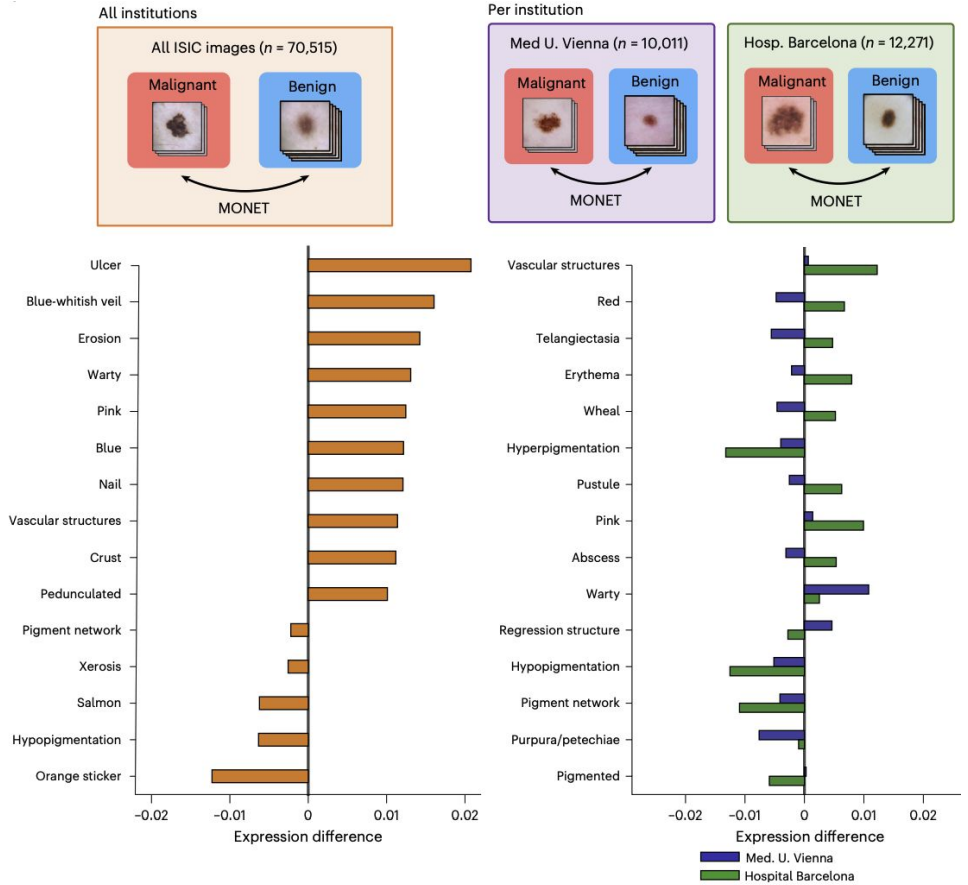
Kim et al. Transparent medical image AI via an image–text foundation model grounded in medical literature. Nature Medicine, 2024.

MONET: Leveraging a contrastively trained model to perform dataset and model auditing



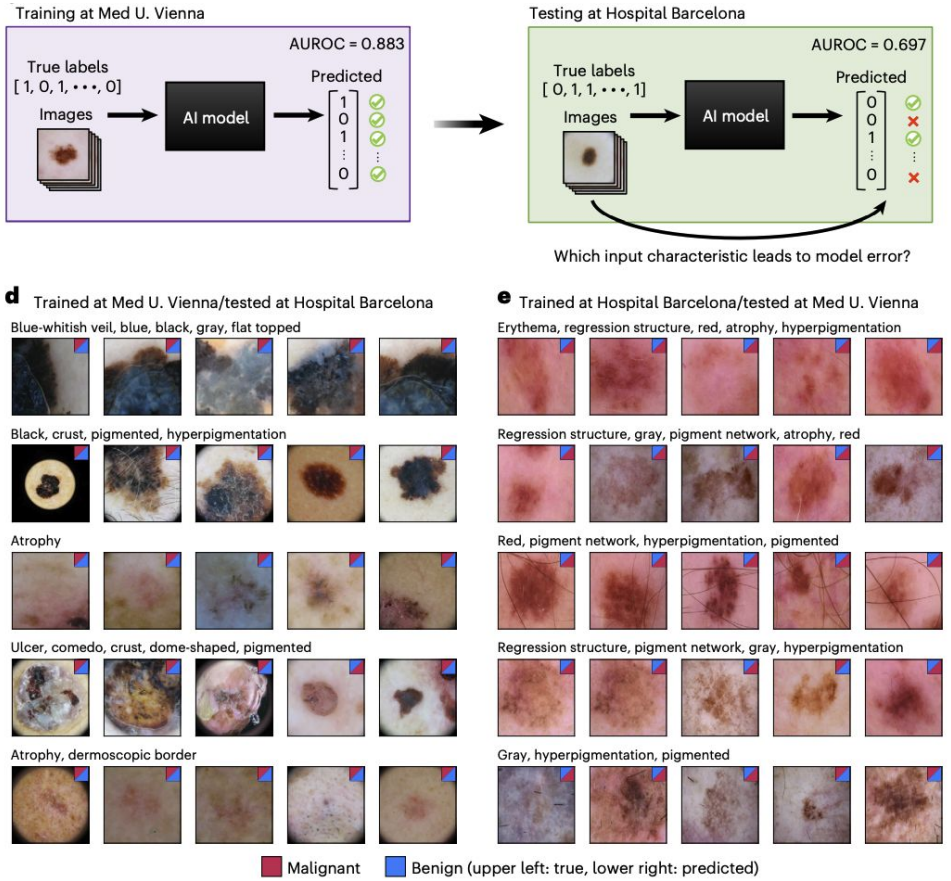
Kim et al. Transparent medical image AI via an image–text foundation model grounded in medical literature. Nature Medicine, 2024.

Concept differences identified by MONET during data auditing



Kim et al. Transparent medical image AI via an image–text foundation model grounded in medical literature. Nature Medicine, 2024.

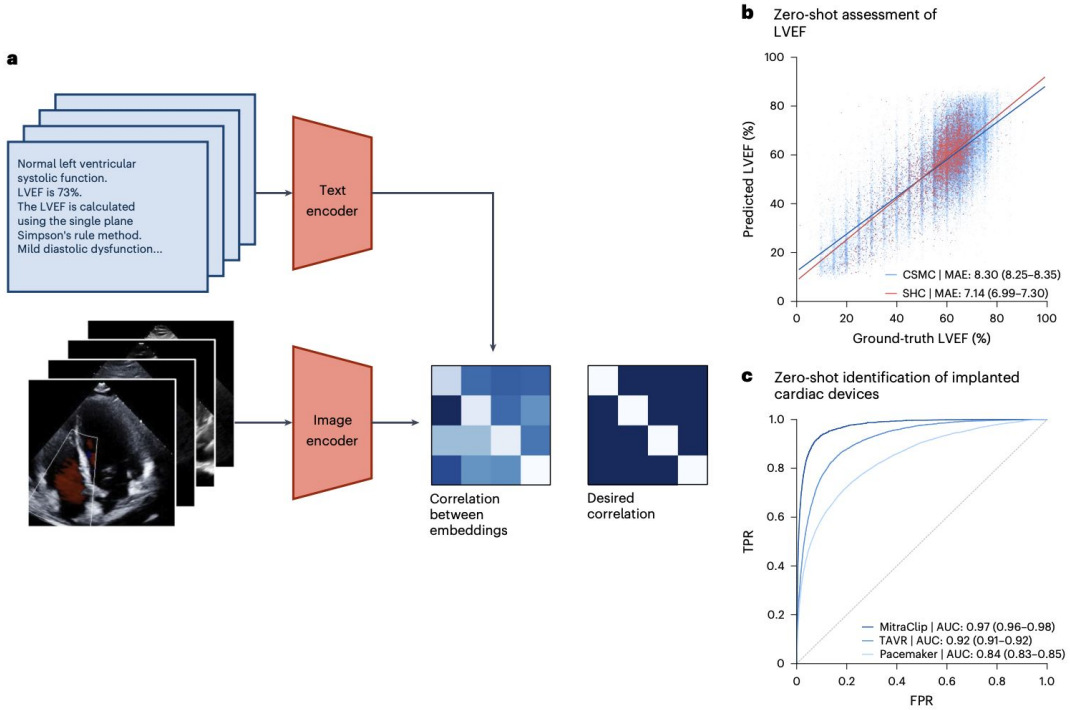
Concept differences identified by MONET during model auditing



Kim et al. Transparent medical image AI via an image–text foundation model grounded in medical literature. Nature Medicine, 2024.

Some more examples of CLIP-based foundation models…

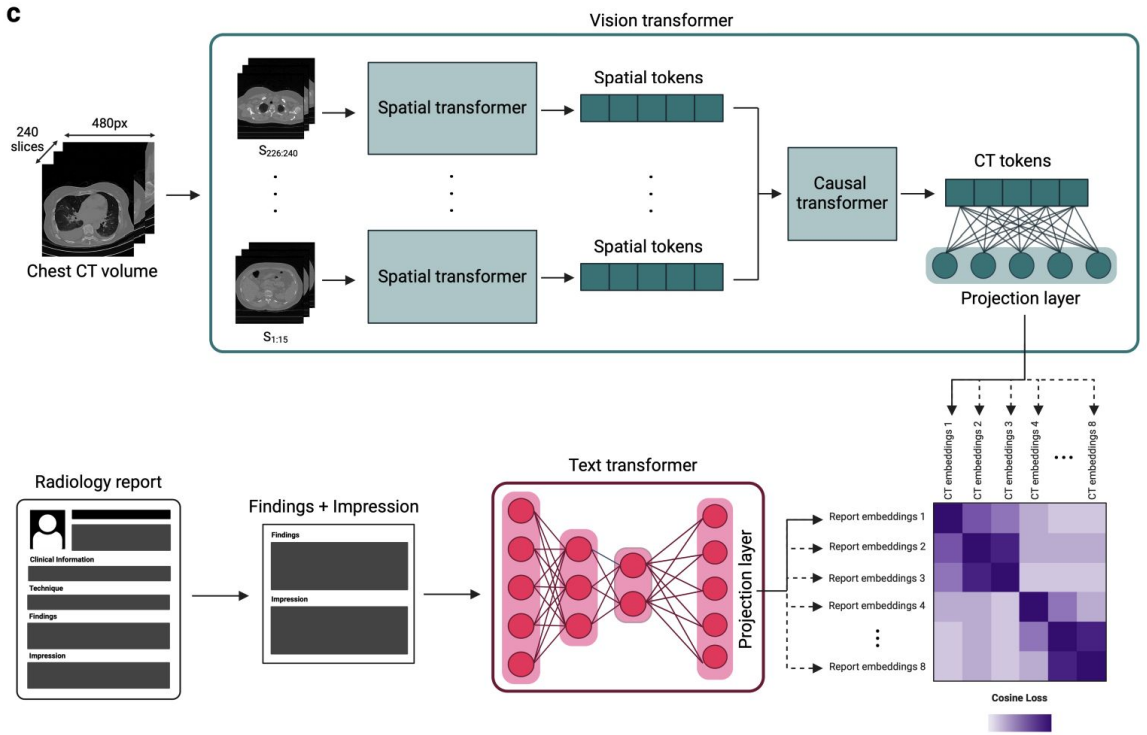
EchoCLIP: Based on 1,032,975 video-text pairs (but trained on images) for echocardiogram interpretation



Christensen et al. Vision–language foundation model for echocardiogram interpretation. Nature Medicine, 2024.

Some more examples of CLIP-based foundation models…

CT-CLIP: trained on 50,188 3D chest CT volumes with accompanying reports, using a previously developed 3D image encoder

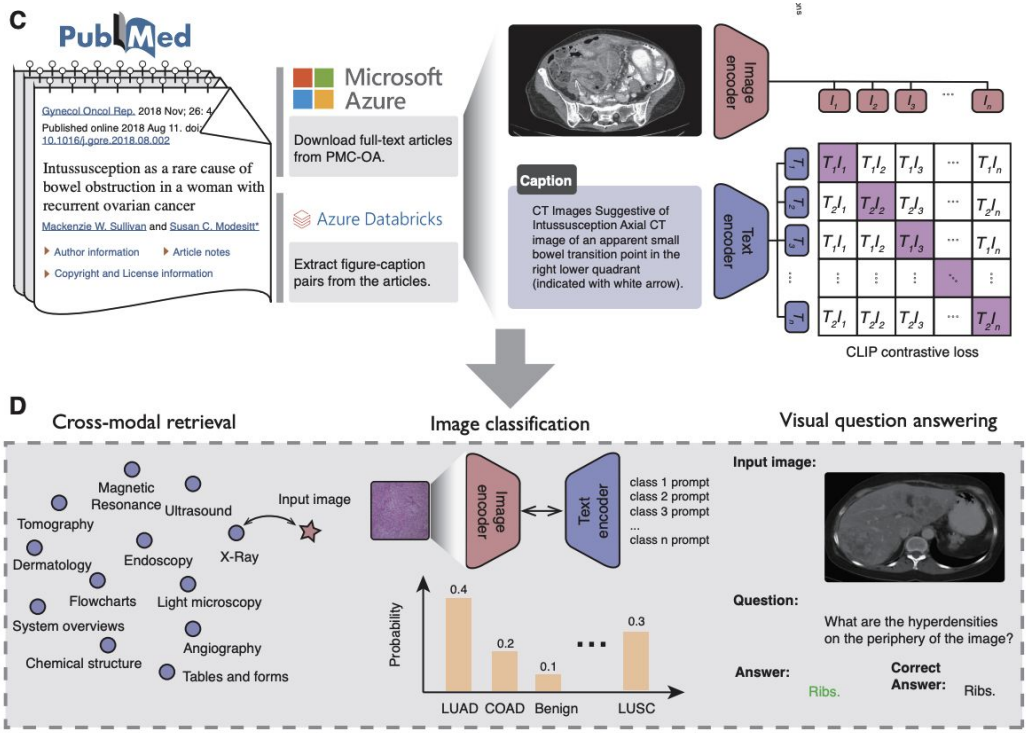


Hamamci et al. A foundation model utilizing chest CT volumes and radiology reports for supervised-level zero-shot detection of abnormalities. arXiv, 2024.

BiomedCLIP: Generalist foundation model trained on PubMed

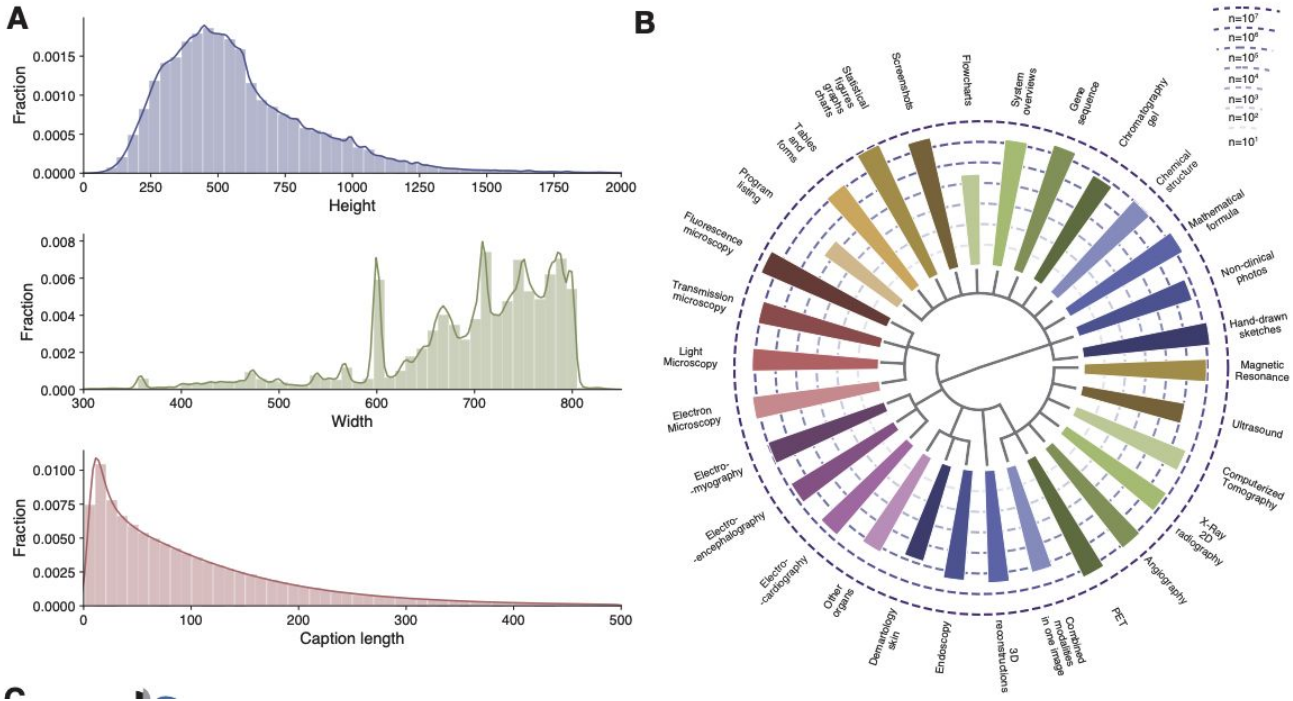
PMC-15M: 15 million image-caption pairs from 4.4 million publicly available full-text articles in PubMed Central

Complete article packages are downloaded, and figure files and matching captions are extracted



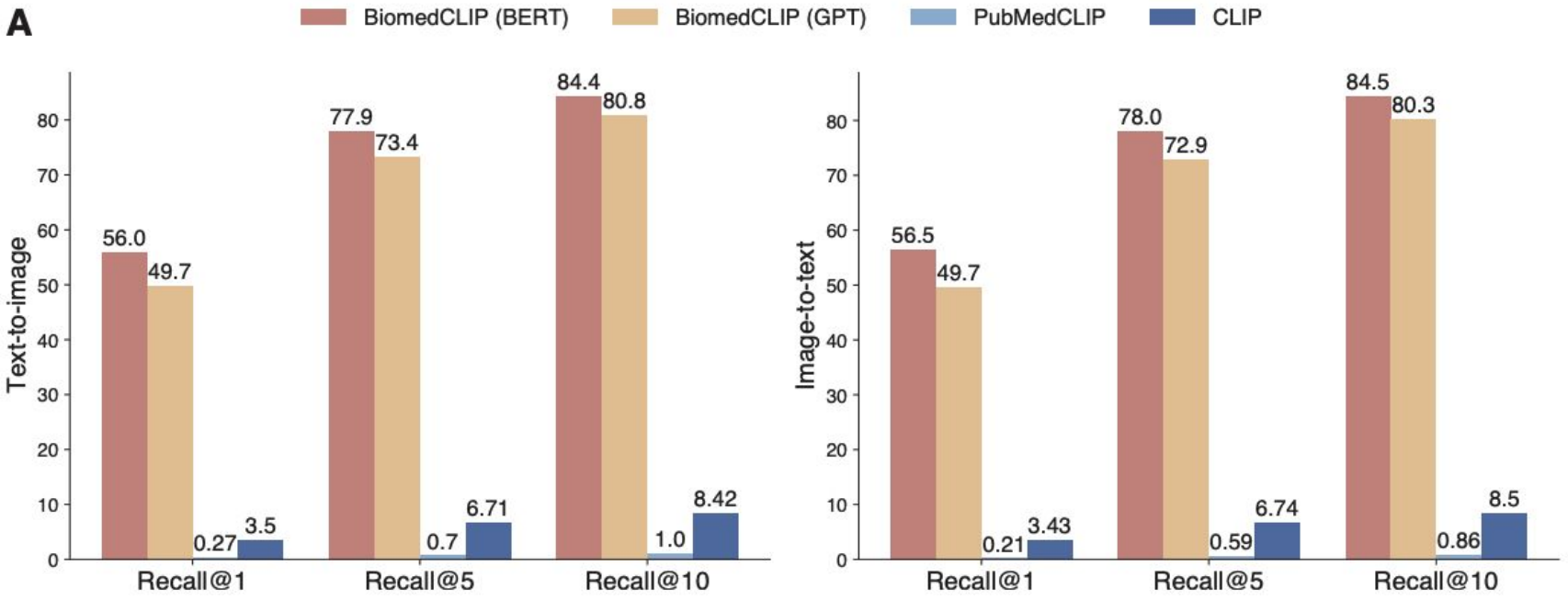
Zhang et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv, 2024.

PMC-15M statistics of image sizes and caption lengths



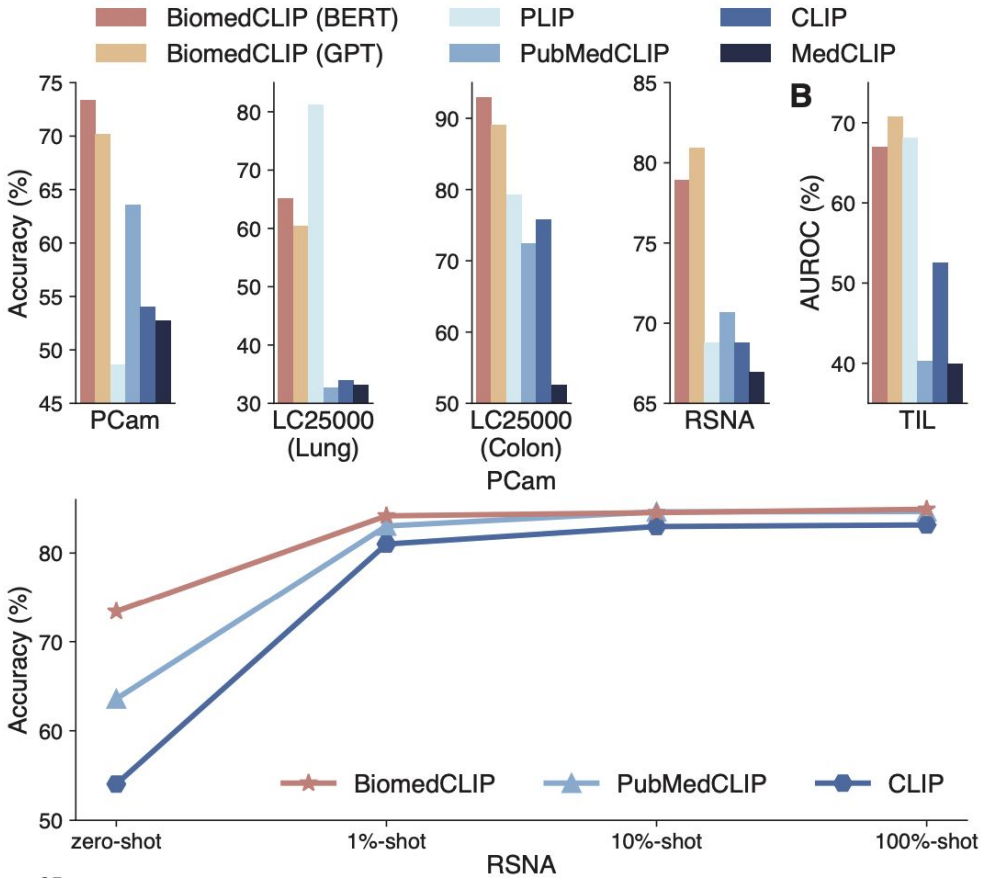
Zhang et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv, 2024.

BiomedCLIP comparison with baseline methods on cross-modal retrieval



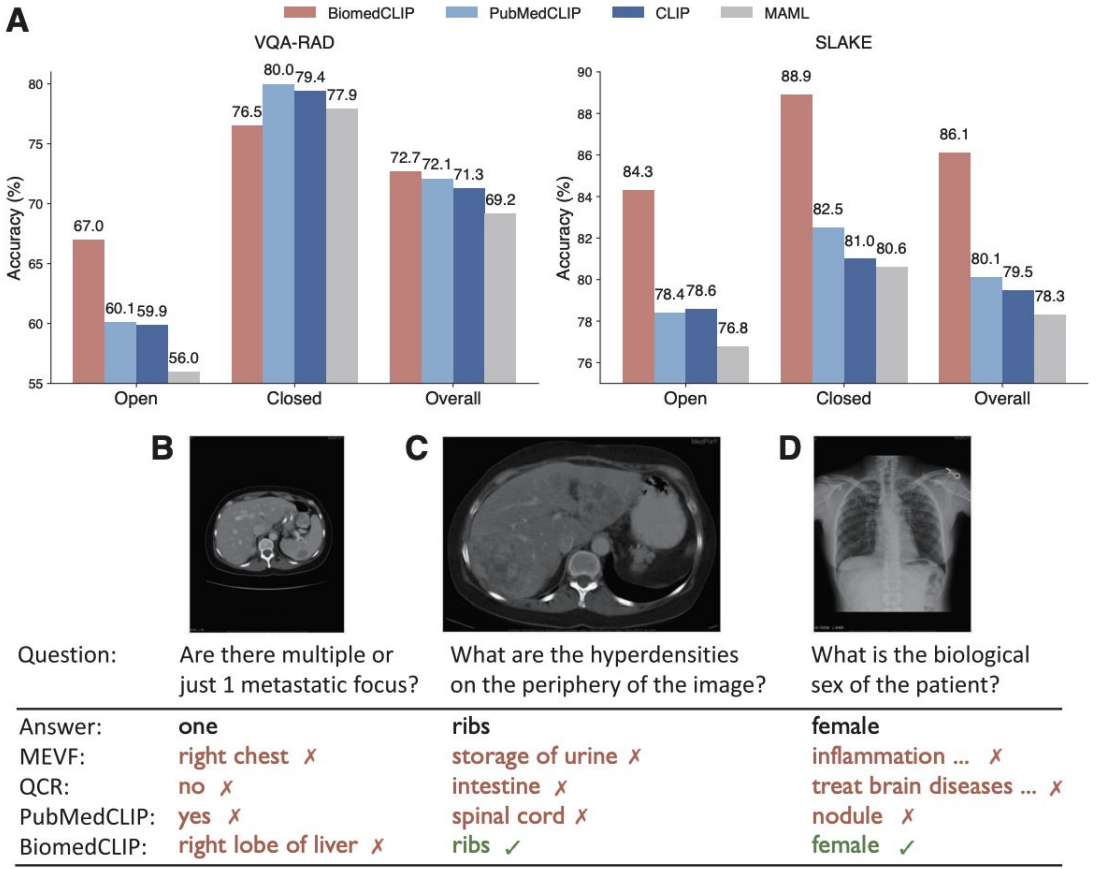
Zhang et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv, 2024.

BiomedCLIP comparison with baseline methods on zero-shot classification and linear probing



Zhang et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv, 2024.

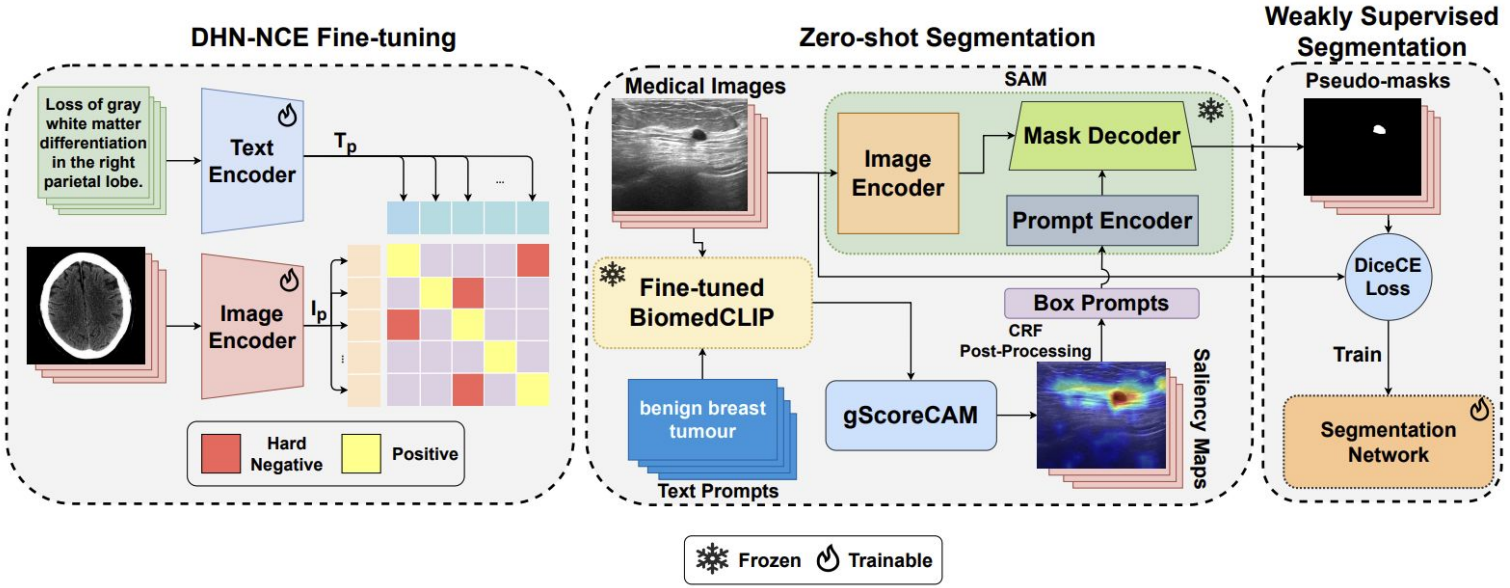
BiomedCLIP comparison with baseline methods on medical visual question answering (VQA)



Zhang et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv, 2024.

Combining biomedical CLIP models with SAM

- Approaches such as MedCLIP-SAM can perform zero-shot and weakly supervised segmentation



Koleilat et al. MedCLIP-SAM: Bridging Text and Image Towards Universal Medical Image Segmentation. MICCAI, 2024.

Next:
Vision Diffusion and Generative Models

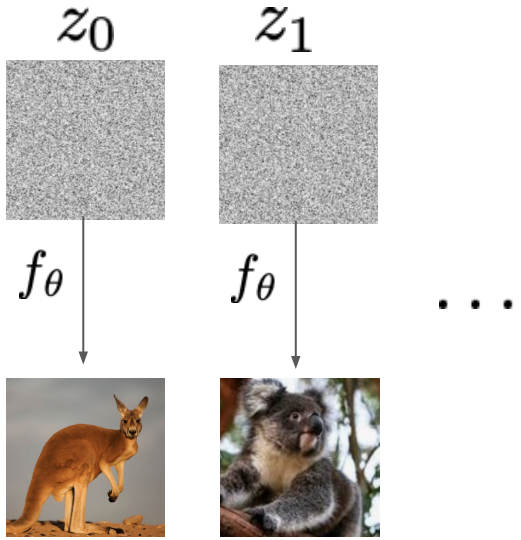
Today's agenda

- Introduction to generative models & diffusion models
- Sampling
- Training
- Advanced sampling
- Important diffusion model designs
- Evaluation
- Diffusion beyond text-to-image
- Some special topics

Introduction to generative models & diffusion models

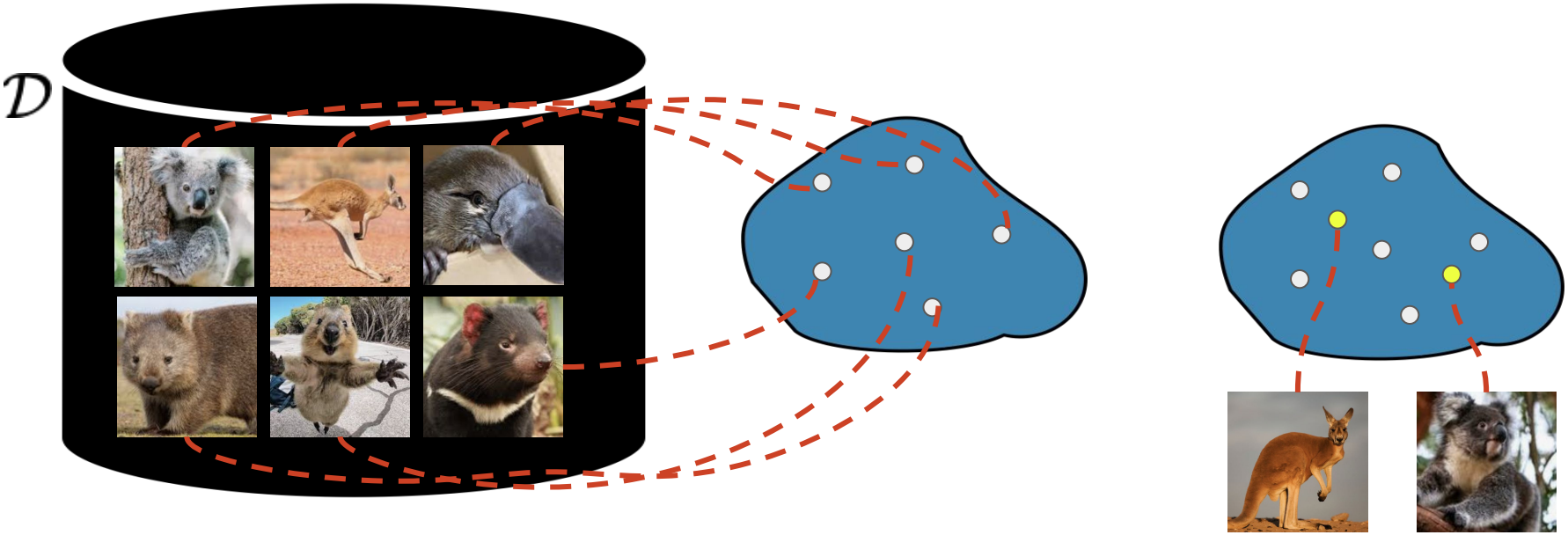
Generative modeling: the goal

Training



Generation

Generative modeling: the goal



"Testing the Manifold Hypothesis"

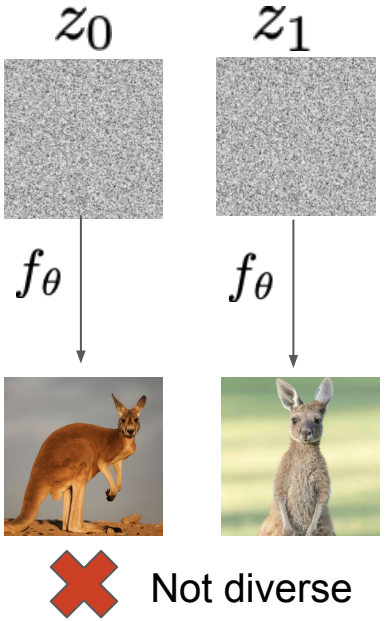
Generative modeling: the goal

Generative modeling is often motivated with other goals that we won't focus on:

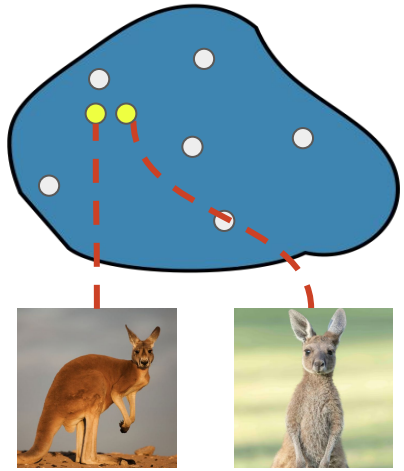
- Estimate likelihood of data, $p(x)$
- Representation learning

Generative modeling: some important properties

1. Diversity / coverage: samples should cover the whole distribution of data



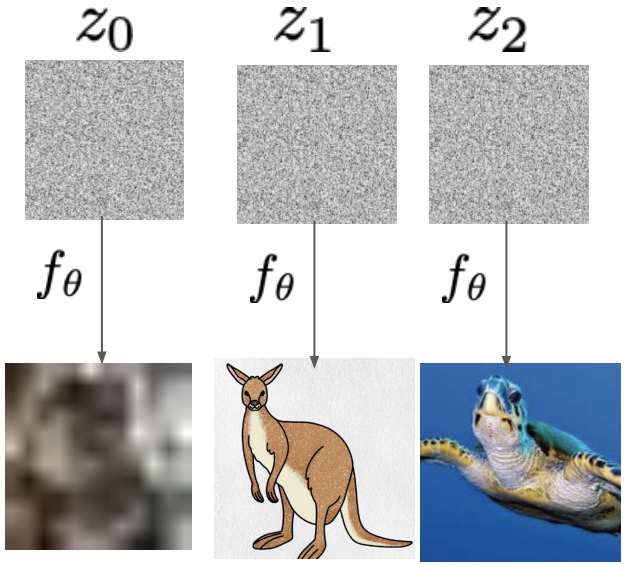
Not diverse

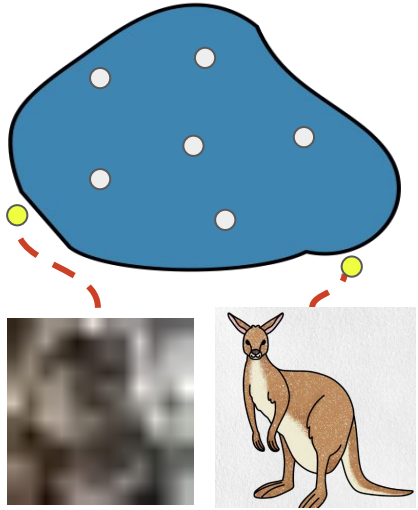


Not sampling the whole distribution

Generative modeling: some important properties

2. Fidelity: samples should come from the same distribution as the data



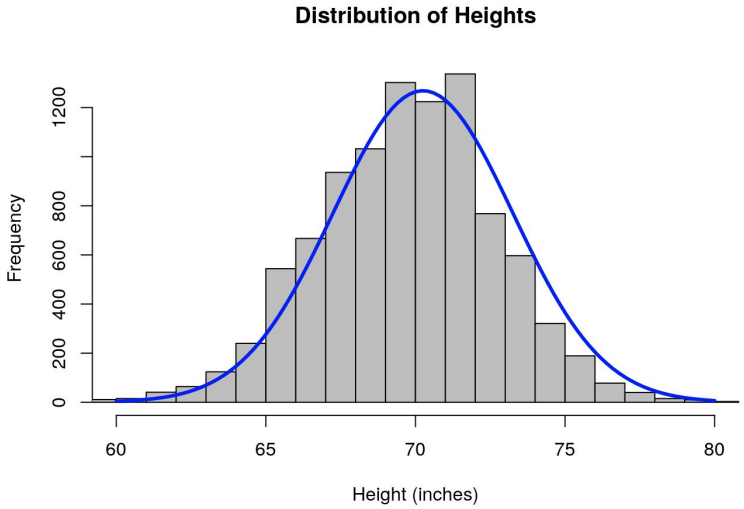


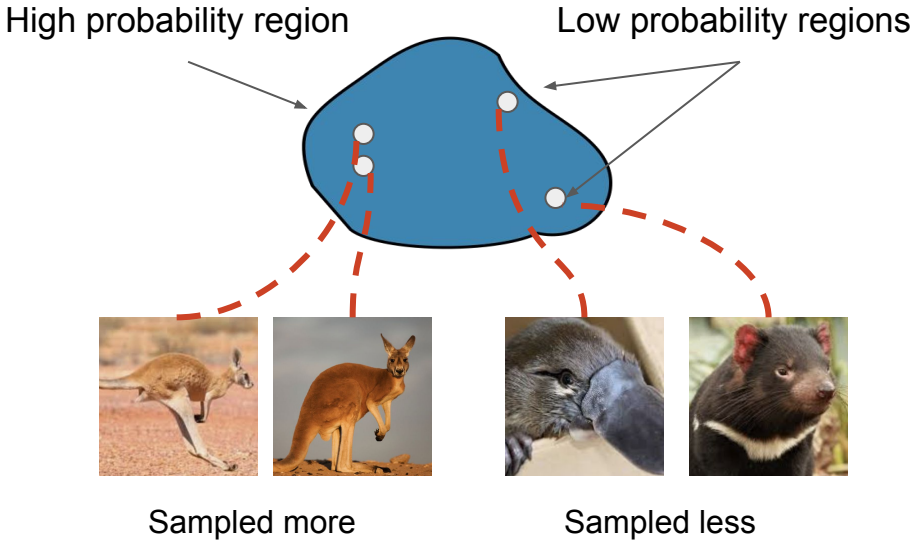
Different distribution from the data

"Different distribution" is often used in a fuzzy way

Generative modeling: some important properties

3. Sampling frequency reflects the true distribution

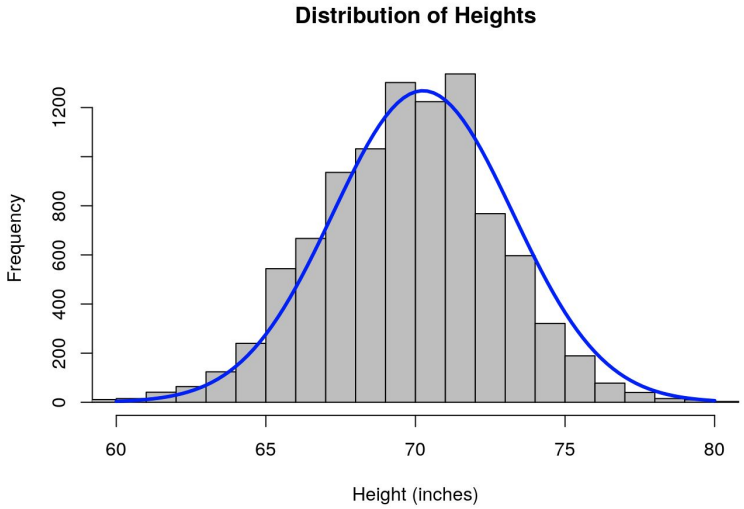


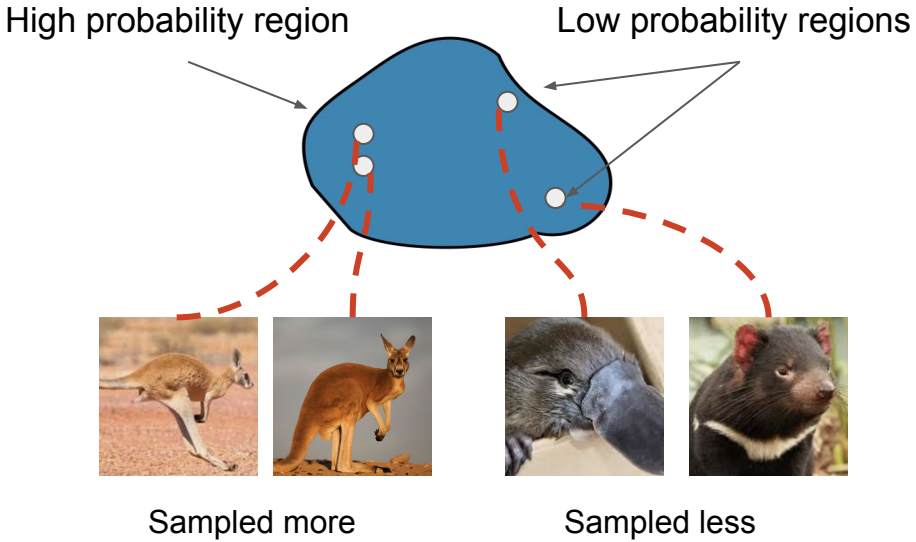


Note: this point subsumes the goals of ‘diversity’ and ‘fidelity’

Generative modeling: some important properties

3. Sampling frequency reflects the true distribution



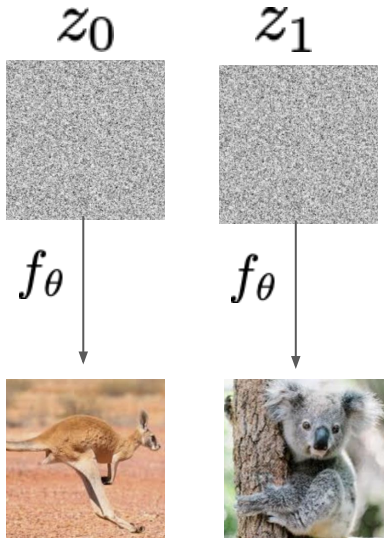


Objective is to match training distribution

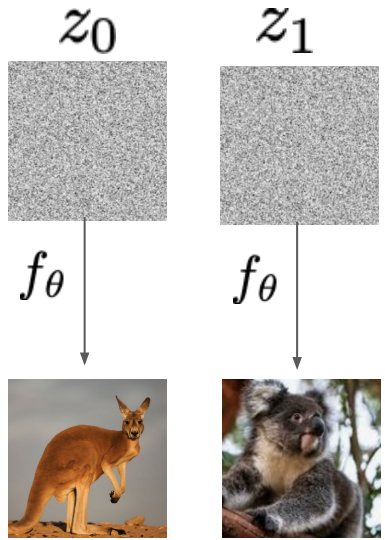
$$\text{argmin}_\theta D_{KL}\left(p_{data}(x) \mid p_\theta\right)$$

Generative modeling: some important properties

4. Generalization: new samples are different from the training data



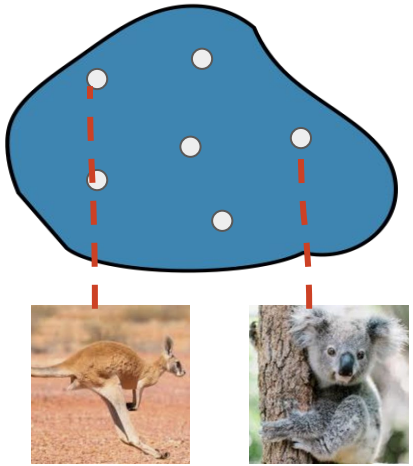
Same as training samples



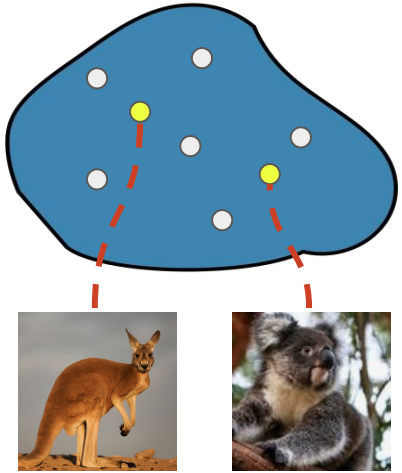
Different from training but ‘same distribution’

Generative modeling: some important properties

4. Generalization: new samples are different from the training data



Same as training samples

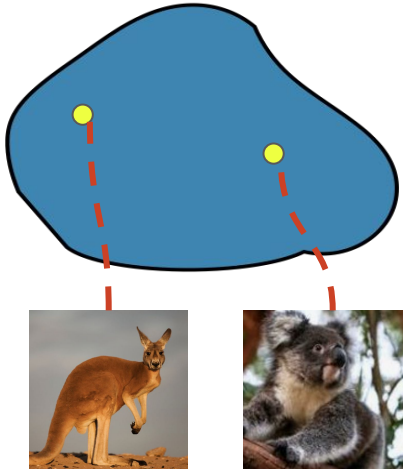


Different from training but ‘same distribution’

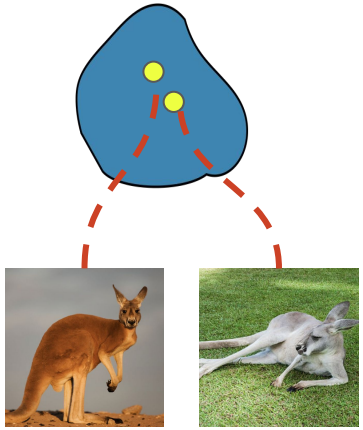
Generative modeling: some important properties

5. Conditioning can allow us to ‘tighten’ the distribution

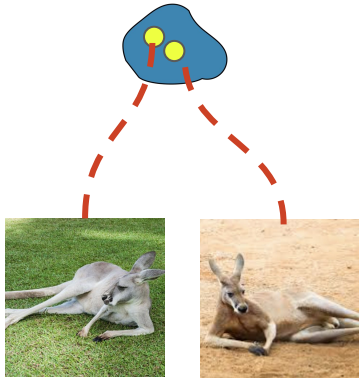
<no condition>



“kangaroo”



“Kangaroo lying down”’

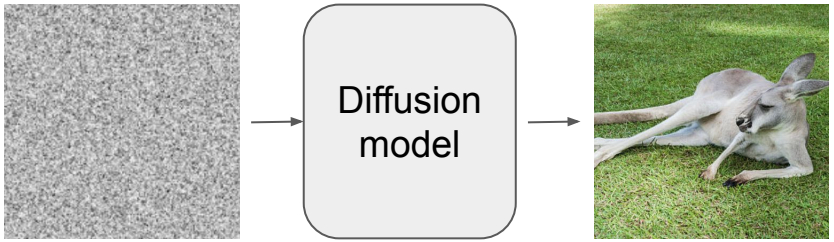


Related: in language models, writing a more specific prompt

Generative modeling: two popular types

Latent variable models (LVMs)

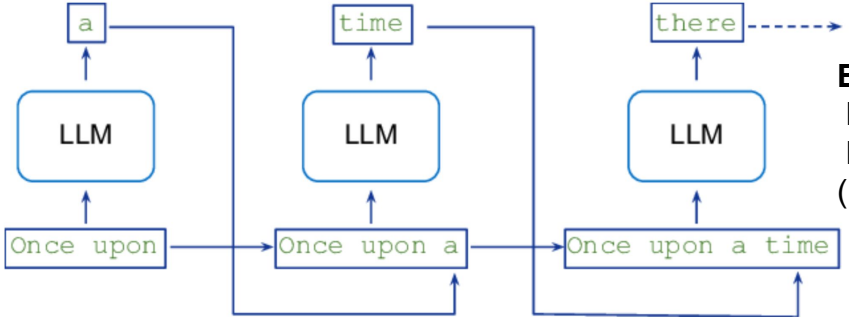
"Map noise to data"



Examples
Diffusion models
Flow models
Variational autoencoders
Generative Adversarial Nets (GANs)

Autoregressive models (ARs)

"Sample one piece at at a time"



Examples
Large language models (LLMs)
Multimodal large language models (mLLMs)

This lecture is about latent variable models like diffusion - are SOTA in image output
Next week is about autoregressive models like multimodal LLMs - SOTA for language output

A few examples of generative model tasks in images

Text-to-image

"A kangaroo lying down"'

Super-resolution

Image inpainting

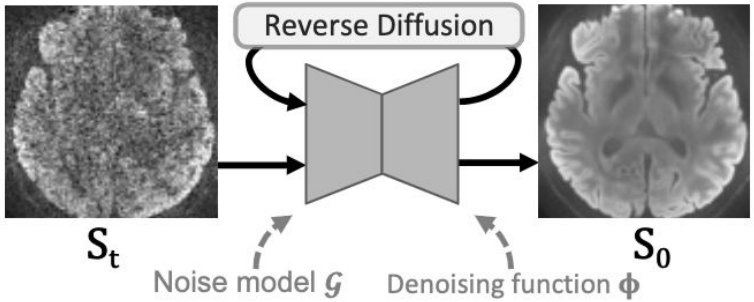
Image editing with text

"Turn it into a Western"'

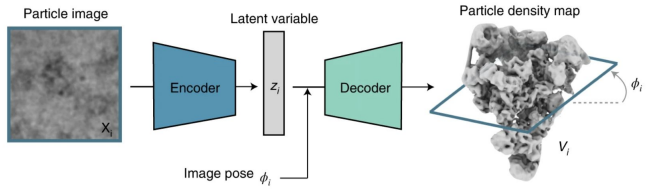
Notice how the output is a 'distribution'. Many reasonable outputs for one input.

A few examples of generative models in biomedicine

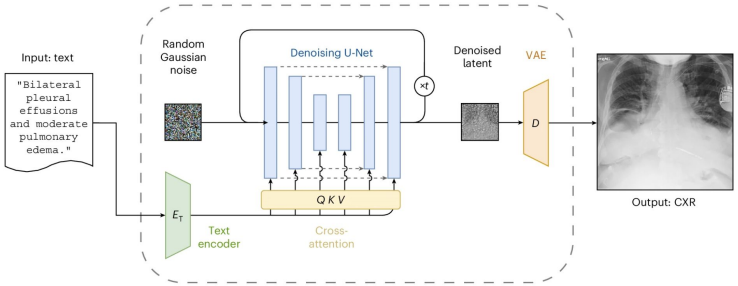
MRI denoising



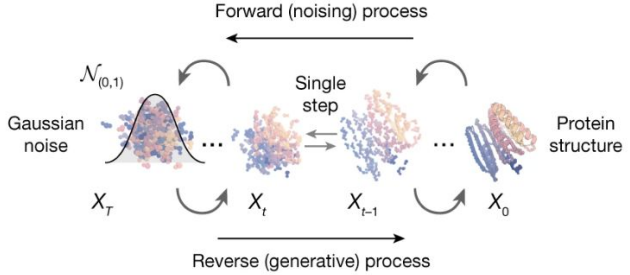
Protein reconstruction from EM



Synthetic data generation



De novo protein design



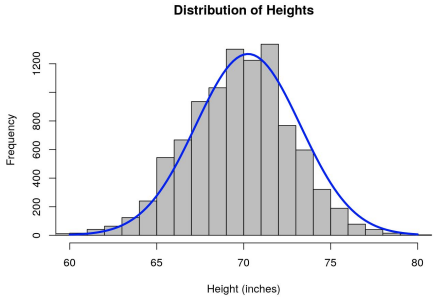
"Self-Supervised Diffusion MRI Denoising with Generative Diffusion Models"
"A vision–language foundation model for the generation of realistic chest X-ray images"

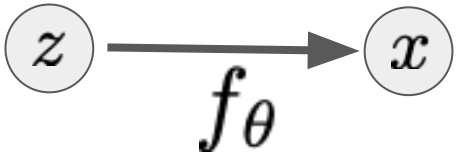
"CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks"
"De novo design of protein structure and function with RFdiffusion""

Sampling

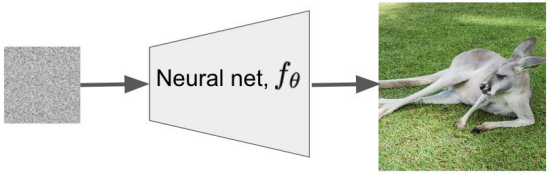
Sampling in three *Latent Variable Models*

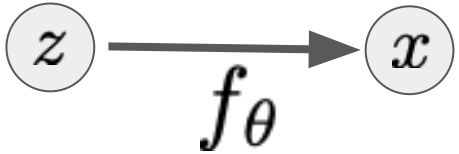
1D gaussian



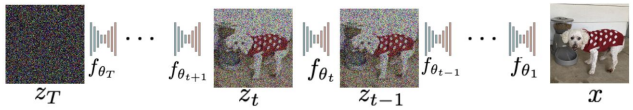


Variational Autoencoder (VAE)



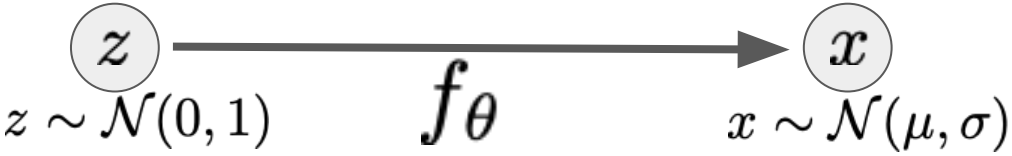


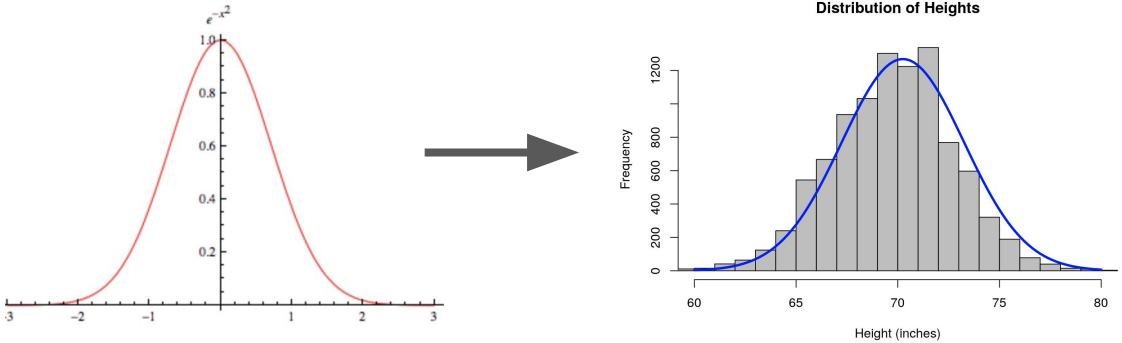
Diffusion model



Think about sampling in terms of random variables.

Sampling a simple 1d Gaussian

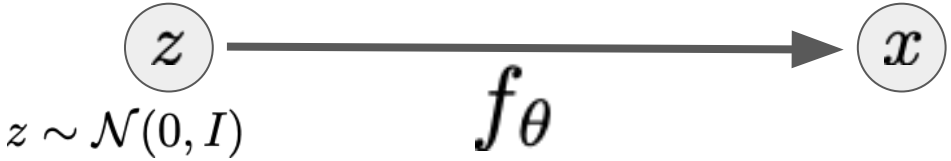


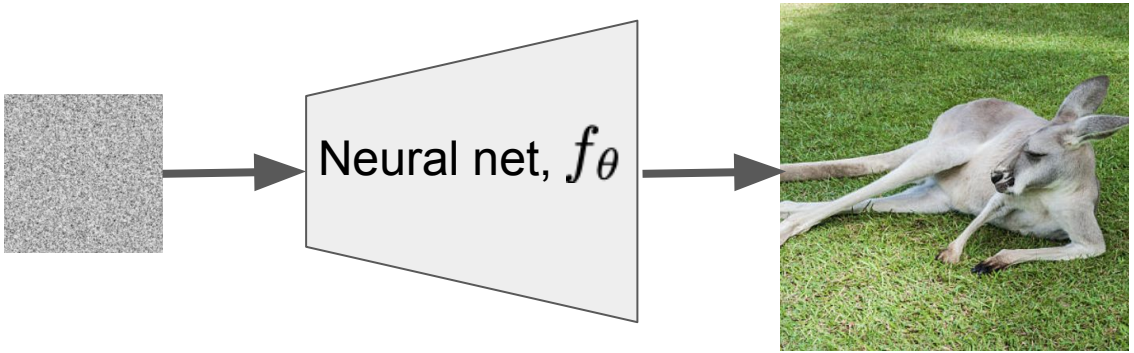


$$f_\theta(z) = \mu + \sigma \cdot z$$

Here, the noise-to-data function is very simple

Sampling a variational autoencoder

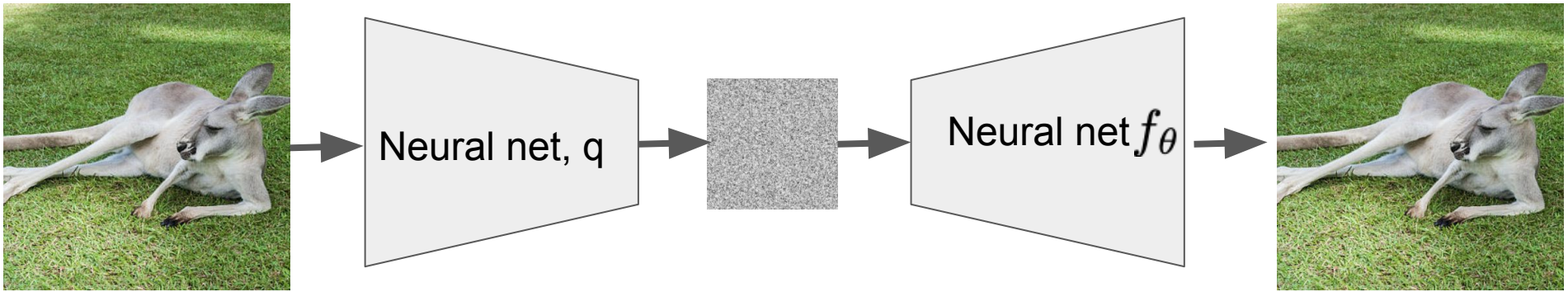




Architecture can be anything. A classic one is stacked CNN layers
Dimension of `z` typically smaller than `x`
Other models are sampled the same way, e.g. GANs

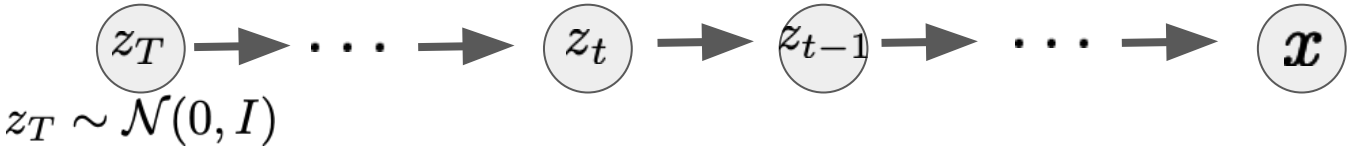
Aside: full VAE architecture

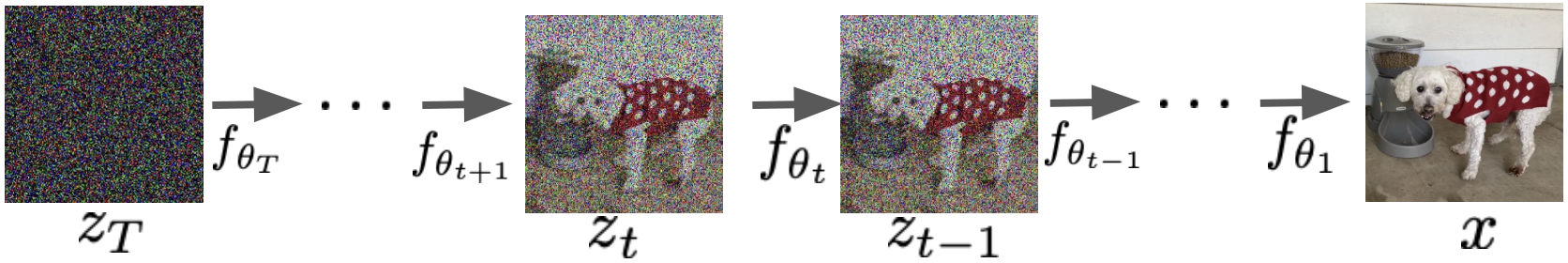
The full VAE architecture has a second neural net, but it is not used in sampling



Sampling a diffusion model

Noise to data with multiple steps

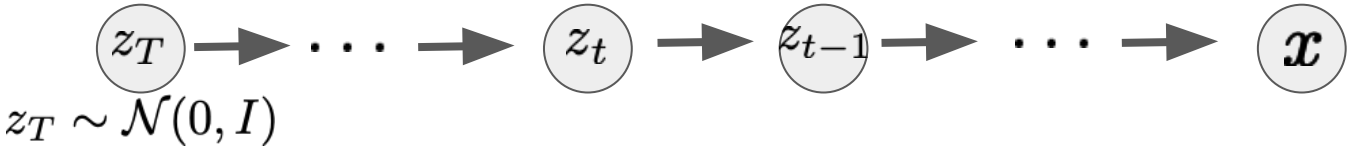


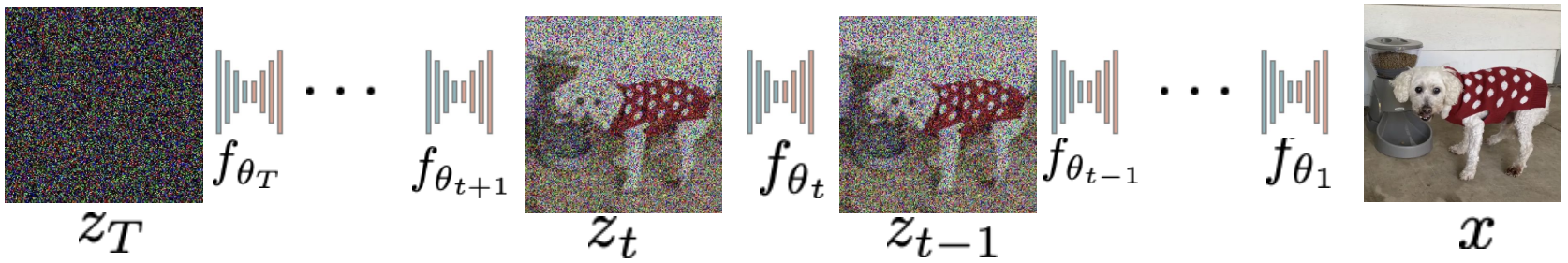


The number of timesteps, `T` is a design choice. A common choice is T=1000

Sampling a diffusion model

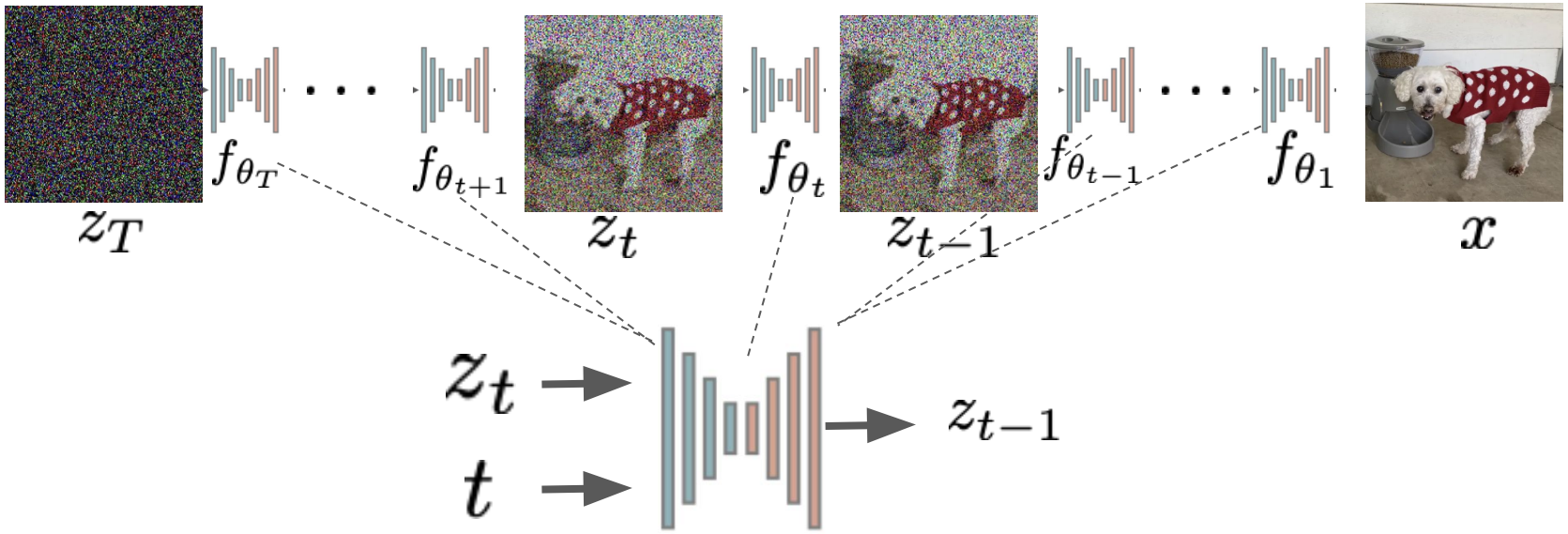
Noise to data with multiple steps





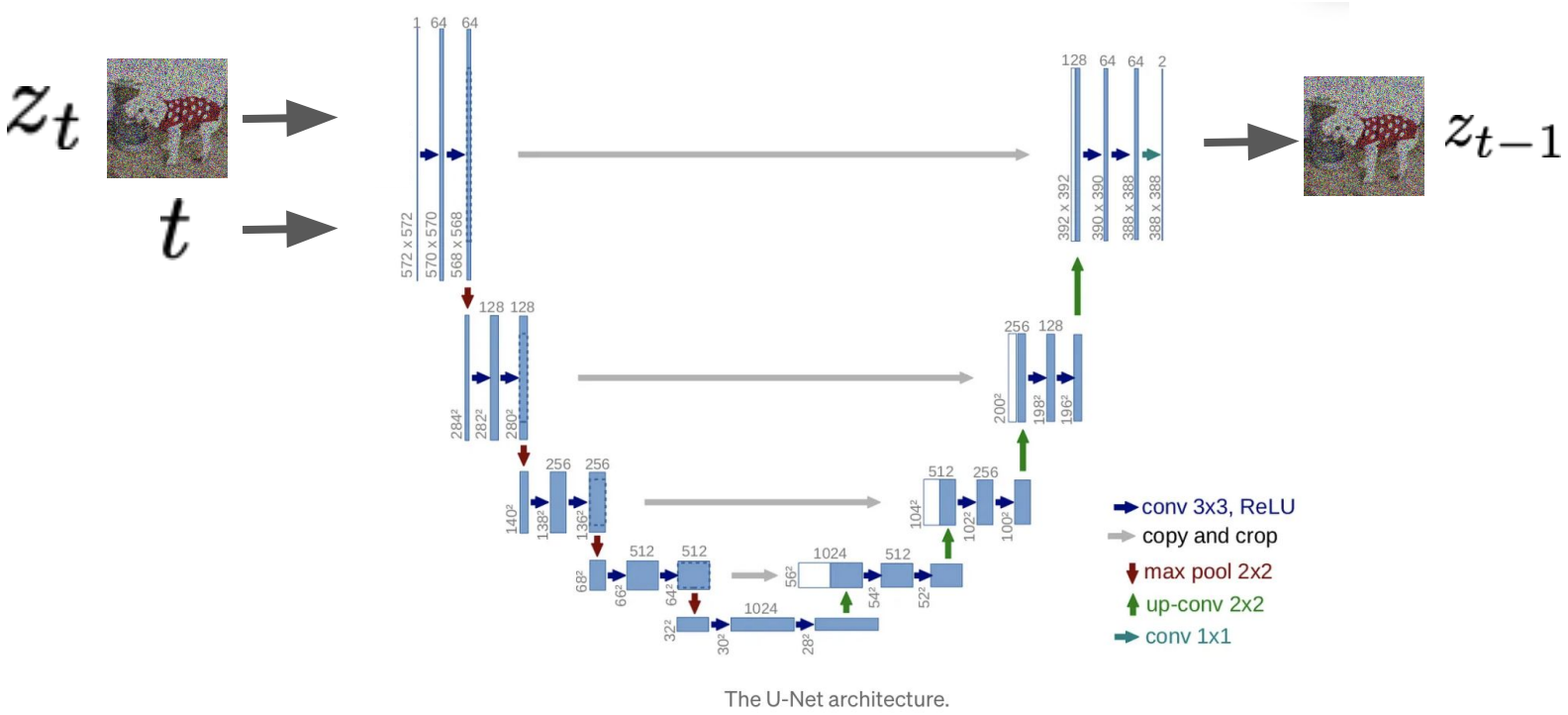
One neural net for each timestep, f_{θ_t}

Sampling a diffusion model



Use a single neural net for all steps, and condition on `t`
Parameter sharing makes learning easier, and reduces total # weights

Typical diffusion architecture: the UNet

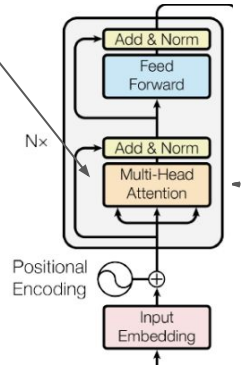


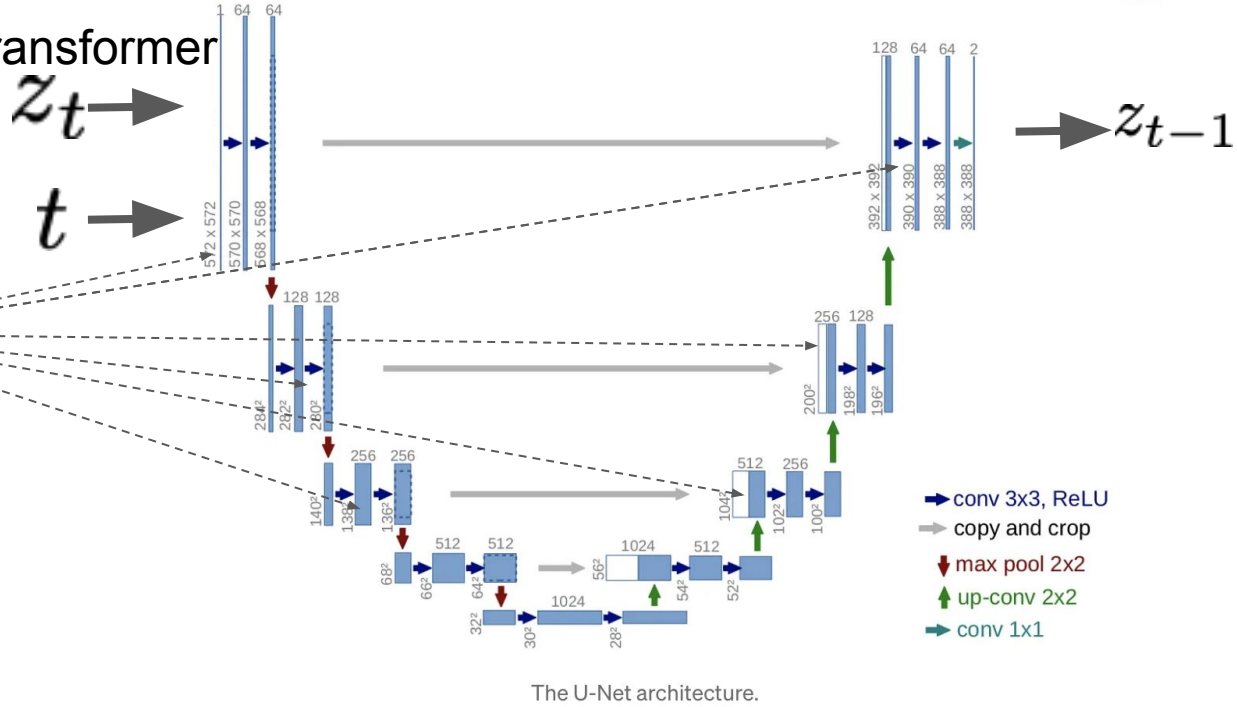
The U-Net architecture.

Intuitively, the bottom of the U-shape captures ‘higher level’ features

Typical diffusion architecture: the UNet

Use self-attention layers from transformer to each bock

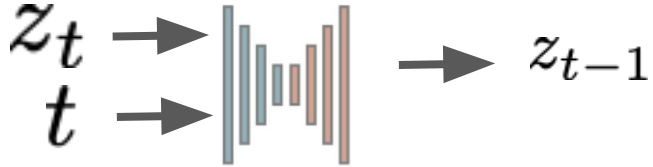




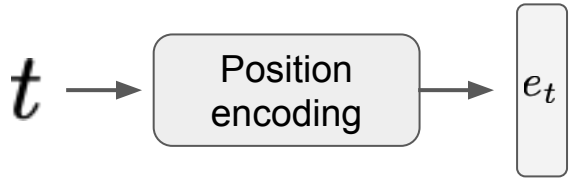
The U-Net architecture.

Typical diffusion architecture: UNet timestep condition

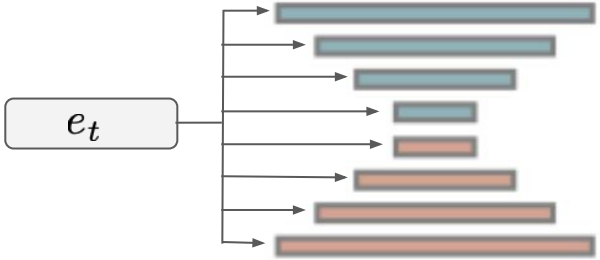
`t` is a scalar. How to pass it to the Unet?



Answer: positional encoding (like in transformer)



Typically *added* at each feature map layer



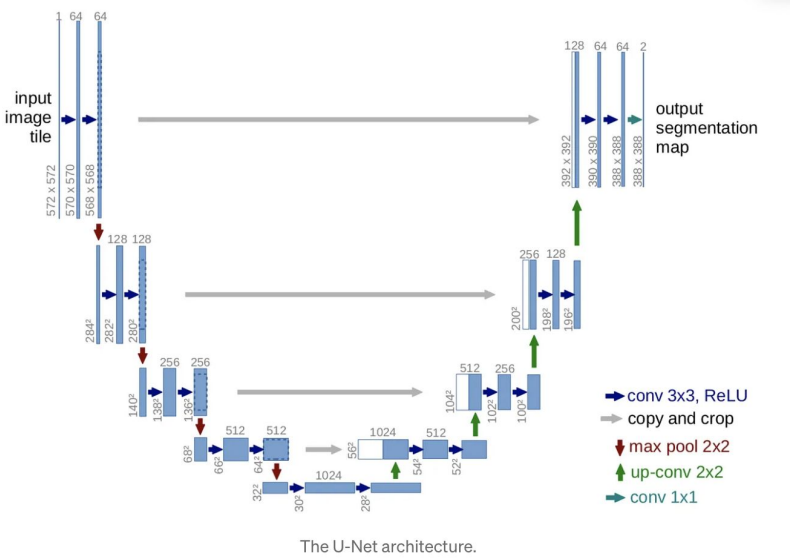
Not shown: timestep dimension *i* added to channel *i* in the feature map.
The e_t is projected to size *c* (the channel dimension) with a linear layer

Aside: UNets in diffusion are from biomedicine

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

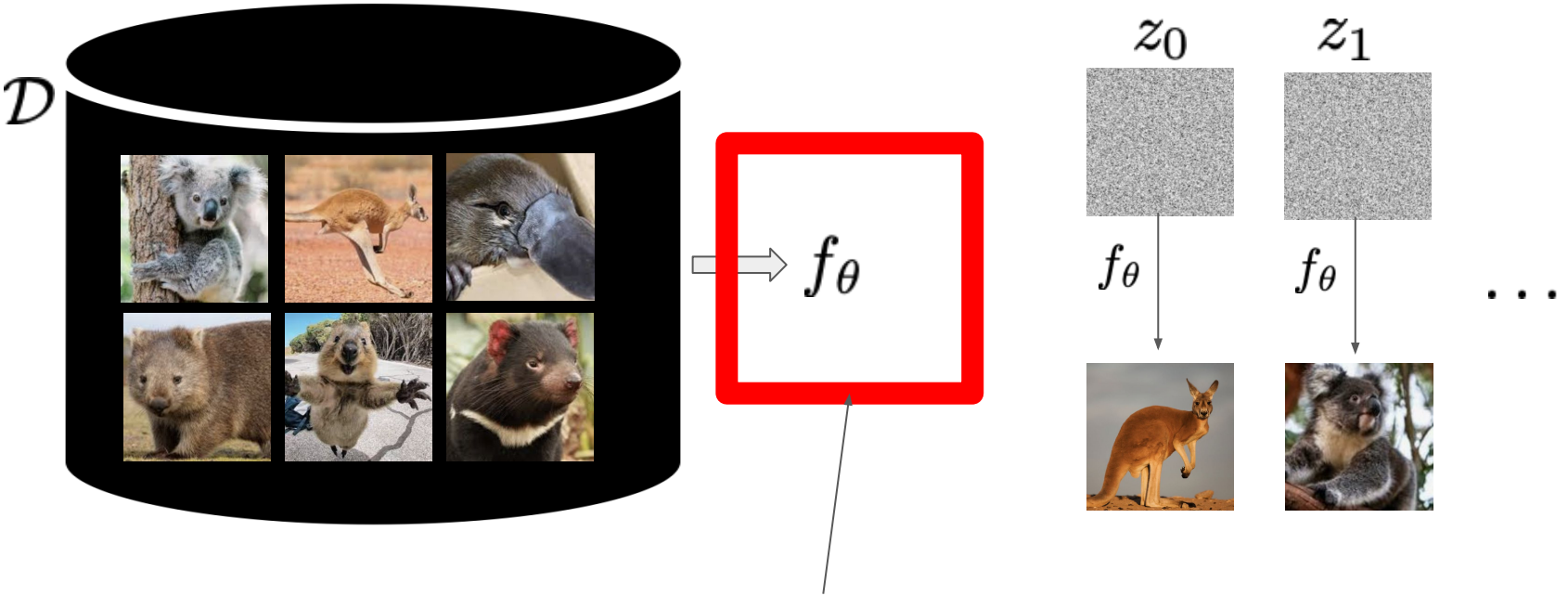


The U-Net architecture.

Unet architecture was created for biomedical segmentation.
It's a rare case of mainstream ML taking an idea from an application space

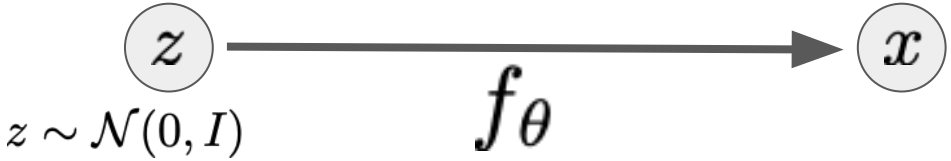
Training

Training: how to learn the noise-to-data map

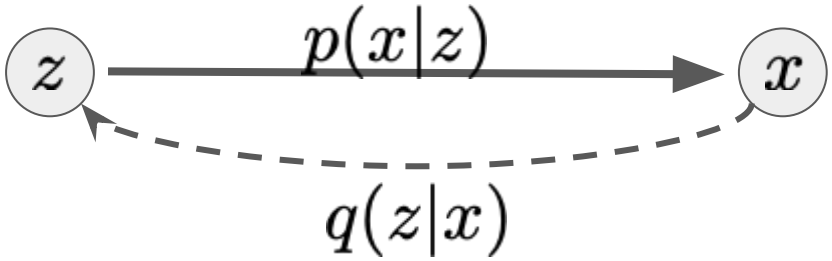


How to learn the parameters from data?

VAE training (this will help understand diffusion)



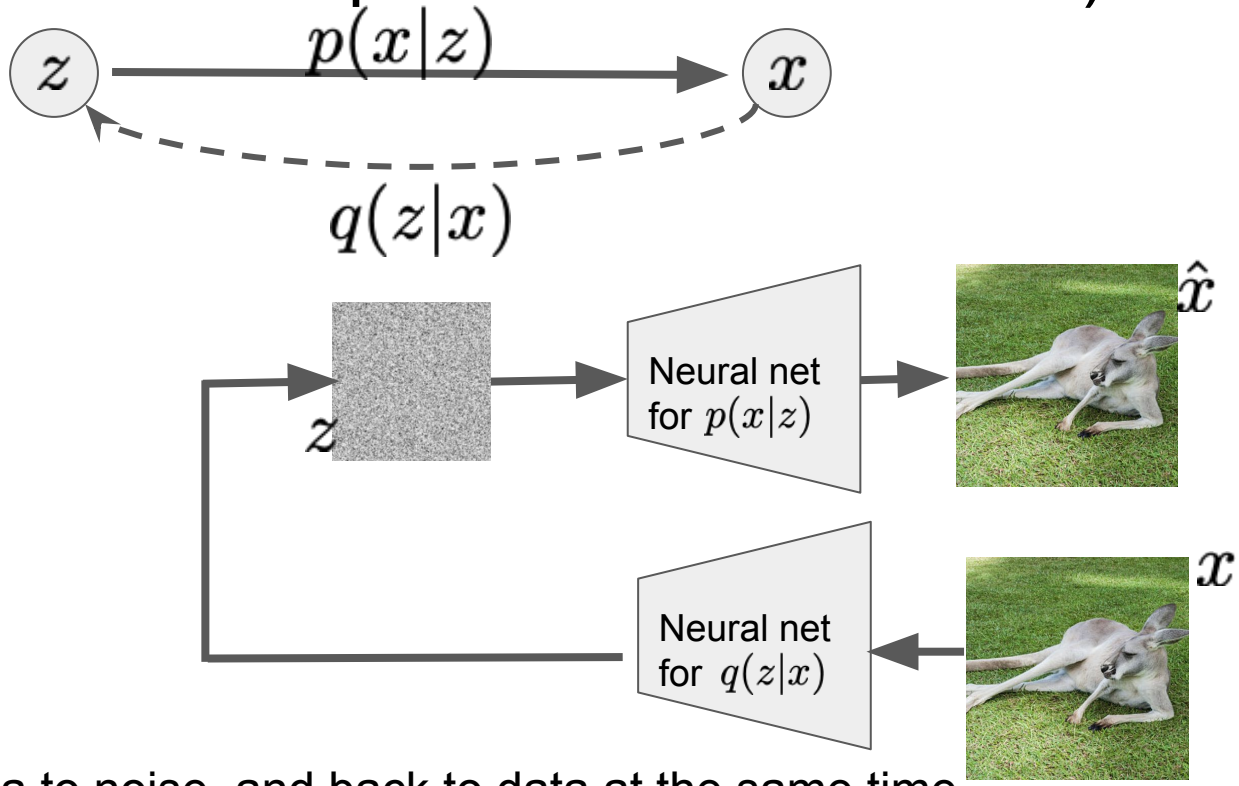
To model $p(x|z)$ we want pairs (z_i, x_i) but only have access to x_i



Idea: jointly learn to map the data to the noise AND the noise to data.

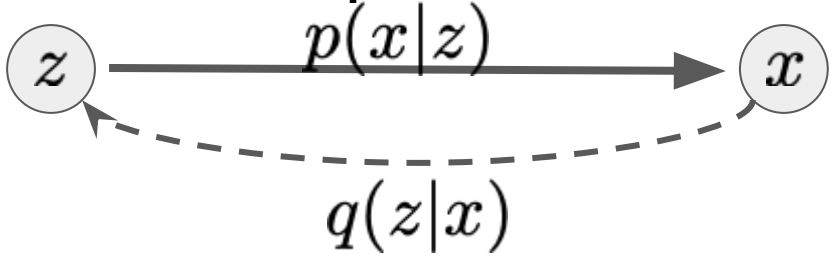
"Auto-Encoding Variational Bayes"

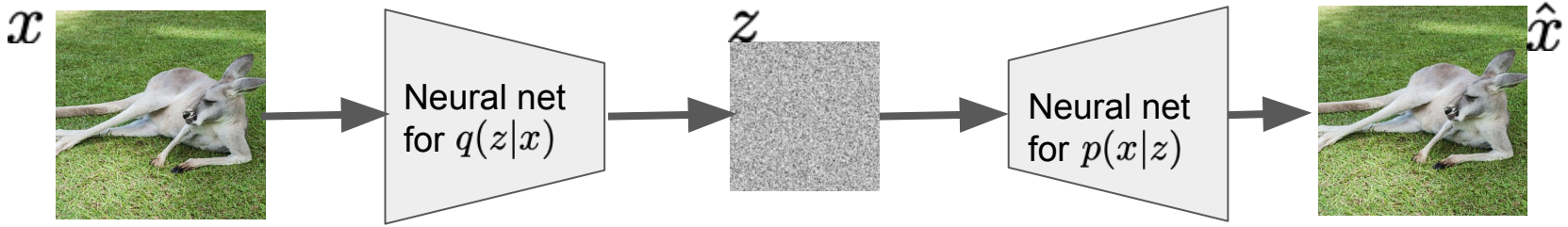
VAE training (this will help understand diffusion)



Idea: learn to map the data to noise, and back to data at the same time

VAE training (this will help understand diffusion)

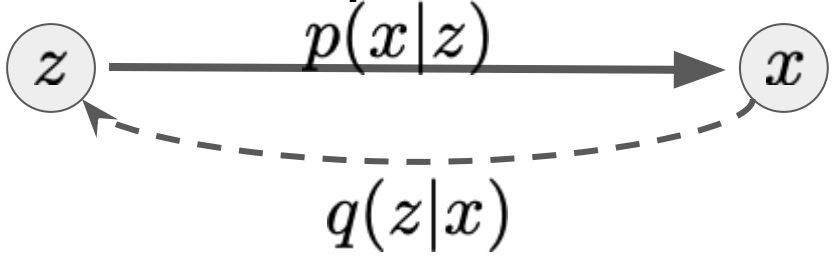


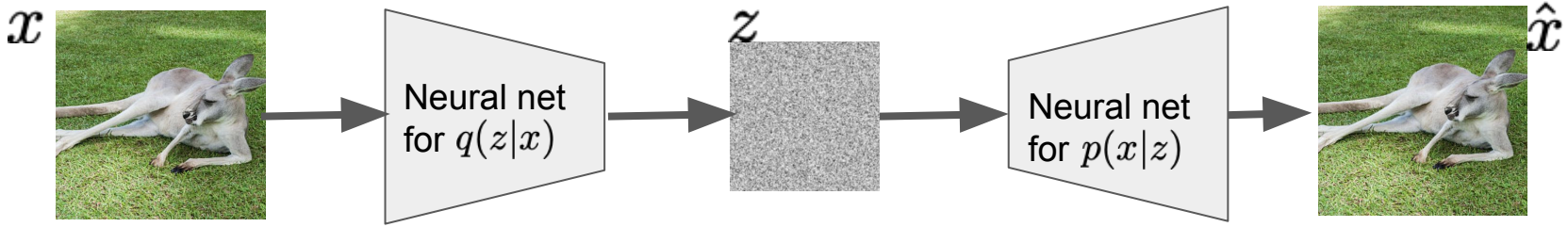


Idea: learn to map the data to noise, and back to data at the same time

*It's more standard to draw it like this

VAE training (this will help understand diffusion)

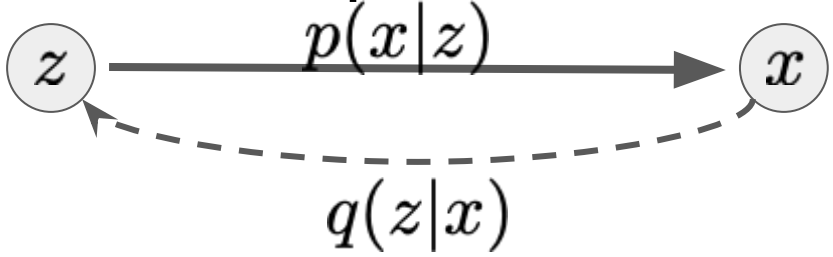


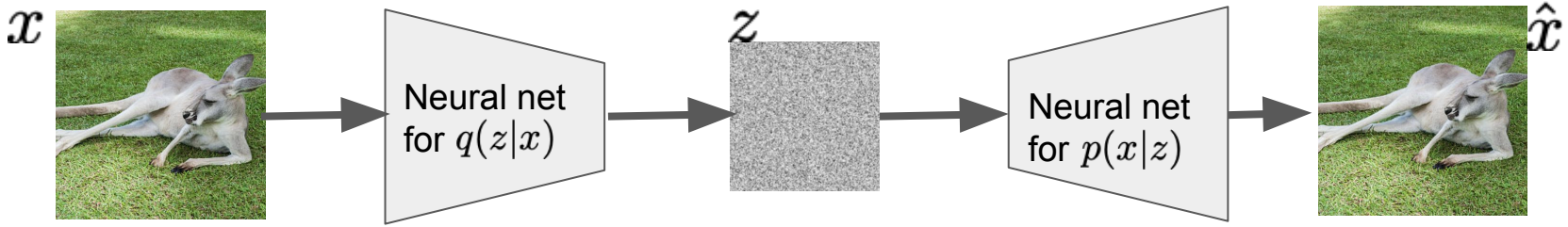


First loss term: reconstruction

$$L_{recon} = \mathbb{E}_{q(z|x)}[p(x|z)] = \|x - \hat{x}\|$$

VAE training (this will help understand diffusion)





First loss term: reconstruction $L_{recon} = \mathbb{E}_{q(z|x)}[p(x|z)] = \|x - \hat{x}\|$

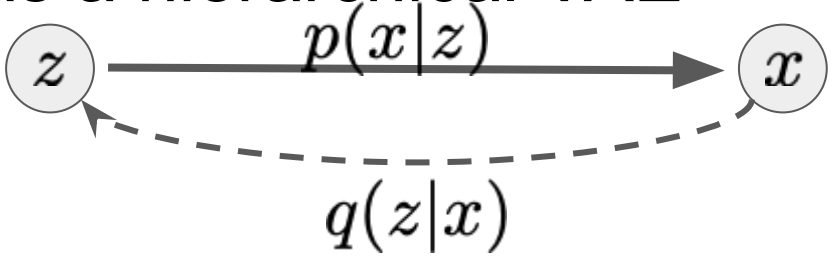
This can be viewed as ‘compression’ if z is a small vector

VAE training (this will help understand diffusion)

** the real VAE is more complicated than that
You can read details in this good blog post which also introduces diffusion models

Diffusion training as a hierarchical VAE

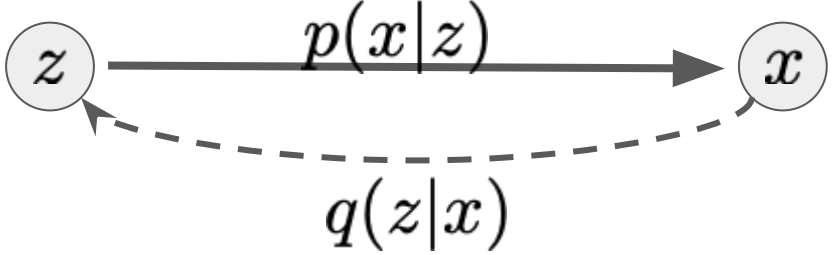
VAE graphical model



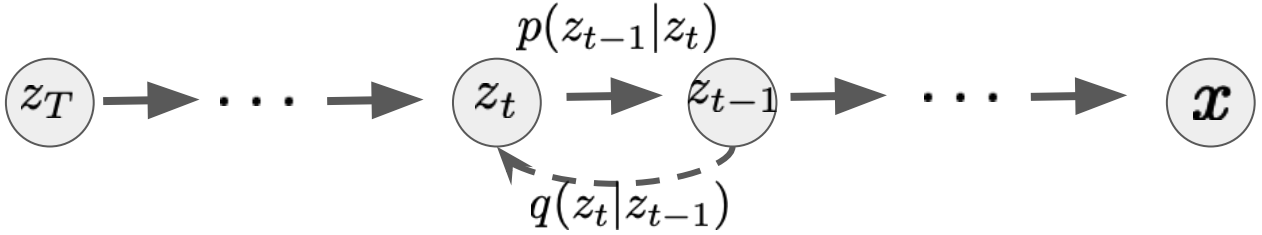
Diffusion models are like VAEs but with a few key differences

Diffusion training as a hierarchical VAE

VAE graphical model

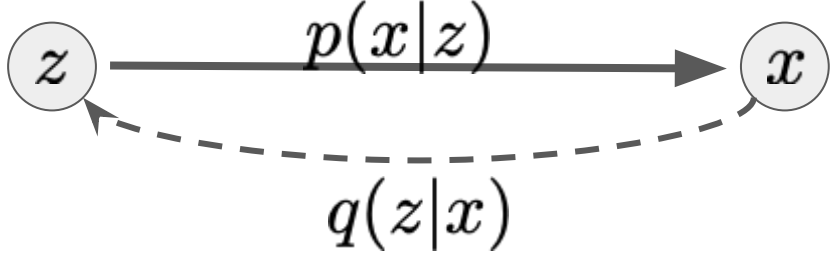


Key difference #1: diffusion has a hierarchy of latent variables

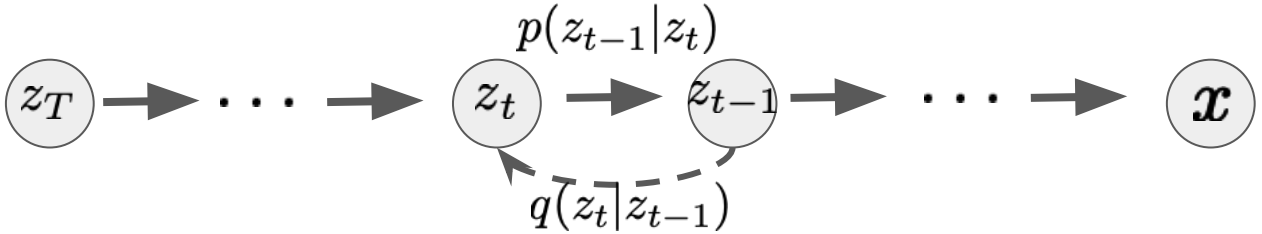


Diffusion training as a hierarchical VAE

VAE graphical model



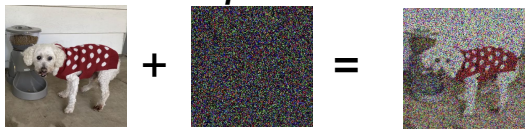
Key difference #1: diffusion has a hierarchy of these vars



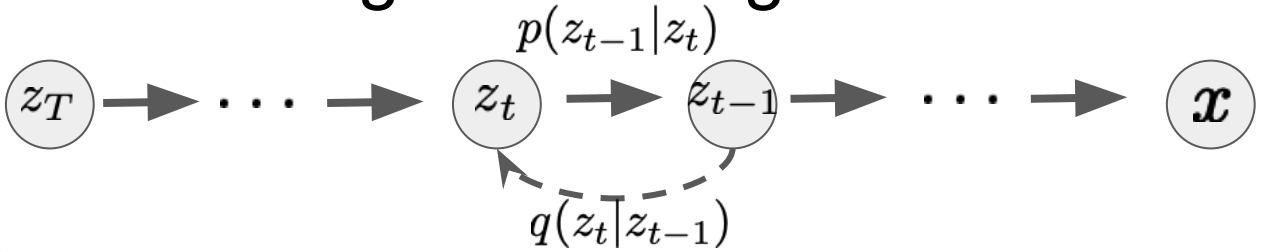
Key difference #2: the $q(z_t|z_{t-1})$ is a *fixed Gaussian diffusion process:*

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{\alpha_t} z_{t-1}, (1-\alpha_t) I)$$

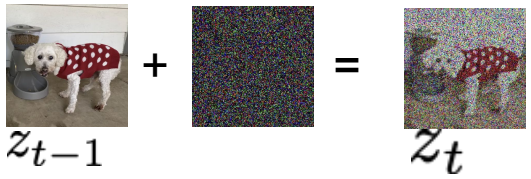
*and α_t is a parameter



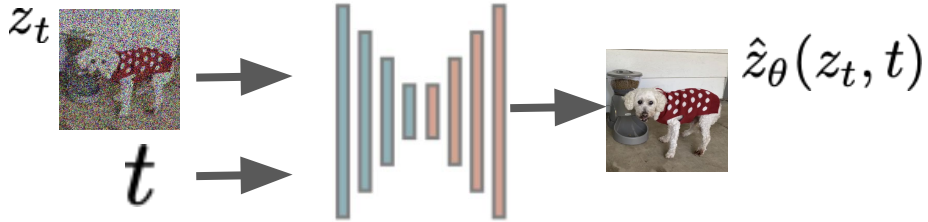
Diffusion training as learning to reverse noising



Sample t,
Apply noise: $q(z_t|z_{t-1})$



Use Unet to reverse the noise



Reconstruction loss

$$\mathbb{E}_{t\sim U(2,T)}\left[c(\alpha_t)\|\hat{z}_\theta(z_t,t) - z_{t-1}\|\right]$$

A sketch of how to derive VAE and diffusion

Matching the data distribution leads to max likelihood principle

$$\text{argmin}_\theta \; D_{KL}\left(p_{data} \mid p_\theta\right) = \text{argmin}_\theta \; E_{x \sim p_{data}}\left[\log p_\theta(x)\right]$$

Introduce a latent variable, z

$$\begin{aligned}
\log p_\theta(x) &= \log \int_z p_\theta(x, z) \\
&= \log \int_z p_\theta(z) p_\theta(x|z) dz \\
&\leq \int_z \log p_\theta(z) p_\theta(x|z) dz
\end{aligned}$$

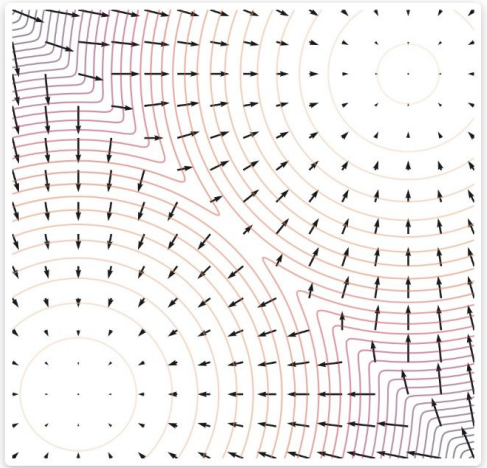
More steps:

- Notice p(z|x) is hard to estimate - introduce variational q(z|x).
- Force each distribution to be Gaussian, with parameters predicted by neural nets.
- Show that the lower bound is a good lower bound
- Do some symbolic manipulations so that all terms are tractable to compute

A good blog post: Understanding Diffusion Models: A Unified Perspective

“Score based models” are similar to diffusion

Key point: score based modeling is fundamentally the same as diffusion modeling.
The score-based modeling perspective is interesting



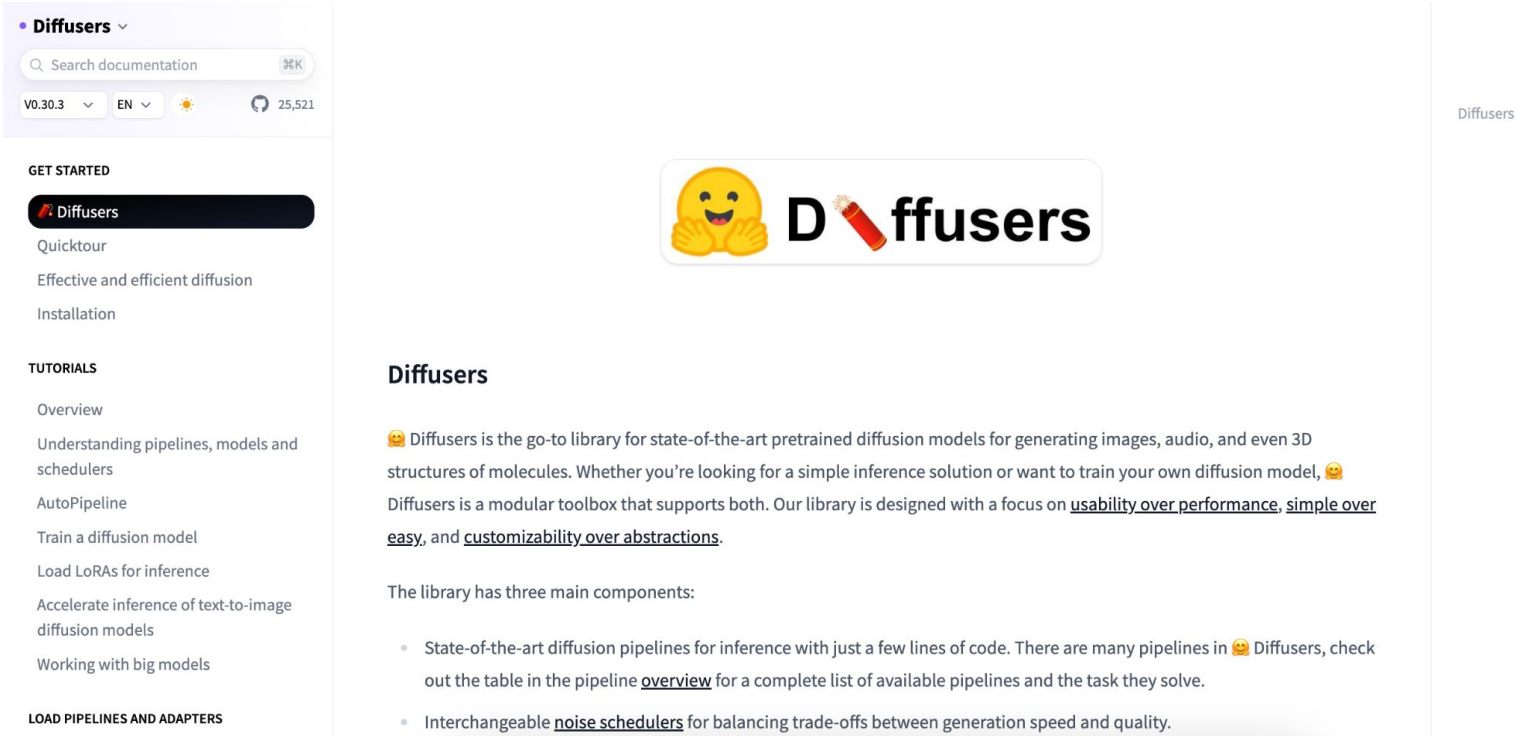
Assume the data space is 2d.

Contours are probability density values, $p(x)$
Arrows are the *score*: $\nabla_x \log p(x)$

Sampling: start somewhere random in space.
Then follow the arrows until you reach a high density area.

Good blog post: [Generative Modeling by Estimating Gradients of the Data Distribution]

Implementing Diffusion models - 🤗 *diffusers* library

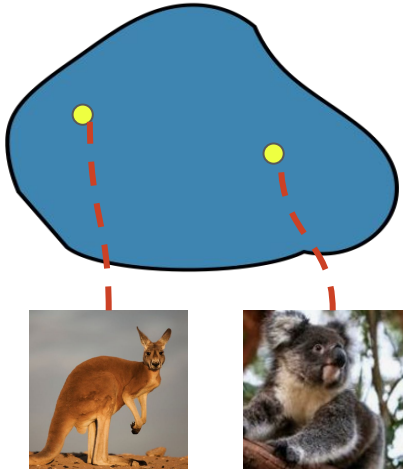


More advanced sampling

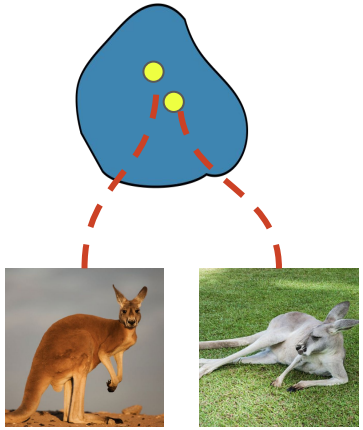
Conditional sampling

Recall: we want to be able to add conditioning information

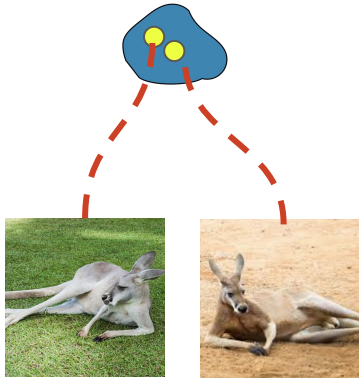
<no condition>



“kangaroo”



“Kangaroo lying down”’

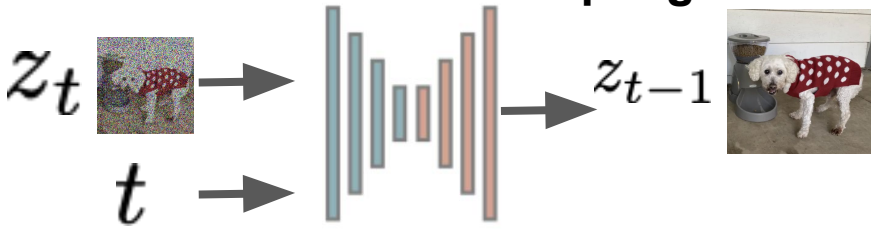


Conditional sampling

Solution: pass the condition info to the Unet

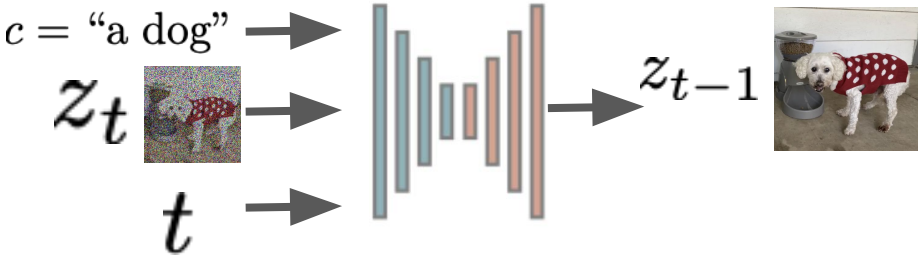
Te loss is almost the same

Before: unconditional sampling



$$\mathbb{E}_{t\sim(2,T)}\left[C(\alpha_t)\cdot\|\hat{z}_\theta(z_t,t)-z_t\|\right]$$

After: with a condition



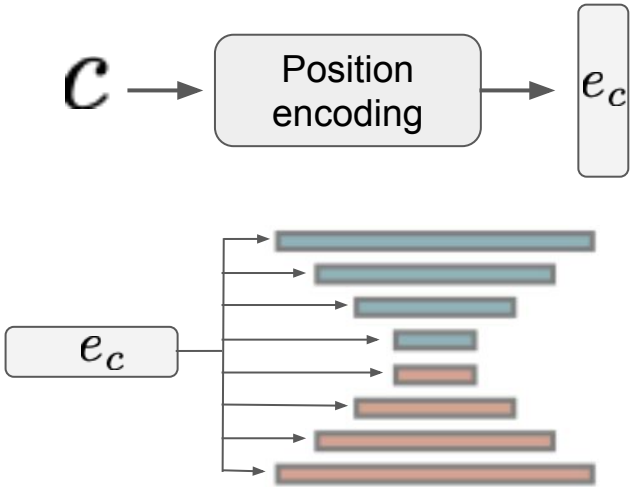
After

$$\mathbb{E}_{t\sim(2,T)}\left[C(\alpha_t)\cdot\|\hat{z}_\theta(z_t,t,c)-z_t\|\right.$$

Implementing conditional sampling

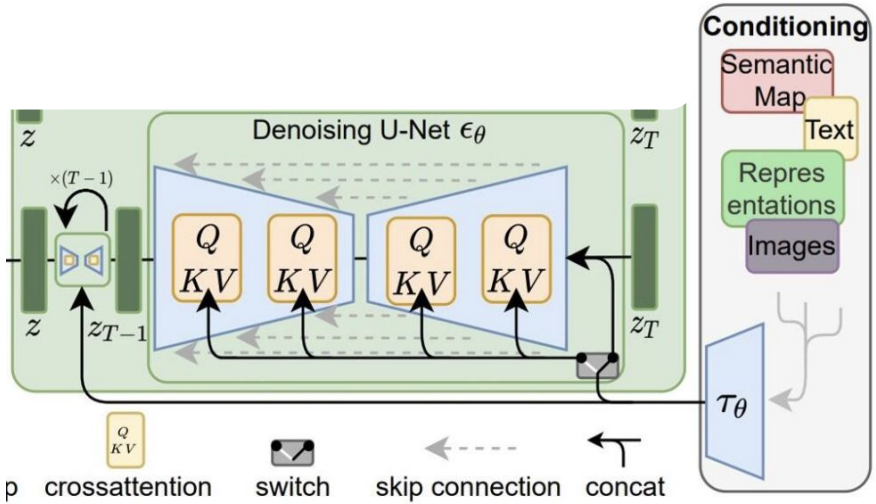
Class-based conditioning

Encode the same way `t` is encoded



General conditioning via cross-attention

Much more common



Conditional sampling in practice with “guidance”

Sampling (using ‘score matching’ perspective)
We move in the gradient direction

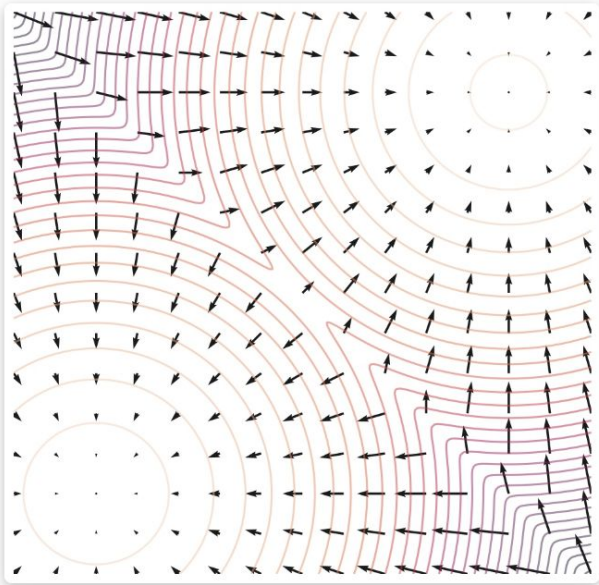
$$\nabla_x \log p(x)$$

Now we do conditioning:

$$\nabla_x \log p(x|c)$$

Guidance says:
“let’s balance the unconditional and conditional directions”

$$\nabla_x \log p(x) + \gamma \cdot \nabla_x \log p(x|c)$$



Good blog post: [Guidance: a cheat code for diffusion models]

Conditional sampling in practice with “guidance”

The γ parameter lets us trade off diversity and fidelity

c=“*A stain glass window of a panda eating bamboo*”

Better diversity - smaller γ

Better fidelity - bigger γ

Good blog post: [Guidance: a cheat code for diffusion models]

Conditional sampling in practice with “guidance”

Guidance has an unconditional and conditional term

$$\nabla_x \log p(x) + \gamma \cdot \nabla_x \log p(x|c)$$

Problem: we train $\nabla_x \log p(x|c)$ so how to get $\nabla_x \log p(x)$

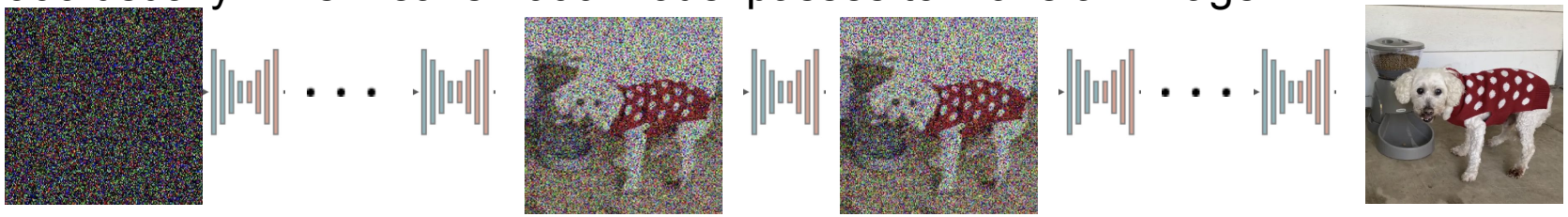
Answer: Use the same model, but let ‘unconditional’ be a class
$p(x) = p(x|c = \emptyset)$

In training, for 10% of samples, set the condition to this.

Good blog post: [Guidance: a cheat code for diffusion models]

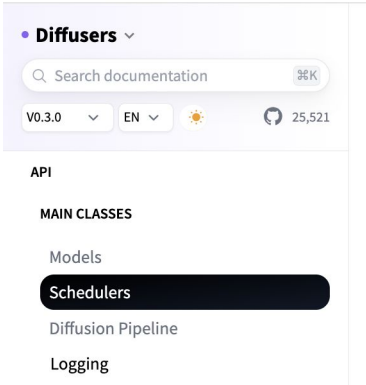
Faster sampling - schedulers

T=1000 usually. This means 1000 model passes to make an image



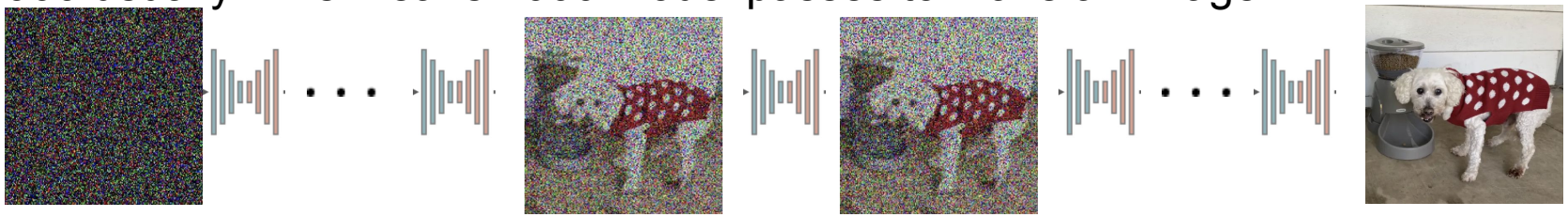
Different *samplers* (like DPMSolver) can reduce to 20 steps without changing the model

In diffusers library, look up “schedulers”

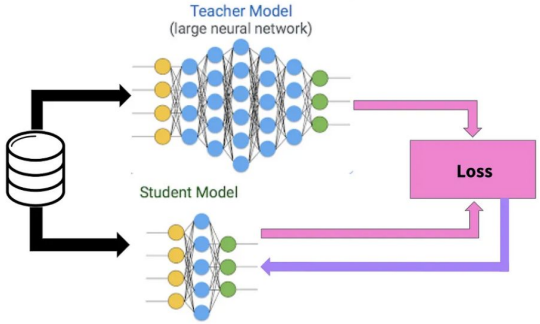


Faster sampling - distillation

T=1000 usually. This means 1000 model passes to make an image



Classic distillation: big model teaches smaller model



Diffusion distillation: High ‘T’ model teaches low ‘T’ model

See blog post [The paradox of diffusion distillation]

Important diffusion model designs

DDPM

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

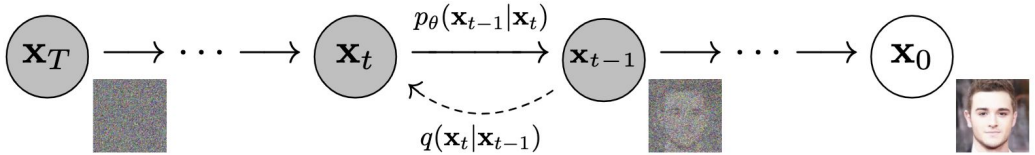
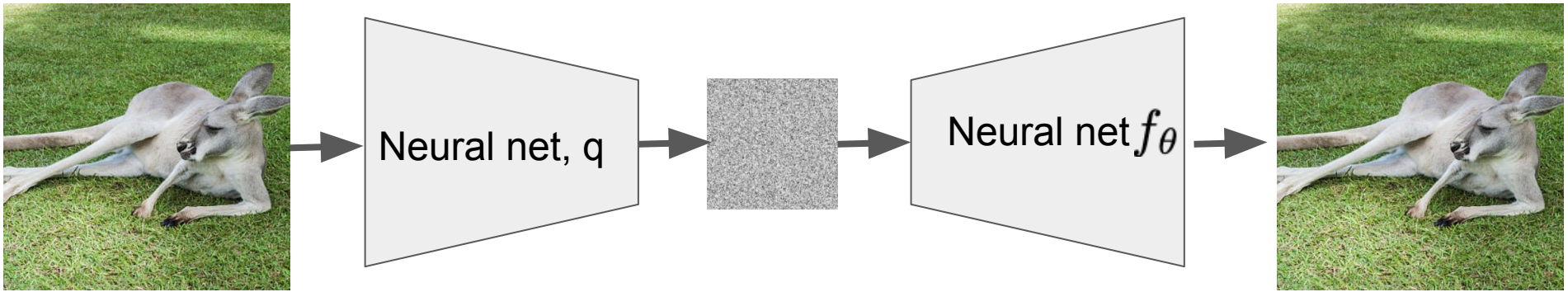


Figure 2: The directed graphical model considered in this work.

DDPM is *mostly* the model we have described up till now

Stable Diffusion or Latent Variable Models

Recall: VAEs can do data compression and reconstruction

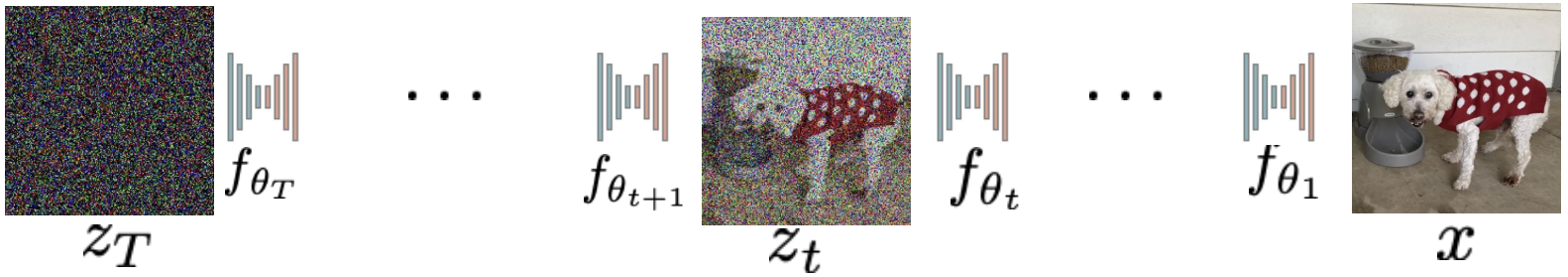


"High-Resolution Image Synthesis with Latent Diffusion Models"
"Score-based Generative Modeling in Latent Space"

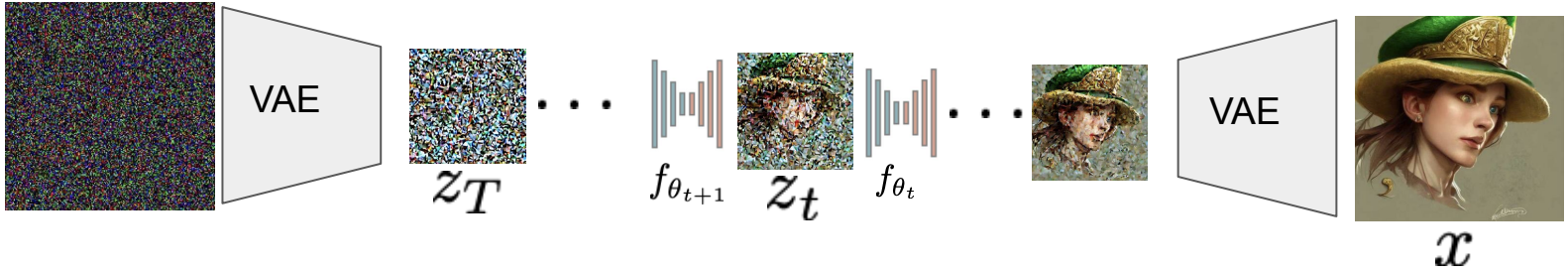
Stable Diffusion or Latent Variable Models

Idea: instead of diffusion models on *pixels*, do diffusion in a *compressed latent space*

Diffusion (DDPM)



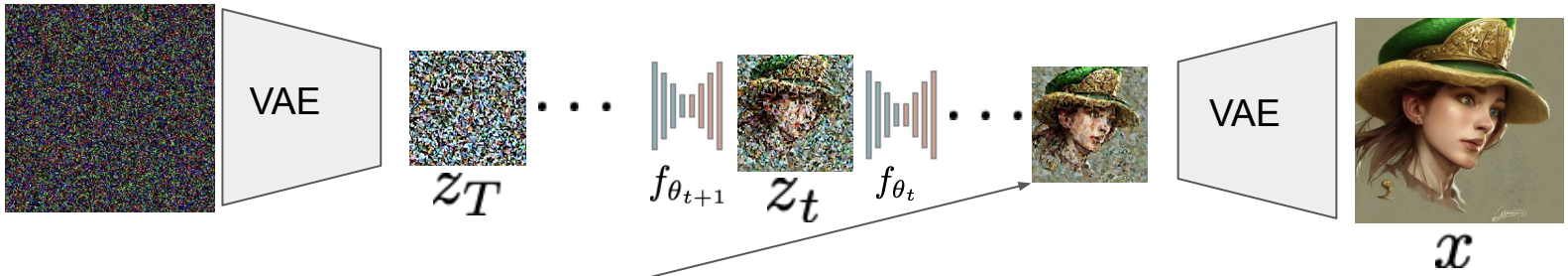
Latent diffusion (Stable Diffusion)



"High-Resolution Image Synthesis with Latent Diffusion Models"
"Score-based Generative Modeling in Latent Space"

Stable Diffusion or Latent Variable Models

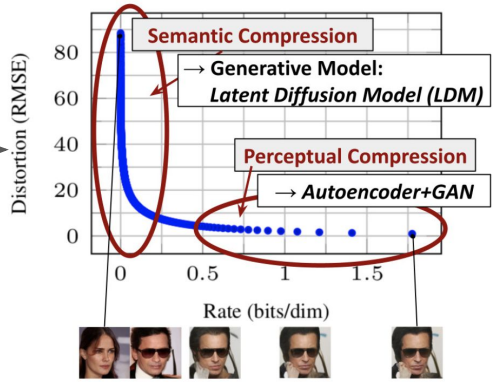
Idea: instead of diffusion models on *pixels*, do diffusion in a *compressed latent space*



Choose a VAE with low distortion

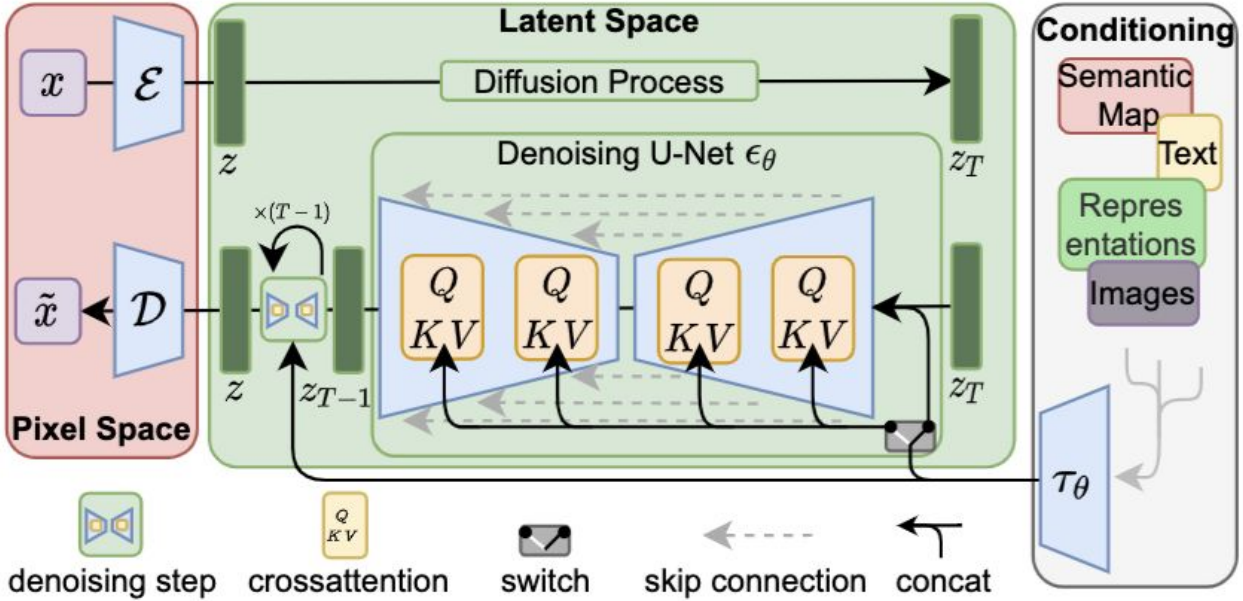
Key insight: VAE can downscale images a lot while maintaining semantic information.

Diffusion in latent is much more efficient



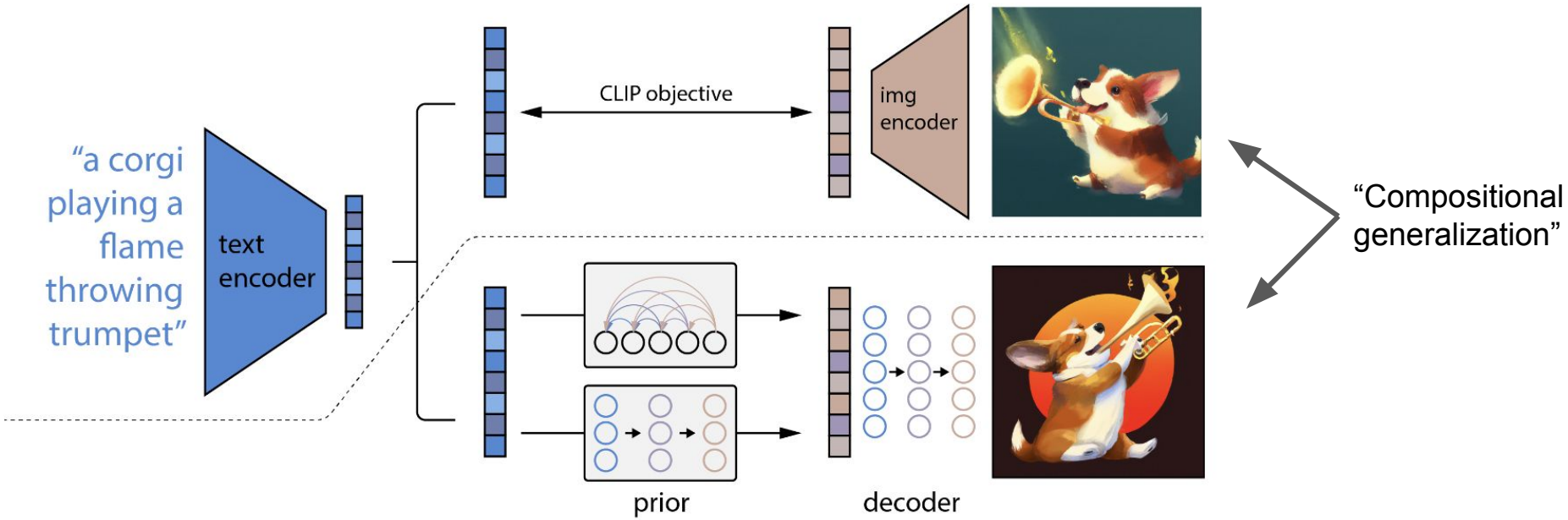
"High-Resolution Image Synthesis with Latent Diffusion Models"
"Score-based Generative Modeling in Latent Space"

Stable Diffusion or Latent Variable Models



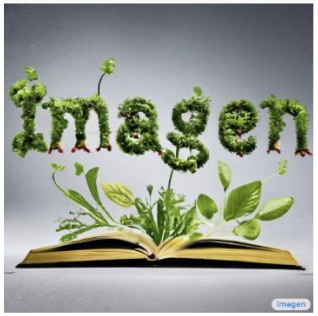
"High-Resolution Image Synthesis with Latent Diffusion Models"
"Score-based Generative Modeling in Latent Space"

DALLE-2



"Hierarchical Text-Conditional Image Generation with CLIP Latents"

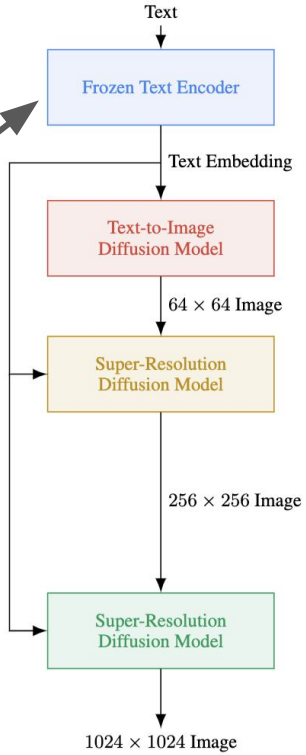
Imagen

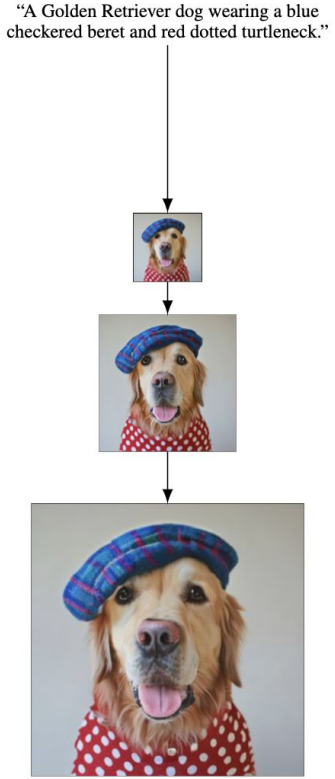


Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

Teddy bears swimming at the Olympics 400m Butterfly event.

Pretrained language-only model (others use CLIP language encoder)



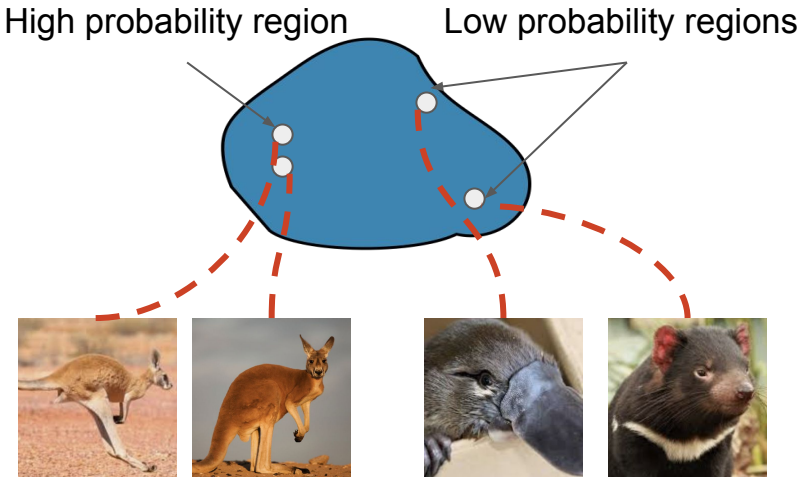


"Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding"

Evaluation

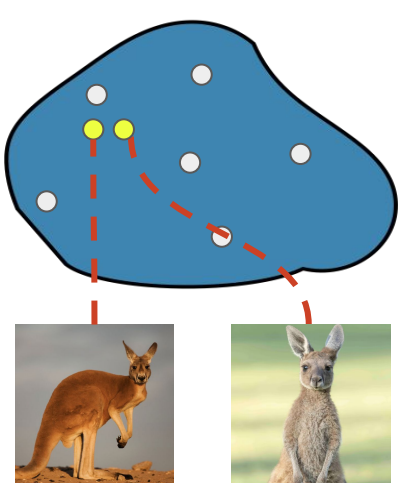
Evaluation

Desiderata for diffusion models

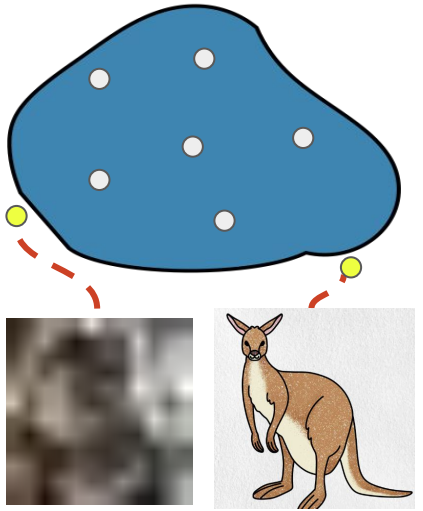


Sampling likelihood should match the data likelihood

Common issues



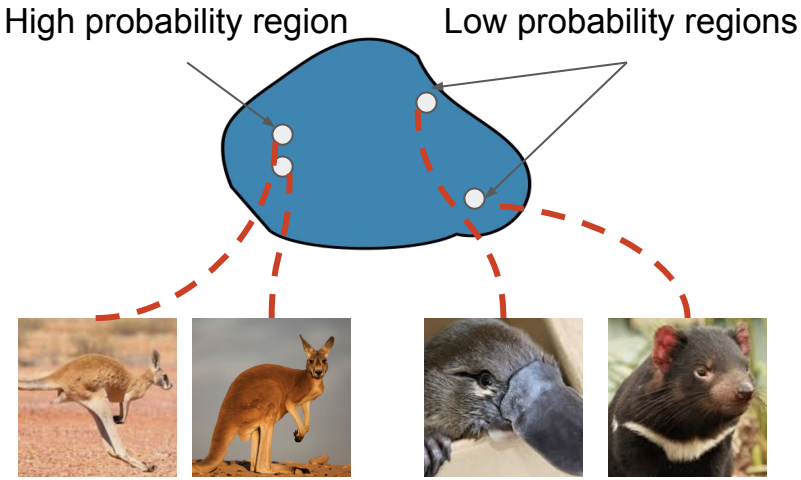
Diversity: not sampling the whole distribution



Fidelity: samples not from the right distribution

Evaluation

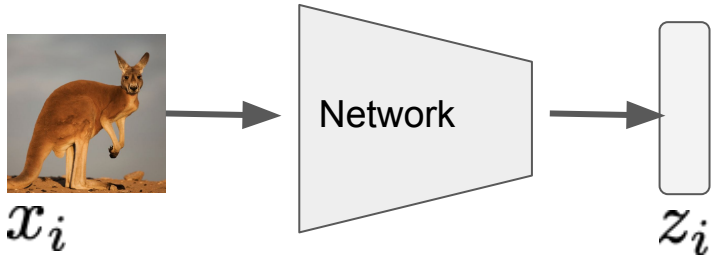
Desiderata for diffusion models



Sampling likelihood should match the data likelihood

Challenge: how to evaluate distribution in this “semantic latent space”

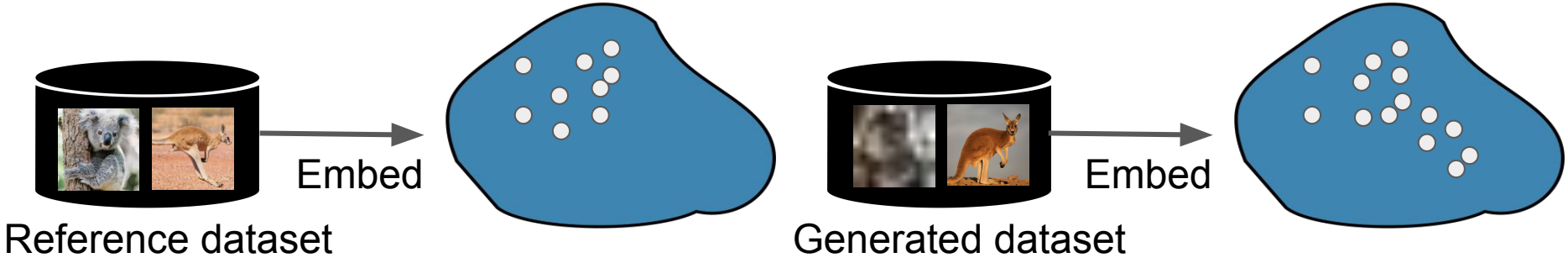
Approach: embed with a pretrained network, e.g. ‘inception network’ trained on ImageNet



*choice of network will depend on data
E.g. MRI images should not use networks trained on ImageNet

Evaluation: Frechet Inception Distance (FID)

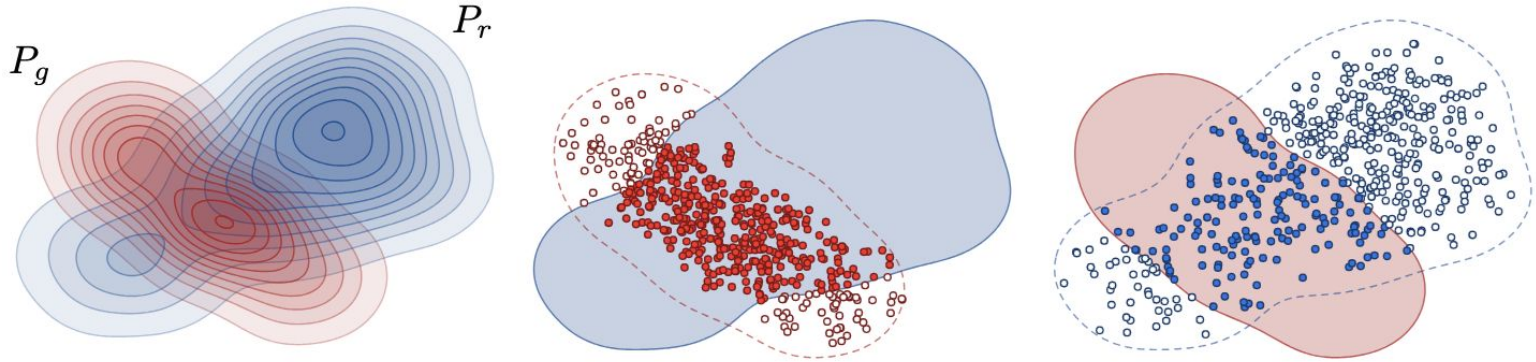
Idea: generated samples should match a target distribution, e.g. the ImageNet test set



1) Map dataset and embeddings to embeddings
2) fit a Gaussian to each distribution of embeddings
3) Measure a distance between those distributions

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu-\mu'\|_2^2 + \mathrm{tr}\left(\Sigma+\Sigma'-2(\Sigma\Sigma')^{\frac{1}{2}}\right)$$

Evaluation: precision and recall



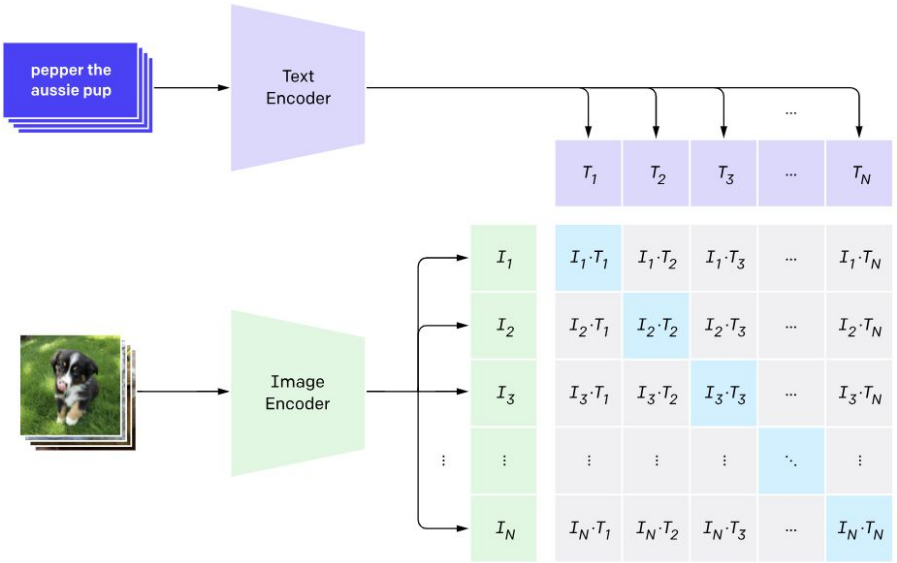
(a) Example distributions (b) Precision (c) Recall

Figure 1: Definition of precision and recall for distributions [25]. (a) Denote the distribution of real images with P_r (blue) and the distribution of generated images with P_g (red). (b) Precision is the probability that a random image from P_g falls within the support of P_r. (c) Recall is the probability that a random image from P_r falls within the support of P_g.

"Improved Precision and Recall Metric for Assessing Generative Models"

Evaluation: CLIP score

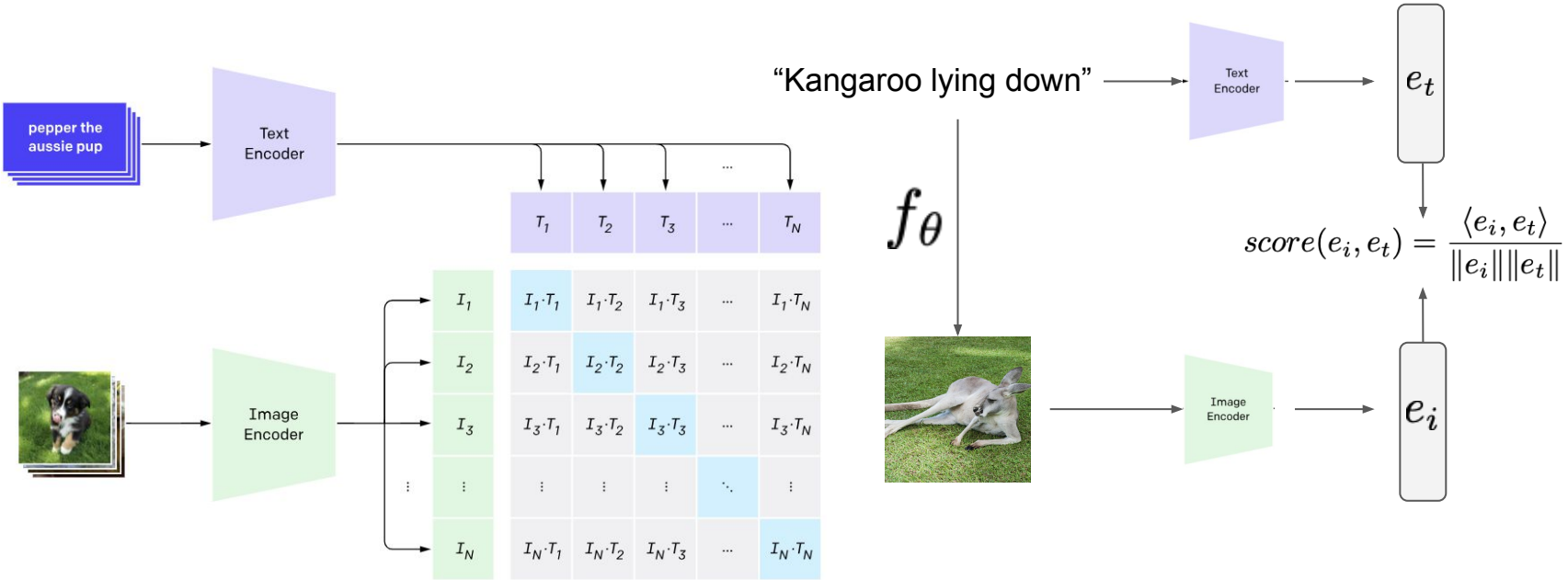
Idea: image should be close to its text prompt in CLIP space



"CLIPScore: A Reference-free Evaluation Metric for Image Captioning"

Evaluation: CLIP score

Idea: image should be close to its text prompt in CLIP space



"CLIPScore: A Reference-free Evaluation Metric for Image Captioning"

Evaluation: compositional generalization

People are interested in composing concepts in novel ways in diffusion models

Concept categories: object | texture | color | shape | style | number | spatial | size

A small pink duck stands on a hill with a metallic texture. The image is photorealism.

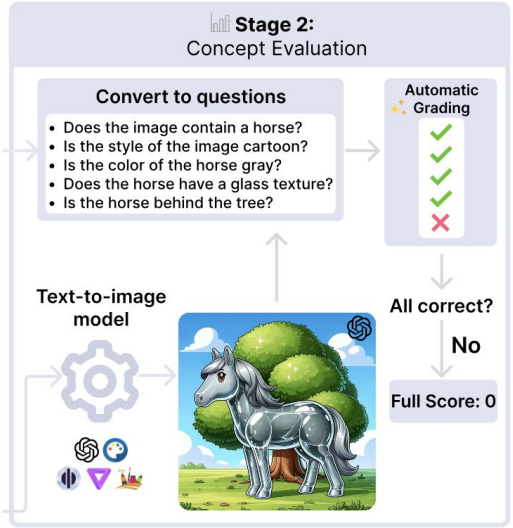
k=5

Four tiny, rectangular sushi pieces with a glass-like texture are positioned behind a tray in an expressionist style.

k=6

In a cubism style, four tiny, gray, glass-textured, rectangular laptops are positioned on top of a table.

k=7



This benchmark uses automatic eval using an image recognition model

"ConceptMix: A Compositional Image Generation Benchmark with Controllable Difficulty"

Diffusion beyond text-to-image

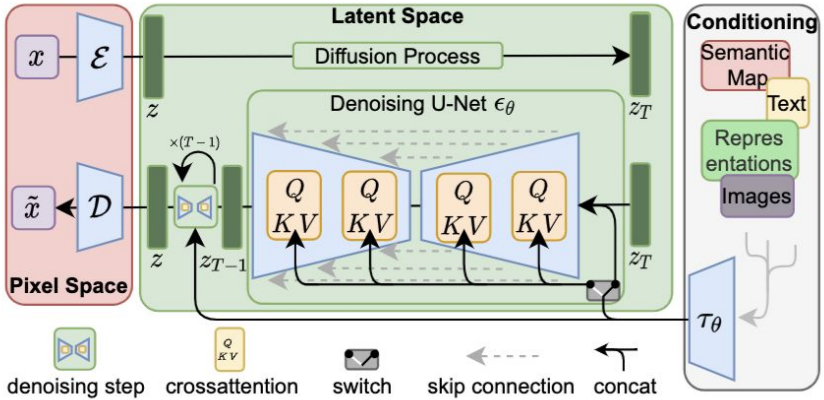
A few examples of generative model tasks in images

Text-to-image

"A kangaroo lying down"'

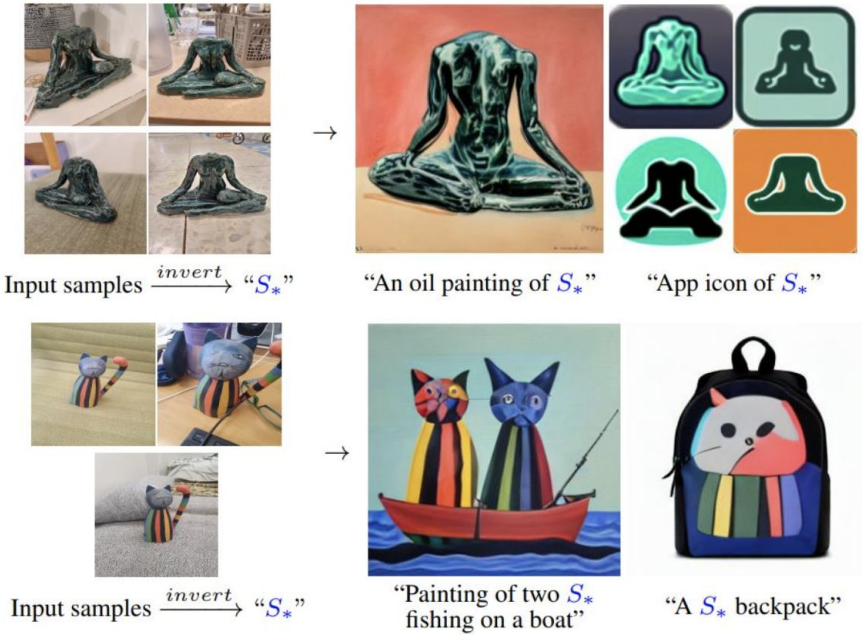
Super-resolution

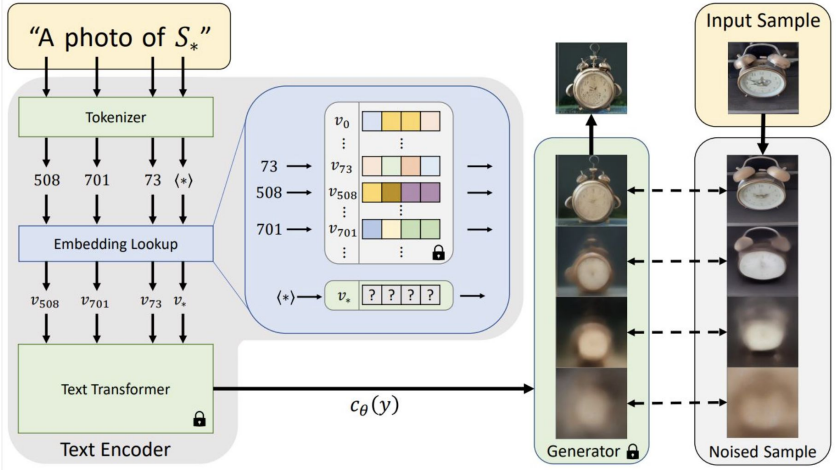
Image inpainting



Textual inversion: learning to prompt new concepts

Idea: learn an input “word” to capture a new concept from a few images, and generate it

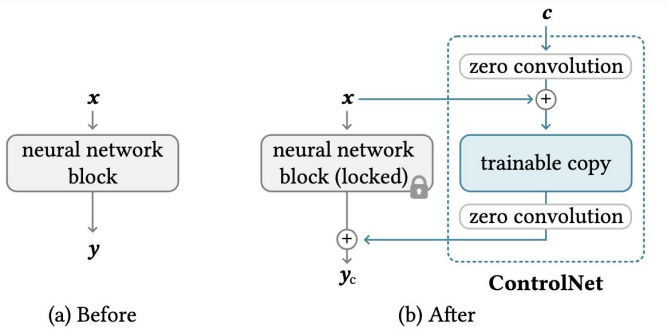




“An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion”

ControlNet: adding new conditions to text-to-image

If we start with a t2i model like StableDiffusion, then it *should* be easier to learn to map other conditioning information



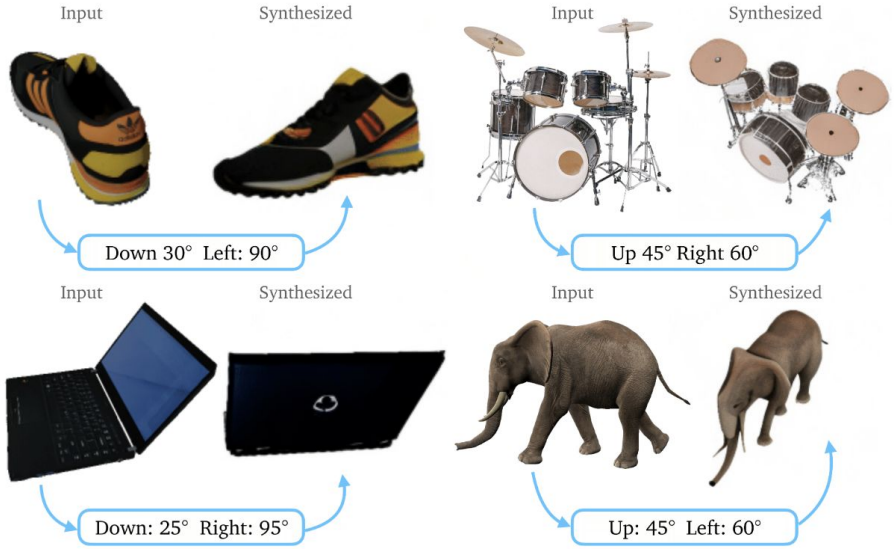
"Adding Conditional Control to Text-to-Image Diffusion Models"

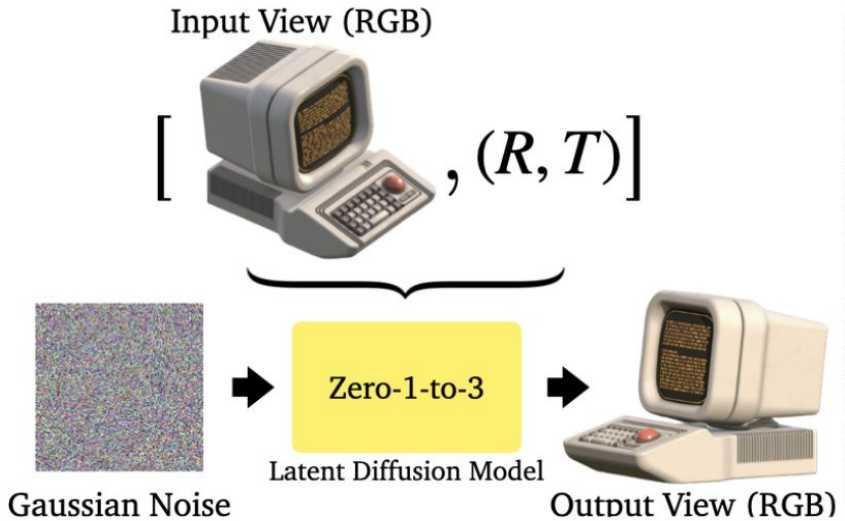
General theme: better control of image generation

"MoA : Mixture-of-Attention for Subject-Context Disentanglement in Personalized Image Generation"

Zero-1-to-3: rotating objects in 3D

Key idea: rotate objects by conditioning diffusion on starting image = angles



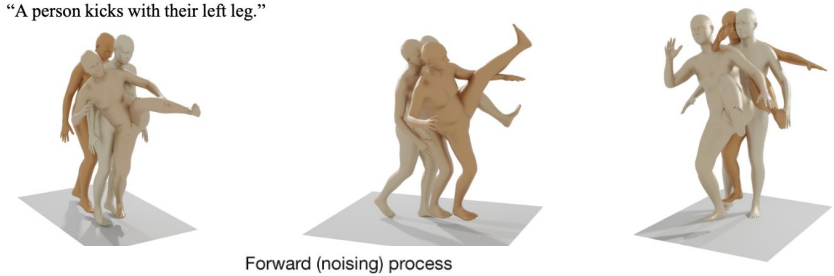


"Zero-1-to-3: Zero-shot One Image to 3D Object"

Diffusion on other data types

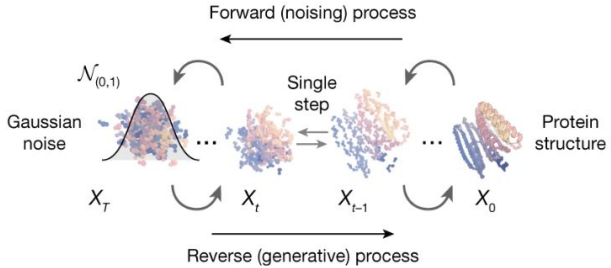
Human motion

"MDM: Human Motion Diffusion Model"



Proteins

"De novo design of protein structure and function with RFdiffusion"



Language

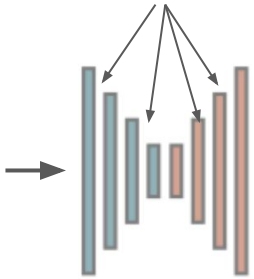
"Language Modeling by Estimating the Ratios of the Data Distribution"

study ants bear burrito skyline song

Diffusion representations for dense perception tasks

Idea: Unet activations capture meaningful content at the pixel level

Features from Unet

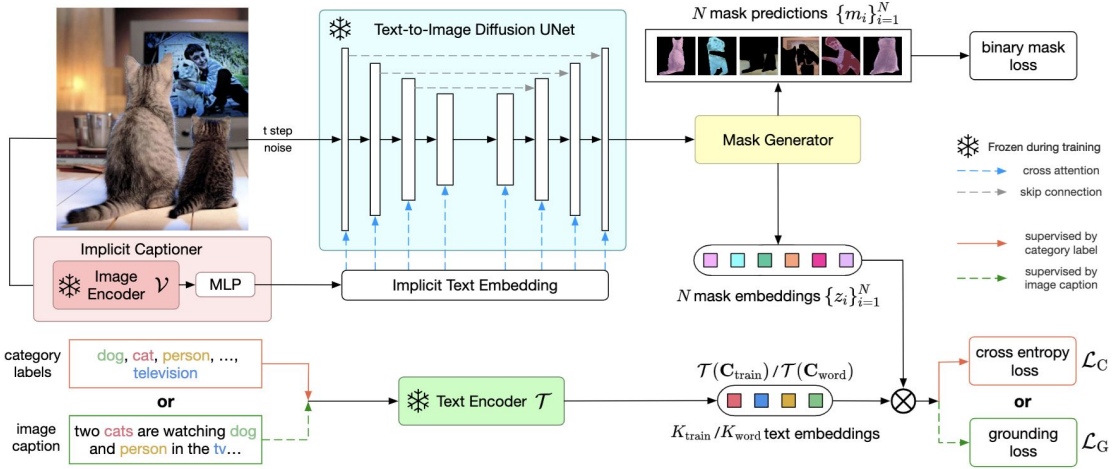


K-Means Clustering of
Frozen Diffusion Features

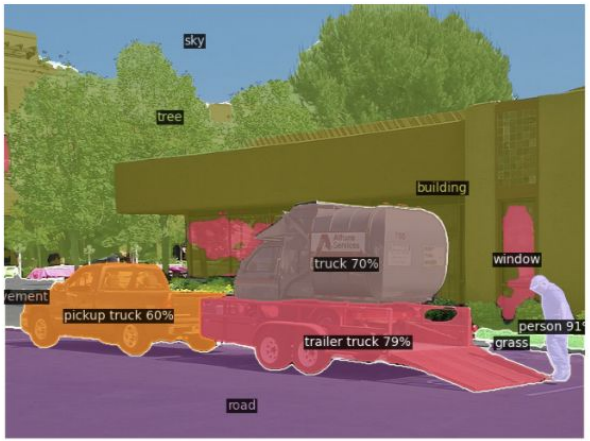
"Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models"

Diffusion representations for dense perception tasks

Idea: Unet activations capture meaningful content at the pixel level
So: use it as a backbone in segmentation



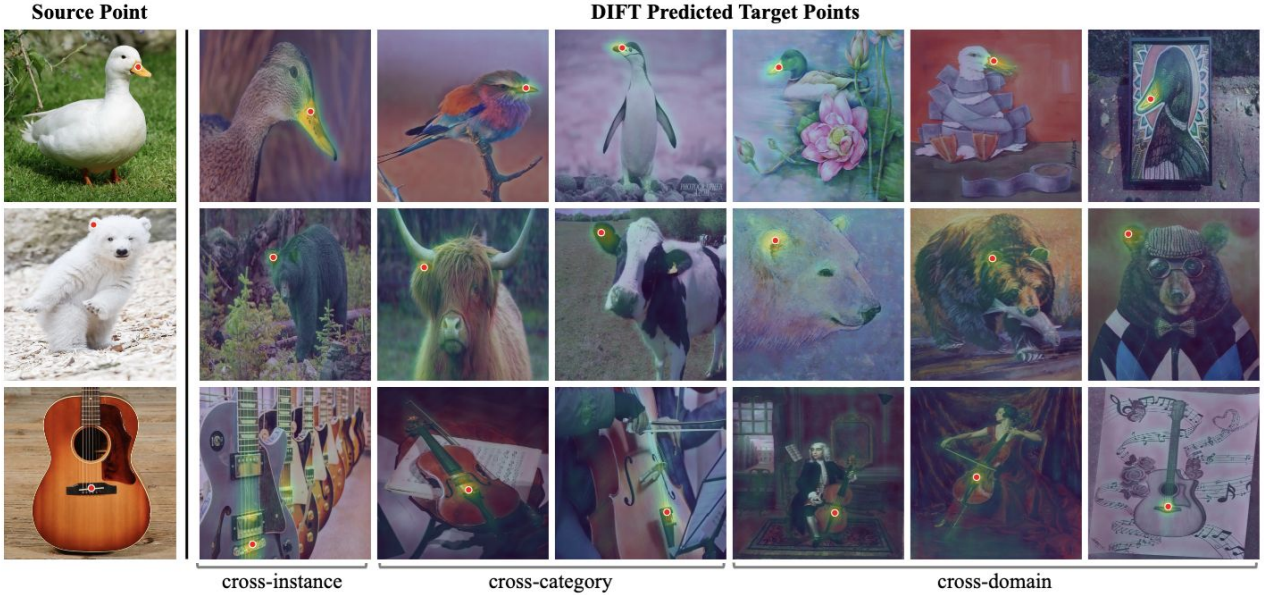
Open-Vocabulary Panoptic Segmentation
Prediction from ODISE



"Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models"

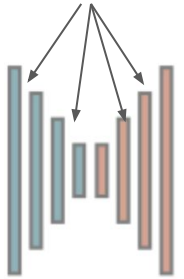
Diffusion representations for dense perception tasks

Idea: use Unet activations and find point-point correspondence by feature similarity



"Emergent Correspondence from Image Diffusion"

Copy features from Unet



Use vector dot product to find most similar point in 2nd image