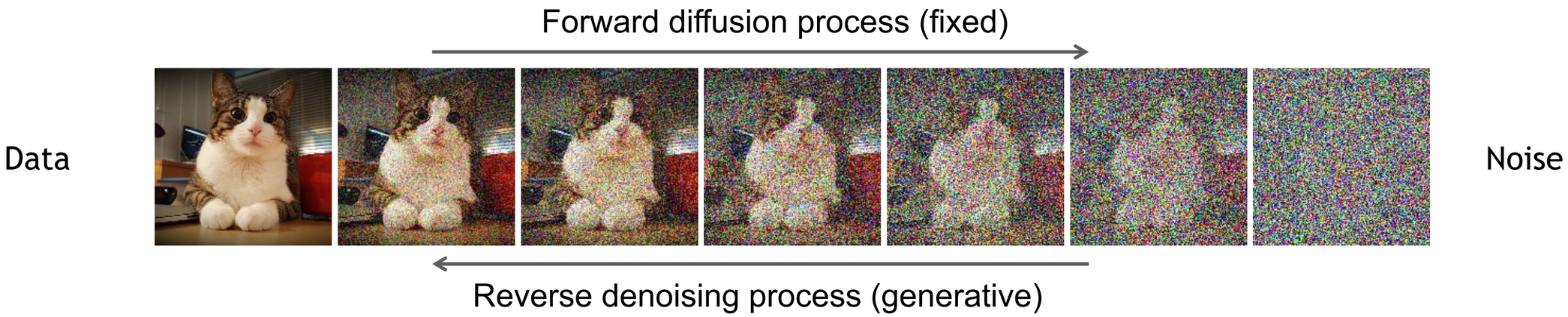
Denoising Diffusion Models

Learning to generate by denoising

Denoising diffusion models consist of two processes:

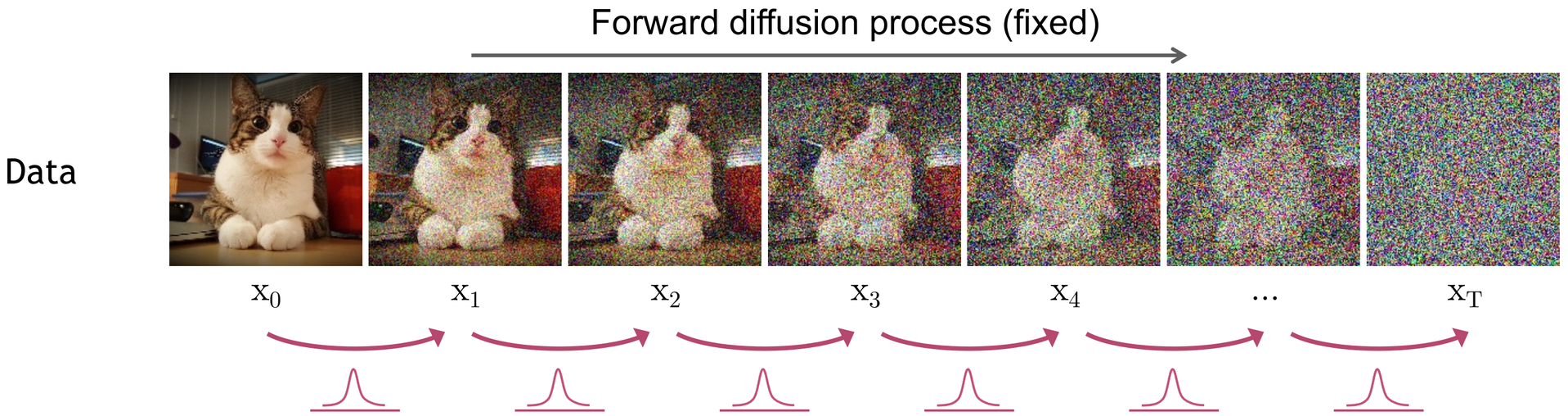
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Forward Diffusion Process

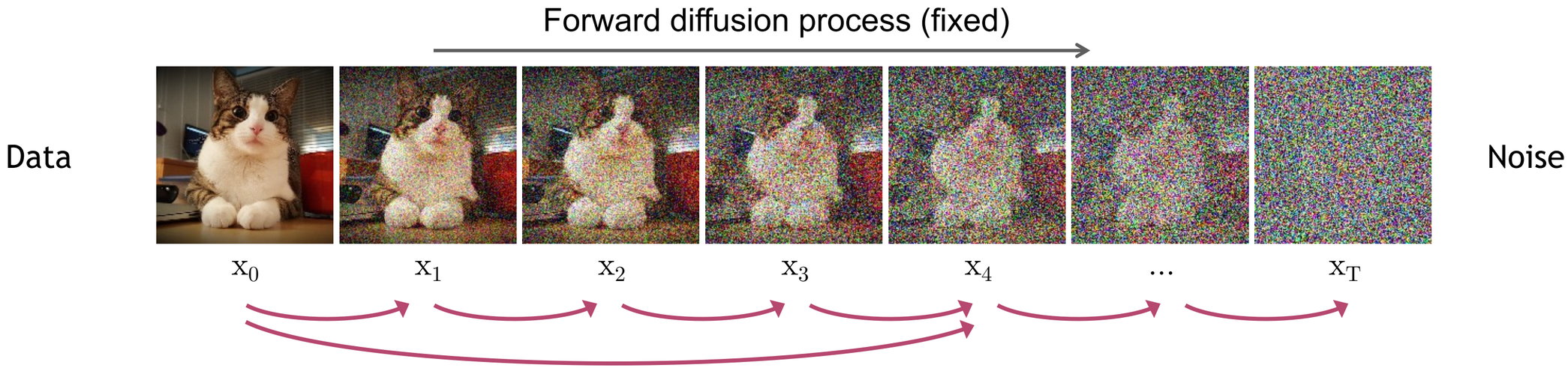
The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(joint)

Diffusion Kernel



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

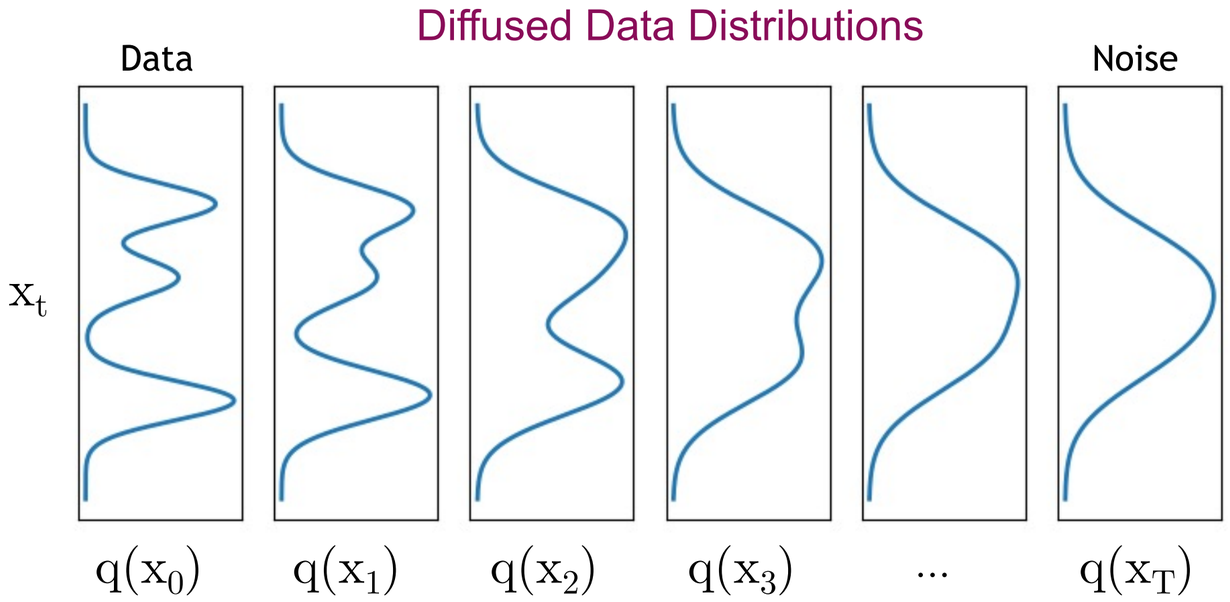
β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

What happens to a distribution in the forward diffusion?

So far, we discussed the diffusion kernel $q(\mathbf{x}_t|\mathbf{x}_0)$ but what about $q(\mathbf{x}_t)$?

$$\underbrace{q(\mathbf{x}_t)}_{\text{Diffused data dist.}} = \int \underbrace{q(\mathbf{x}_0, \mathbf{x}_t)}_{\text{Joint dist.}} d\mathbf{x}_0 = \int \underbrace{q(\mathbf{x}_0)}_{\text{Input data dist.}} \underbrace{q(\mathbf{x}_t|\mathbf{x}_0)}_{\text{Diffusion kernel}} d\mathbf{x}_0$$

The diffusion kernel is Gaussian convolution.



We can sample $\mathbf{x}_t \sim q(\mathbf{x}_t)$ by first sampling $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and then sampling $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ (i.e., ancestral sampling).

Generative Learning by Denoising

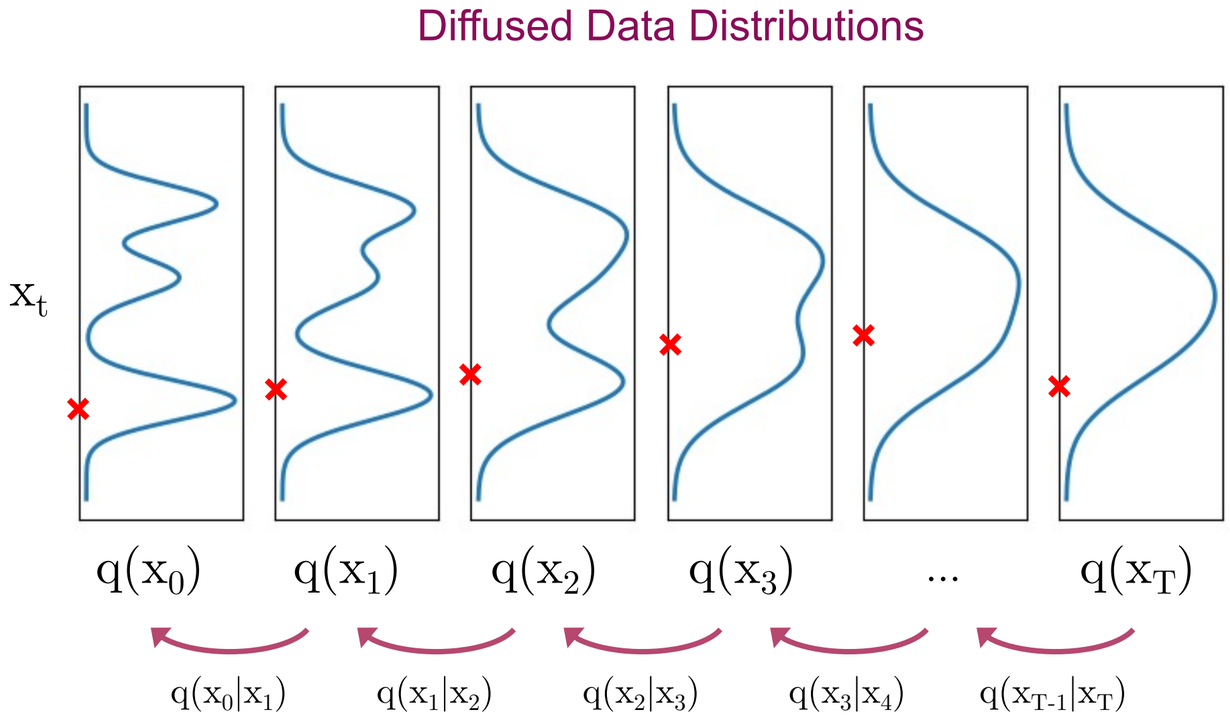
Recall, that the diffusion parameters are designed such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

Generation:

Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

Iteratively sample $\mathbf{x}_{t-1} \sim \underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}$

True Denoising Dist.

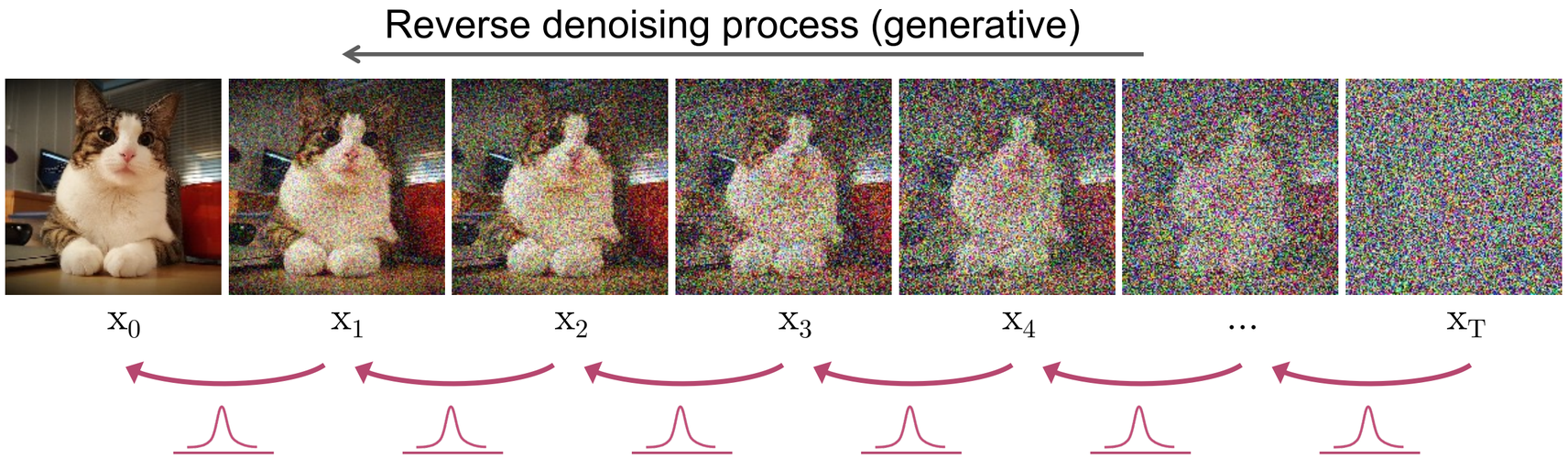


In general, $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \propto q(\mathbf{x}_{t-1})q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is intractable.

Can we approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$? Yes, we can use a Normal distribution if β_t is small in each forward diffusion step.

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}, \sigma_t^2\mathbf{I})$$

Trainable network
(U-net, Denoising Autoencoder)

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Learning Denoising Model

Variational upper bound

For training, we can form variational upper bound that is commonly used for training variational autoencoders:

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] =: L$$

Sohl-Dickstein et al. ICML 2015 and Ho et al. NeurIPS 2020 show that:

$$L = \mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1))}_{L_0}\right]$$

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ is the tractable posterior distribution:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1};\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t\mathbf{I}),$$

$$\text{where } \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \text{ and } \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Parameterizing the Denoising Model

Since both $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are Normal distributions, the KL divergence has a simple form:

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} ||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2 \right] + C$$

Recall that $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$. Ho et al. NeurIPS 2020 observe that:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$$

They propose to represent the mean of the denoising model using a *noise-prediction* network:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\ \epsilon_\theta(\mathbf{x}_t, t) \right)$$

With this parameterization

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\frac{\beta_t^2}{2\sigma_t^2 (1-\beta_t)(1-\bar{\alpha}_t)} ||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\ \epsilon}_{\mathbf{x}_t}, t)||^2 \right] + C$$

Training Objective Weighting

Trading likelihood for perceptual quality

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \Big[\underbrace{\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}}_{\lambda_t} ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\ \epsilon, t)||^2 \Big]$$

The time dependent λ_t ensures that the training objective is weighted properly for the maximum data likelihood training.

However, this weight is often very large for small t's.

Ho et al. NeurIPS 2020 observe that simply setting $\lambda_t = 1$ improves sample quality. So, they propose to use:

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \Big[||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\ \epsilon}_{\mathbf{x}_t}, t)||^2 \Big]$$

For more advanced weighting see Choi et al., Perception Prioritized Training of Diffusion Models, CVPR 2022.

Summary

Training and Sample Generation

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
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

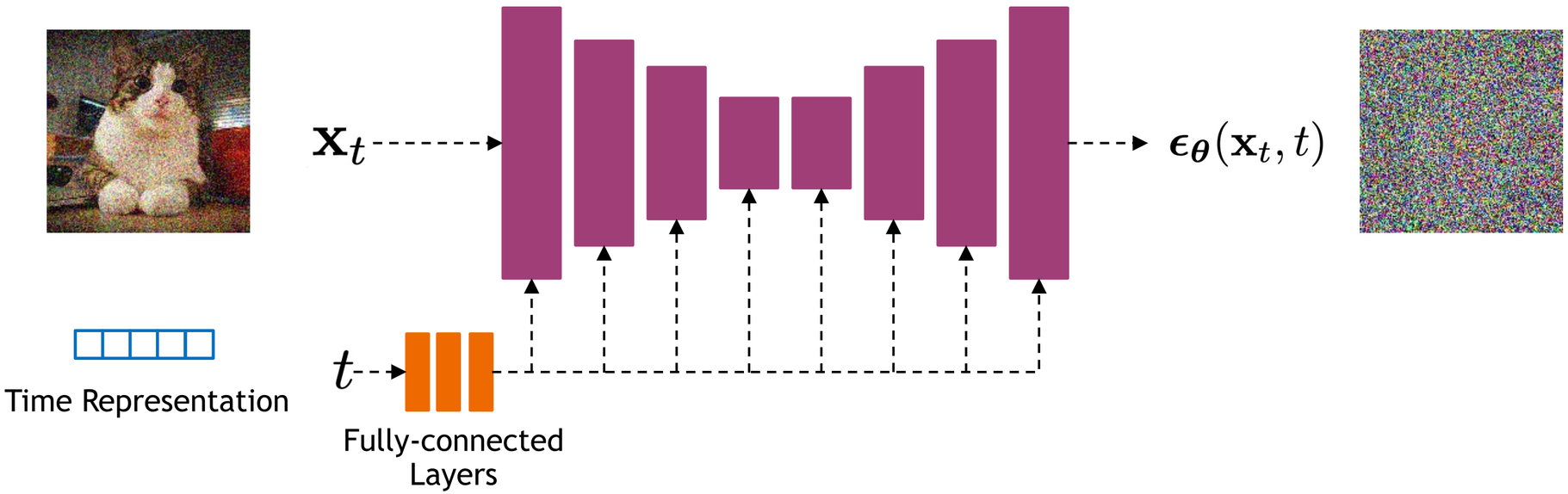
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
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** $\mathbf{x}_0$

Implementation Considerations

Network Architectures

Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent $\epsilon_\theta(\mathbf{x}_t, t)$



Time representation: sinusoidal positional embeddings or random Fourier features.

Time features are fed to the residual blocks using either simple spatial addition or using adaptive group normalization layers. (see Dhariwal and Nichol NeurIPS 2021)

Diffusion Parameters

Noise Schedule

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I})$$

Data

Noise

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

Above, β_t and σ_t^2 control the variance of the forward diffusion and reverse denoising processes respectively.

Often a linear schedule is used for β_t, and σ_t^2 is set equal to β_t.

Kingma et al. NeurIPS 2022 introduce a new parameterization of diffusion models using signal-to-noise ratio (SNR), and show how to learn the noise schedule by minimizing the variance of the training objective.

We can also train σ_t^2 while training the diffusion model by minimizing the variational bound (Improved DPM by Nichol and Dhariwal ICML 2021) or after training the diffusion model (Analytic-DPM by Bao et al. ICLR 2022).

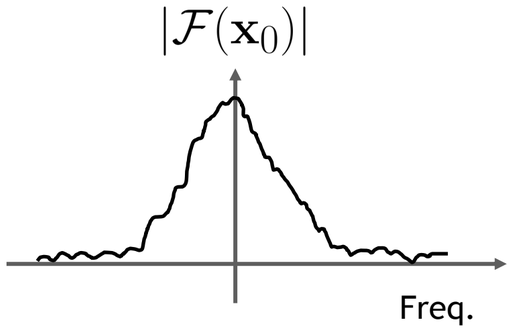
What happens to an image in the forward diffusion process?

Recall that sampling from $q(\mathbf{x}_t|\mathbf{x}_0)$ is done using $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

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

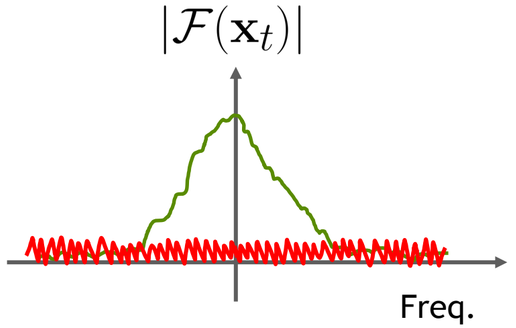
Fourier Transform

$$\mathcal{F}(\mathbf{x}_t) = \sqrt{\bar{\alpha}_t}\ \mathcal{F}(\mathbf{x}_0) + \sqrt{(1-\bar{\alpha}_t)}\ \mathcal{F}(\epsilon)$$



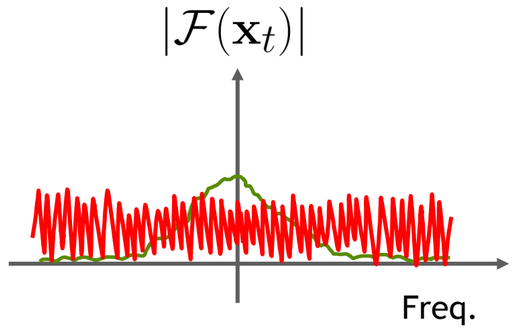
Small t

$\bar{\alpha}_t \sim 1$



Large t

$\bar{\alpha}_t \sim 0$



In the forward diffusion, the high frequency content is perturbed faster.

Content-Detail Tradeoff

Reverse denoising process (generative)

Data

Noise

x_0 x_1 x_2 x_3 x_4 ... x_T

The denoising model is specialized for generating the high-frequency content (i.e., low-level details)

The denoising model is specialized for generating the low-frequency content (i.e., coarse content)

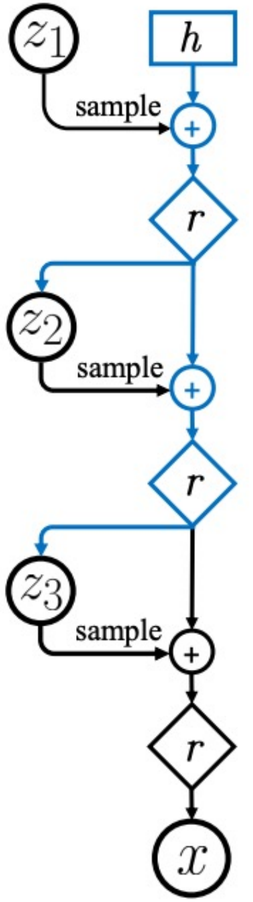
The weighting of the training objective for different timesteps is important!

Connection to VAEs

Diffusion models can be considered as a special form of hierarchical VAEs.

However, in diffusion models:

- The encoder is fixed
- The latent variables have the same dimension as the data
- The denoising model is shared across different timestep
- The model is trained with some reweighting of the variational bound.



Vahdat and Kautz, NVAE: A Deep Hierarchical Variational Autoencoder, NeurIPS 2020
Sønderby, et al.. Ladder variational autoencoders, NeurIPS 2016.

Summary

Denoising Diffusion Probabilistic Models

In this part, we reviewed denoising diffusion probabilistic models.

The model is trained by sampling from the forward diffusion process and training a denoising model to predict the noise.

We discussed how the forward process perturbs the data distribution or data samples.

The devil is in the details:

- Network architectures
- Objective weighting
- Diffusion parameters (i.e., noise schedule)

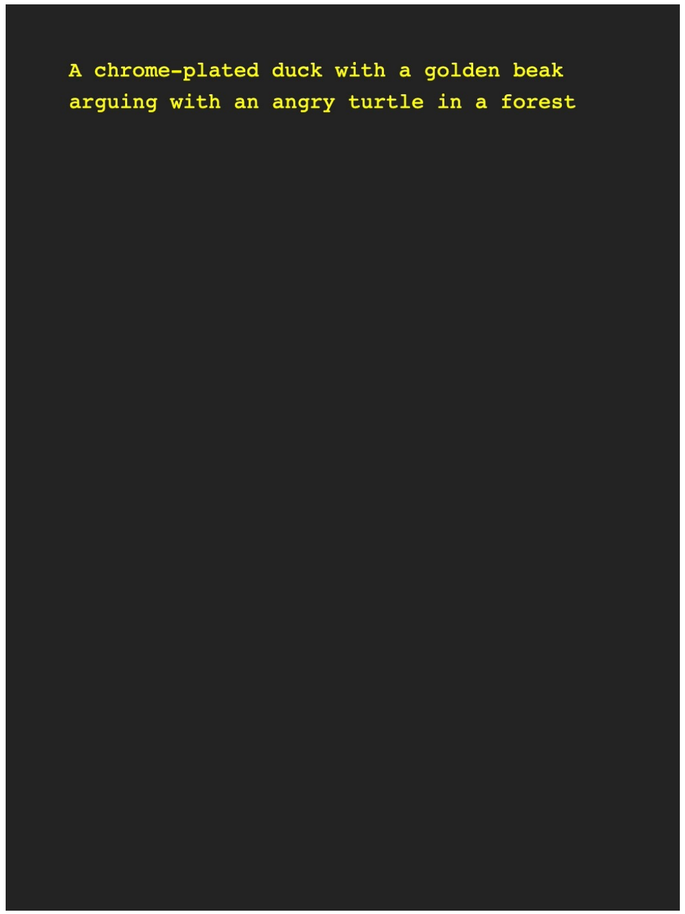
See “Elucidating the Design Space of Diffusion-Based Generative Models” by Karras et al. for important design decisions.

Applications (1):
Image Synthesis, Controllable Generation, Text-to-Image

Text-to-image generation

Inverse of image captioning

- Conditional generation: given a text prompt c, generate high-res images x.



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

GLIDE

OpenAI

- A 64x64 base model + a 64x64 → 256x256 super-resolution model.
- Tried classifier-free and CLIP guidance. Classifier-free guidance works better than CLIP guidance.

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"

"robots meditating in a vipassana retreat"

"a fall landscape with a small cottage next to a lake"

Samples generated with classifier-free guidance (256x256)

Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

CLIP guidance

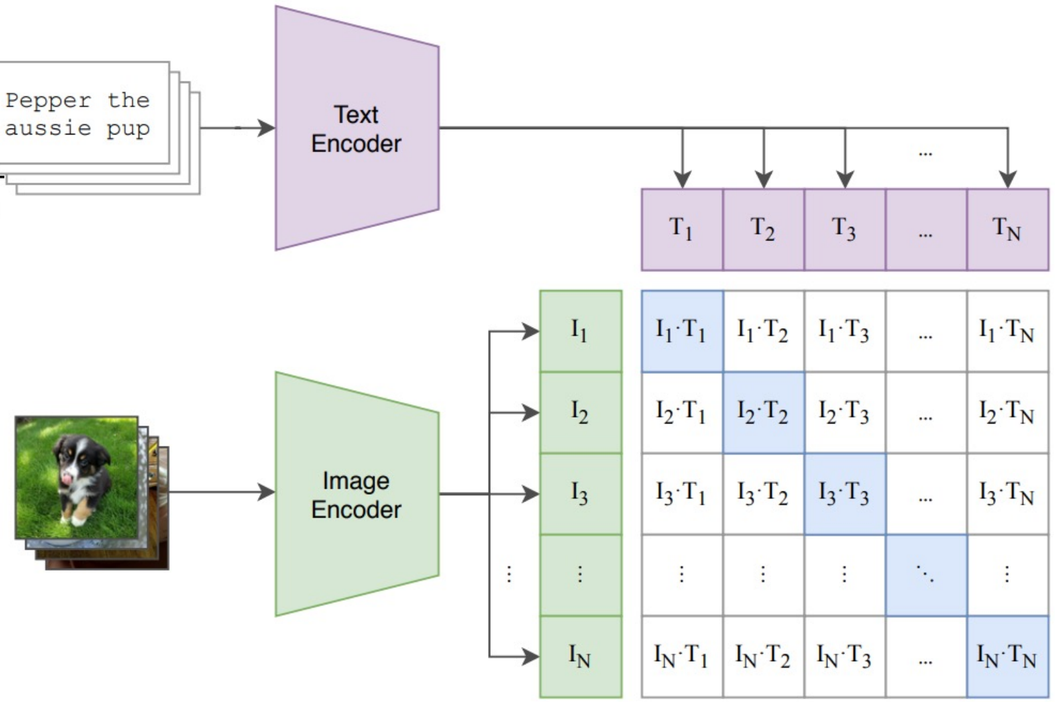
What is a CLIP model?

- Trained by contrastive cross-entropy loss:

$$-\log \frac{\exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_k)/\tau)} - \log \frac{\exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_k) \cdot g(\mathbf{c}_j)/\tau)}$$

- The optimal value of $f(\mathbf{x}) \cdot g(\mathbf{c})$ is

$$\log \frac{p(\mathbf{x}, \mathbf{c})}{p(\mathbf{x})p(\mathbf{c})} = \log p(\mathbf{c}|\mathbf{x}) - \log p(\mathbf{c})$$



Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", 2021.
Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

CLIP guidance

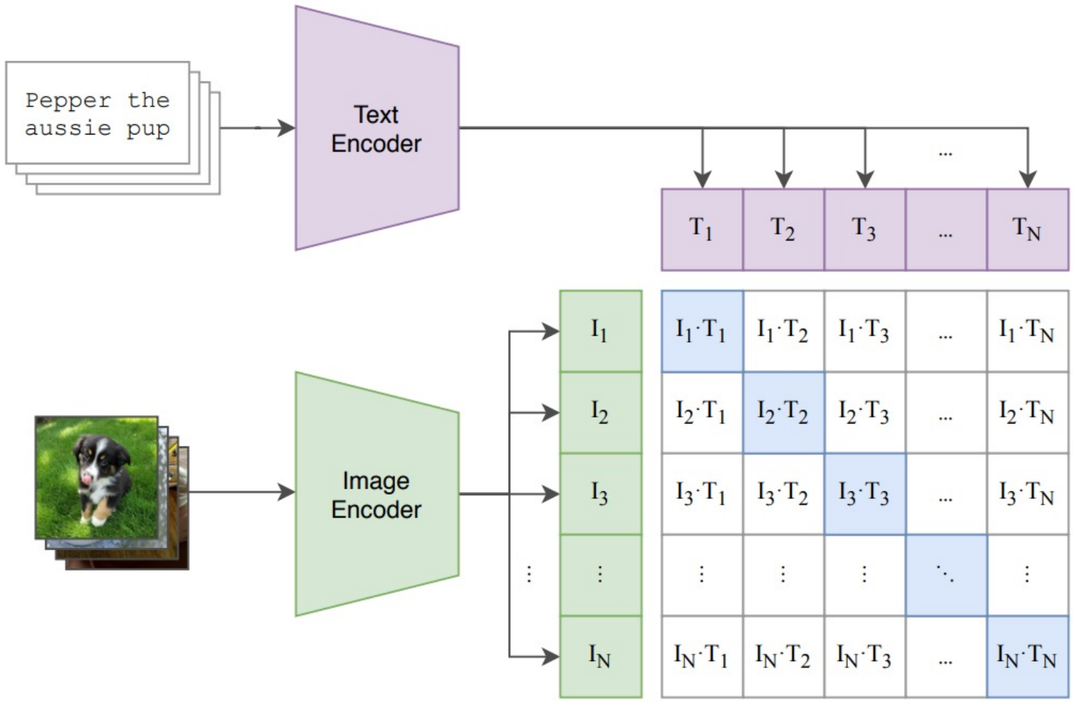
Replace the classifier in classifier guidance with a CLIP model

- Sample with a modified score:

$$\nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega \log p(\mathbf{c}|\mathbf{x}_t)]$$

$$= \nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega(\underbrace{\log p(\mathbf{c}|\mathbf{x}_t) - \log p(\mathbf{c})}_{\text{CLIP model}})]$$

$$= \nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega(f(\mathbf{x}_t) \cdot g(\mathbf{c}))]$$



Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", 2021.
Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

GLIDE

OpenAI

- Fine-tune the model especially for inpainting: feed randomly occluded images with an additional mask channel as the input.

"an old car in a snowy forest"

"a man wearing a white hat"

Text-conditional image inpainting examples

Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

DALL·E 2

OpenAI

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

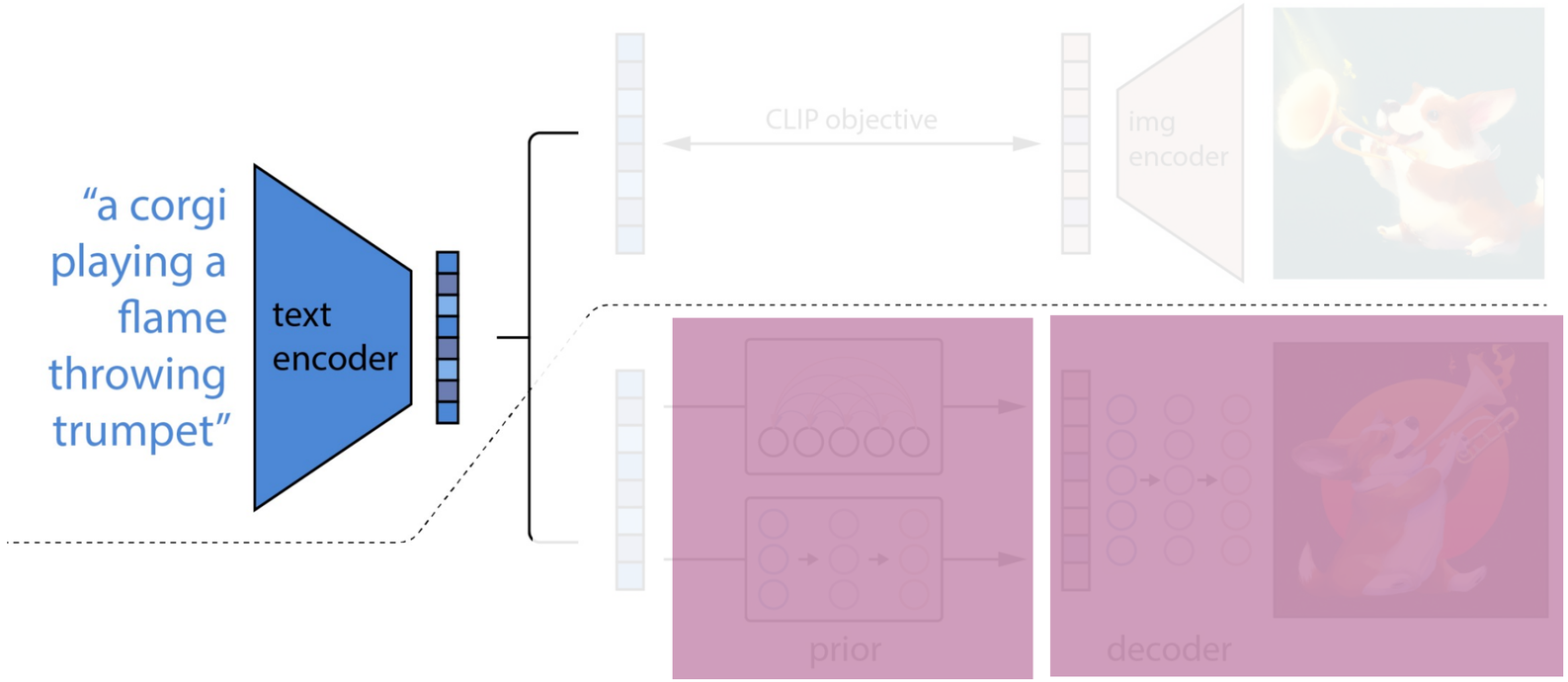
1kx1k Text-to-image generation.

Outperform DALL-E (autoregressive transformer).

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.

DALL·E 2

Model components



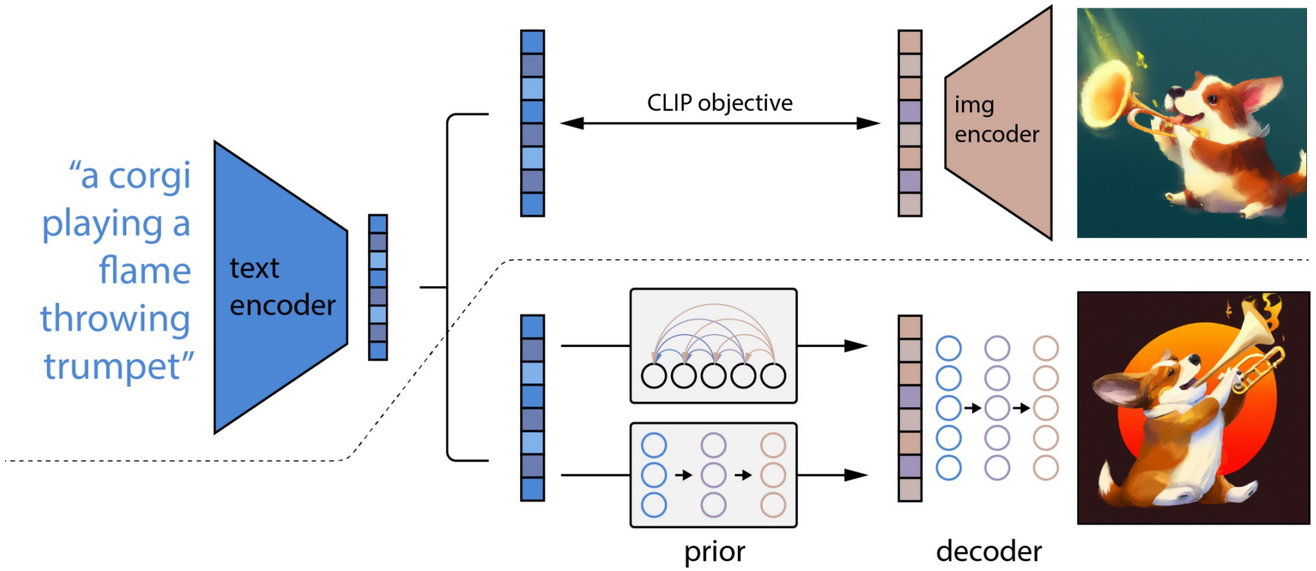
Prior: produces CLIP image embeddings conditioned on the caption.

Decoder: produces images conditioned on CLIP image embeddings and text.

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

DALL·E 2

Model components



Why conditional on CLIP image embeddings?

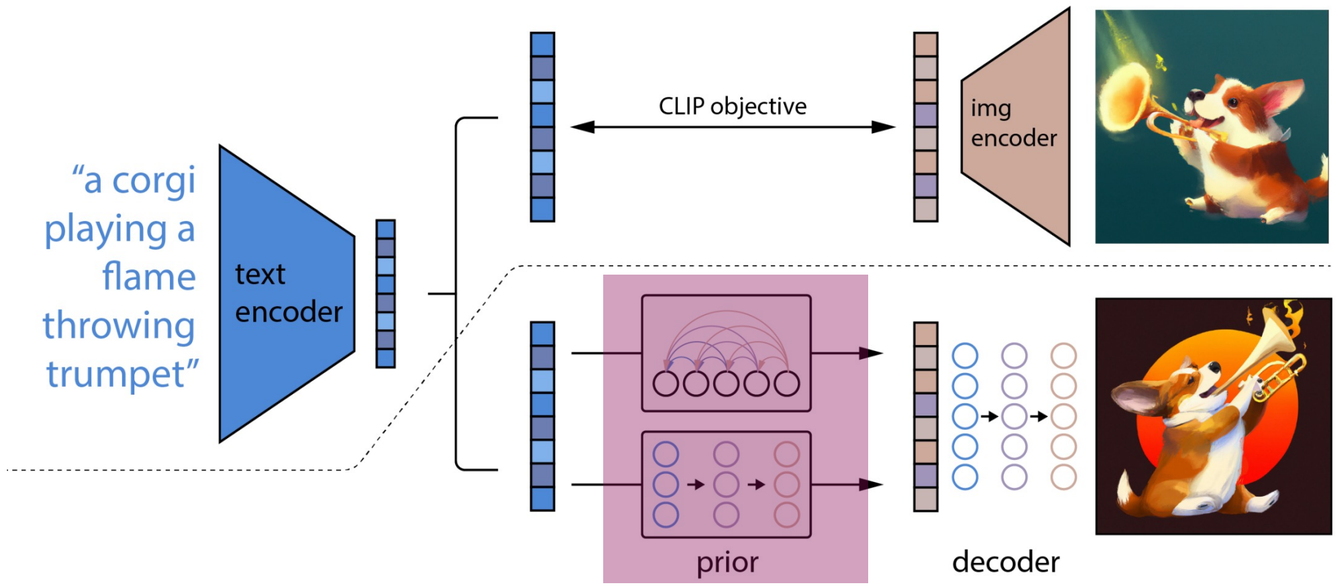
CLIP image embeddings capture high-level semantic meaning; latents in the decoder model take care of the rest.

The bipartite latent representation` enables several text-guided image manipulation tasks.

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

DALL·E 2

Model components (1/2): prior model



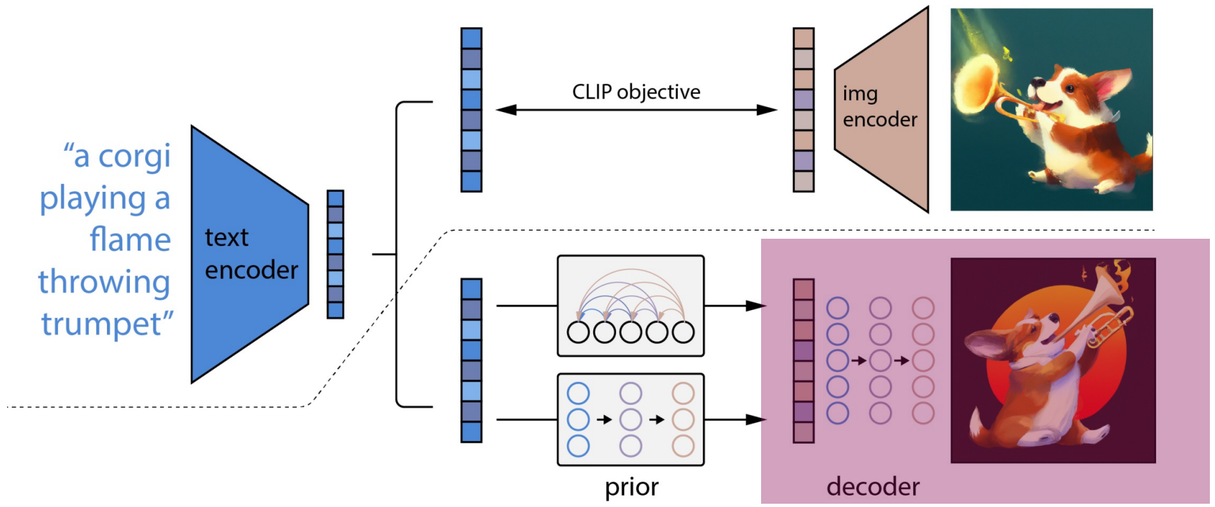
Prior: produces CLIP image embeddings conditioned on the caption.

- Option 1. autoregressive prior: quantize image embedding to a seq. of discrete codes and predict them autoregressively.
- Option 2. diffusion prior: model the continuous image embedding by diffusion models conditioned on caption.

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

DALL·E 2

Model components (2/2): decoder model



Decoder: produces images conditioned on CLIP image embeddings (and text).

- Cascaded diffusion models: 1 base model (64x64), 2 super-resolution models (64x64 → 256x256, 256x256 → 1024x1024).
- Largest super-resolution model is trained on patches and takes full-res inputs at inference time.
- Classifier-free guidance & noise conditioning augmentation are important.

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

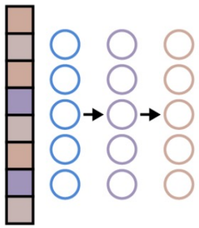
DALL·E 2

Bipartite latent representations

Bipartite latent representations $(\mathbf{z}, \mathbf{x}_T)$

$\mathbf{z}$: CLIP image embeddings

$\mathbf{x}_T$: inversion of DDIM sampler
(latents in the decoder model)



decoder

Near exact reconstruction

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.

DALL·E 2

Image variations

Fix the CLIP embedding $\mathbf{z}$.

Decode using different decoder latents $\mathbf{x}_T$

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.

DALL·E 2

Image interpolation

Interpolate image CLIP embeddings $\mathbf{z}_i$.

Use different $\mathbf{x}_T$ to get different interpolation trajectories.

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.

DALL·E 2

Text Diffs

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

Change the image CLIP embedding towards the difference of the text CLIP embeddings of two prompts.

Decoder latent is kept as a constant.

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.

Imagen

Google Research, Brain team

Input: text; Output: 1kx1k images

- An unprecedented degree of photorealism
  - SOTA automatic scores & human ratings
- A deep level of language understanding
- Extremely simple
  - no latent space, no quantization

A brain riding a rocketship heading towards the moon.

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

Imagen

Google Research, Brain team

A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

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

Imagen

Google Research, Brain team

A dragon fruit wearing karate belt in the snow.

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

Imagen

Google Research, Brain team

A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.

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

Imagen

Google Research, Brain team

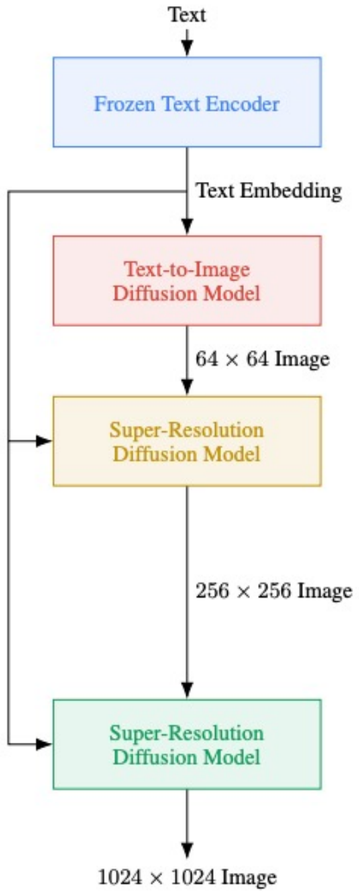
A cute hand-knitted koala wearing a sweater with 'CVPR' written on it.

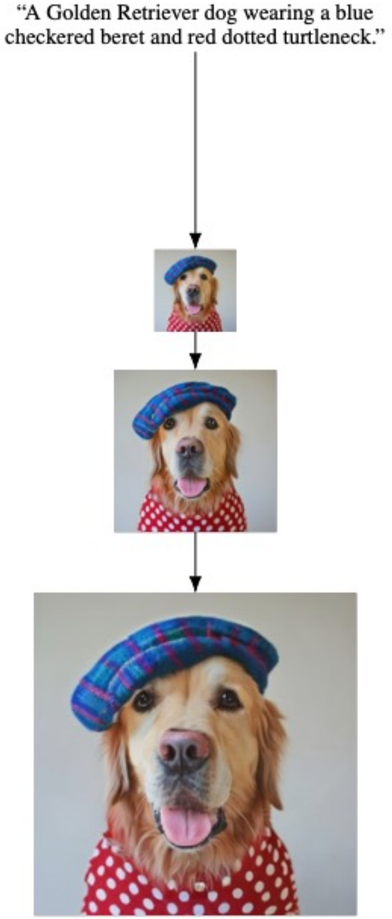
Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, arXiv 2022.

Imagen

Key modeling components:

- Cascaded diffusion models
- Classifier-free guidance and dynamic thresholding.
- Frozen large pretrained language models as text encoders. (T5-XXL)



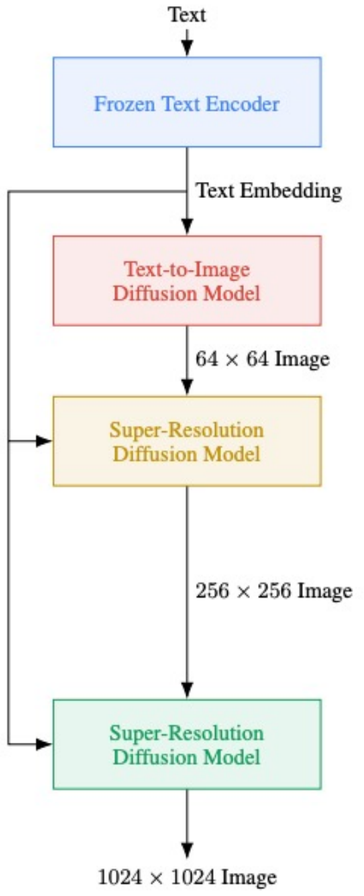


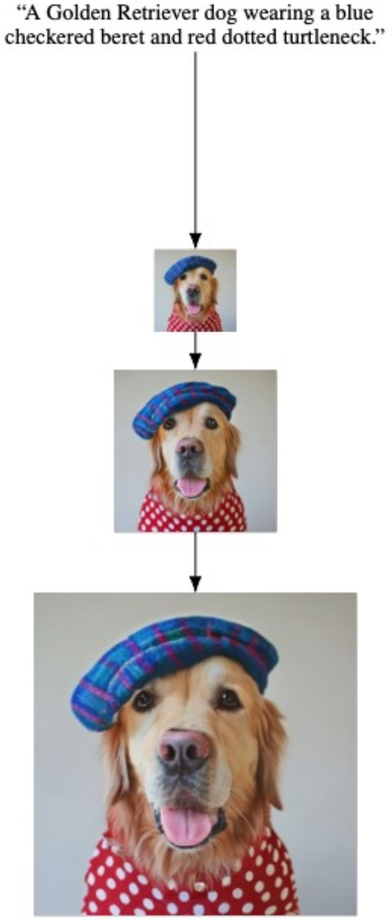
Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

Imagen

Key observations:

- Beneficial to use text conditioning for all super-res models.
    - Noise conditioning augmentation weakens information from low-res models, thus needs text conditioning as extra information input.
- Scaling text encoder is extremely efficient.
    - More important than scaling diffusion model size.
- Human raters prefer T5-XXL as the text encoder over CLIP encoder on DrawBench.



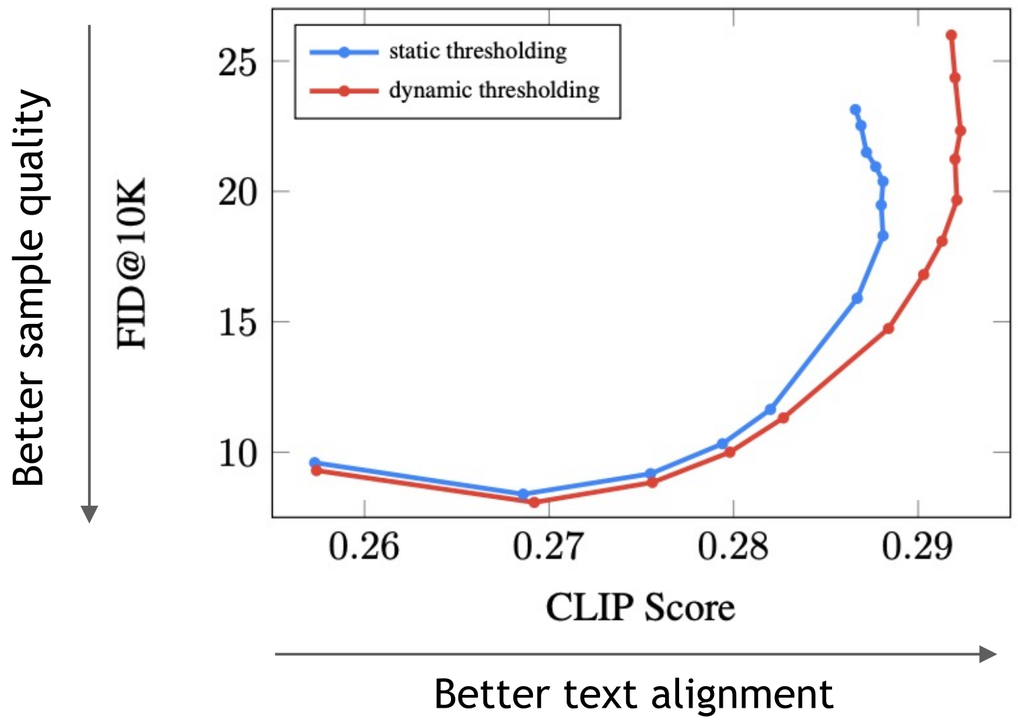


Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

Imagen

Dynamic thresholding

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



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

Imagen

Dynamic thresholding

- Large classifier-free guidance weights → better text alignment, worse image quality
- Hypothesis : at large guidance weight, the generated images are saturated due to the very large gradient updates during sampling
- Solution – dynamic thresholding: adjusts the pixel values of samples at each sampling step to be within a dynamic range computed over the statistics of the current samples.

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

Imagen

Dynamic thresholding

- Large class
- Hypothesis … dates during sampling
- Solution – … dynamic range com

Static thresholding

Dynamic thresholding

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

Imagen

DrawBench: new benchmark for text-to-image evaluations

- A set of 200 prompts to evaluate text-to-image models across multiple dimensions.
    - E.g., the ability to faithfully render different colors, numbers of objects, spatial relations, text in the scene, unusual interactions between objects.
    - Contains complex prompts, e.g, long and intricate descriptions, rare words, misspelled prompts.

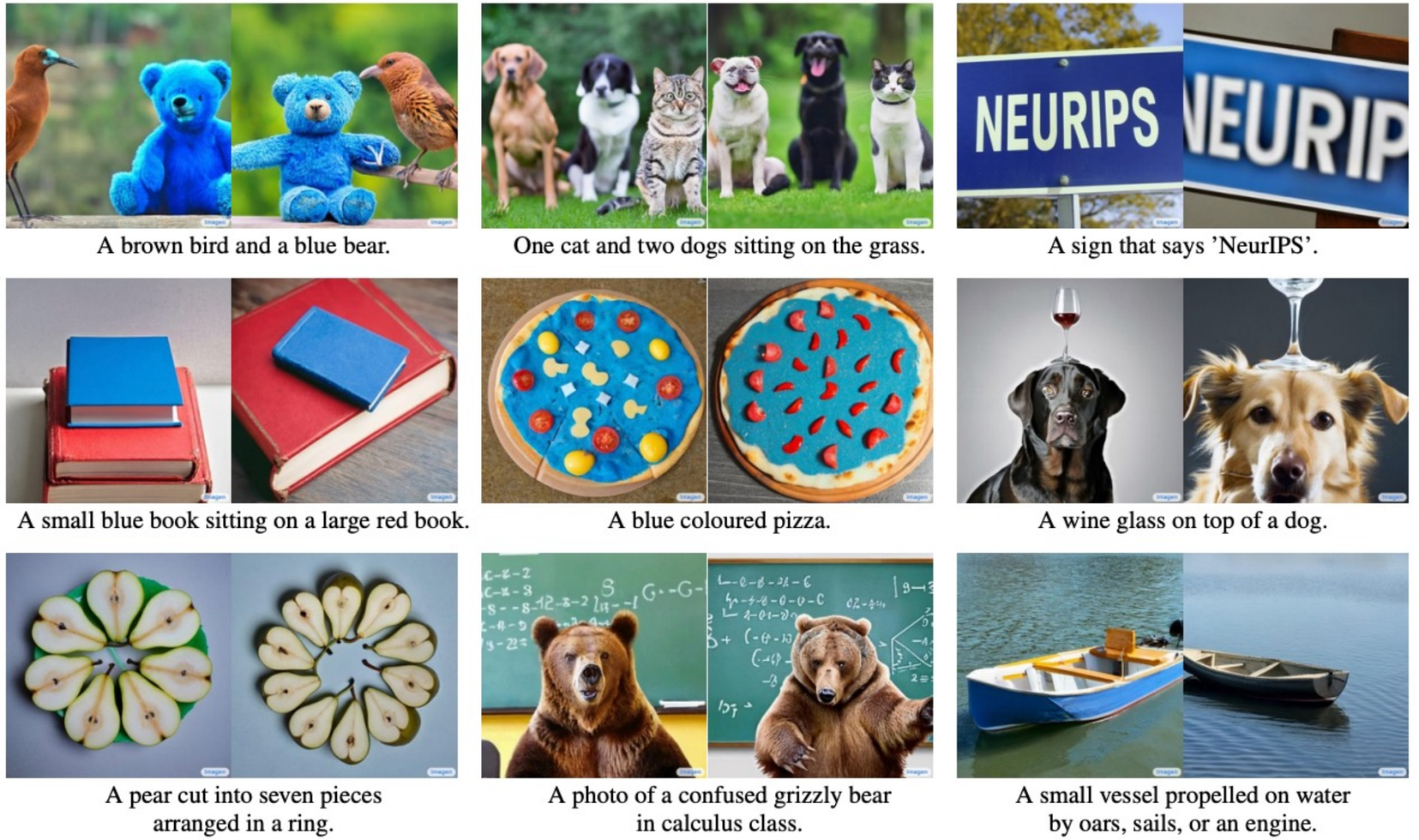
Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, arXiv 2022.

Imagen

DrawBench: new benchmark for text-to-image evaluations

- A set of 200 pro
  - E.g., the
    interactio
  - Contains

cene, unusual



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

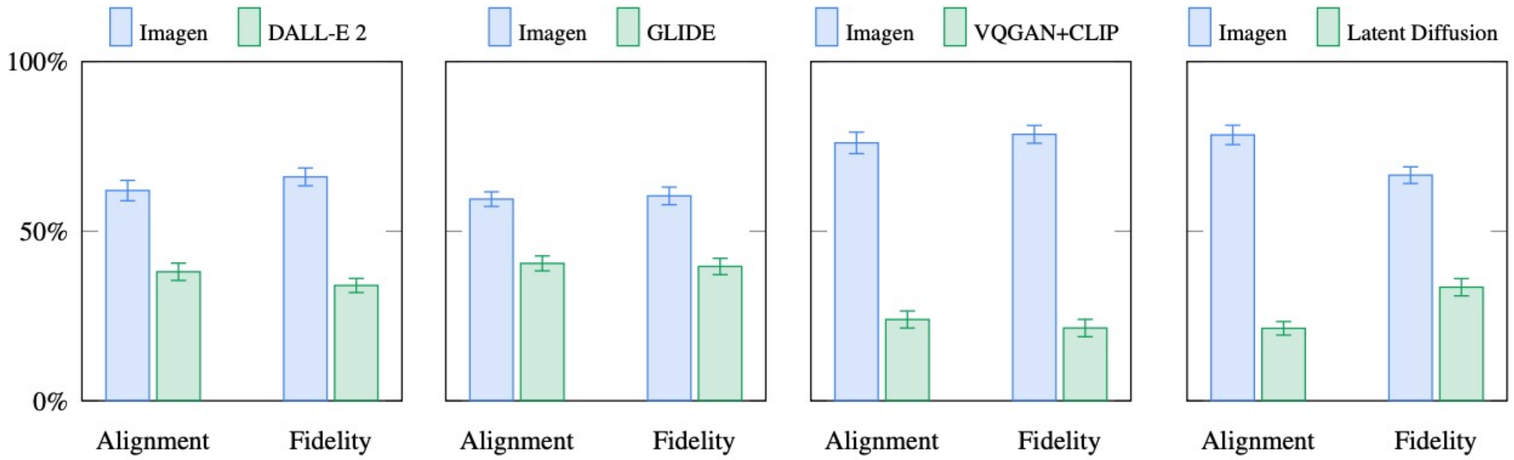
Imagen

Evaluations

Imagen got SOTA automatic evaluation scores on COCO dataset

Model	FID-30K	Zero-shot FID-30K
AttnGAN [76]	35.49	
DM-GAN [83]	32.64	
DF-GAN [69]	21.42	
DM-GAN + CL [78]	20.79	
XMC-GAN [81]	9.33	
LAFITE [82]	8.12	
Make-A-Scene [22]	7.55	
DALL-E [53]		17.89
LAFITE [82]		26.94
GLIDE [41]		12.24
DALL-E 2 [54]		10.39
Imagen (Our Work)		**7.27**

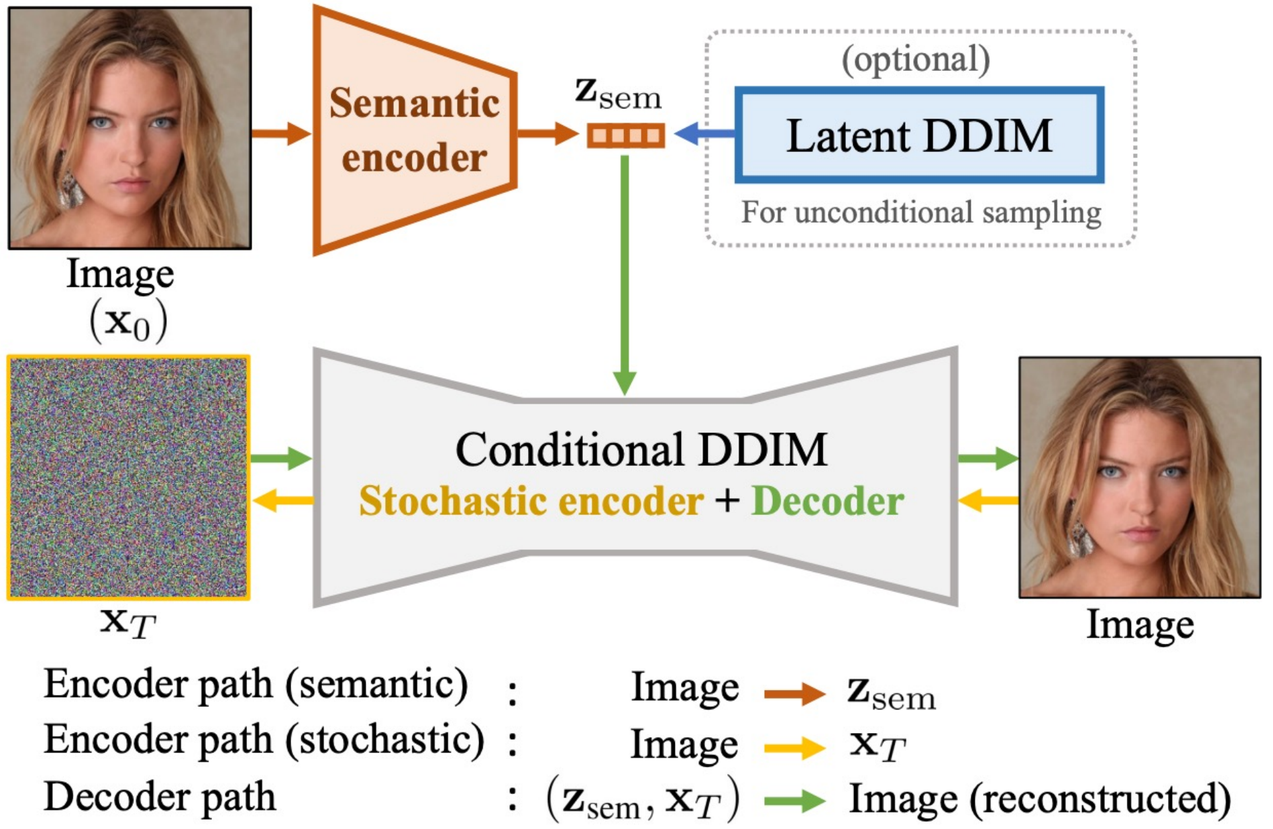
Imagen is preferred over recent work by human raters in sample quality & image-text alignment on DrawBench.



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

Diffusion Autoencoders

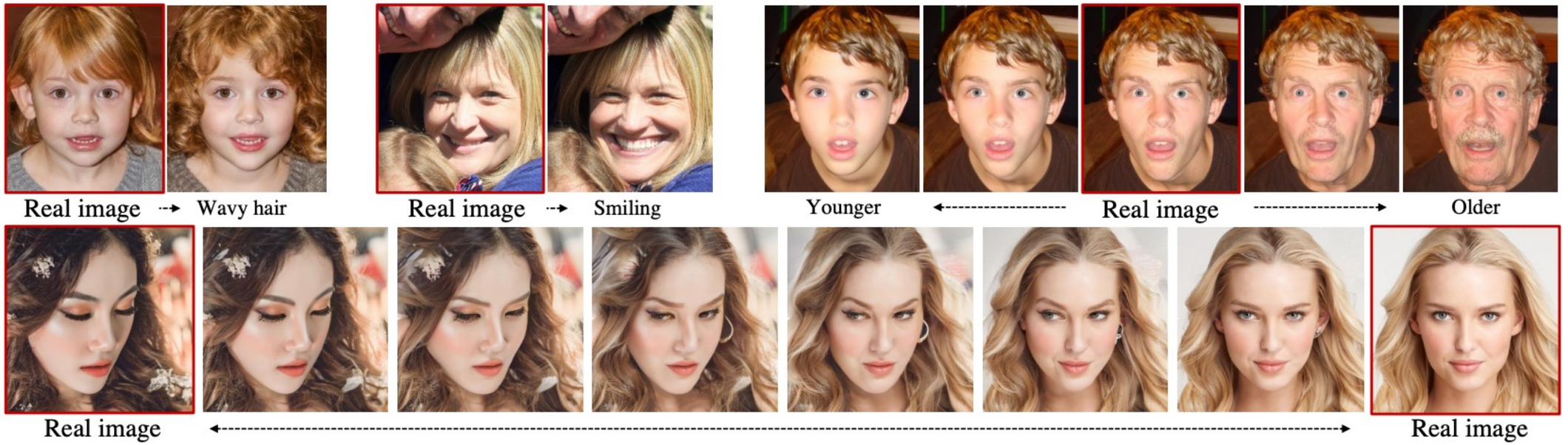
Learning semantic meaningful latent representations in diffusion models



Preechakul et al., "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation", CVPR 2022.

Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models

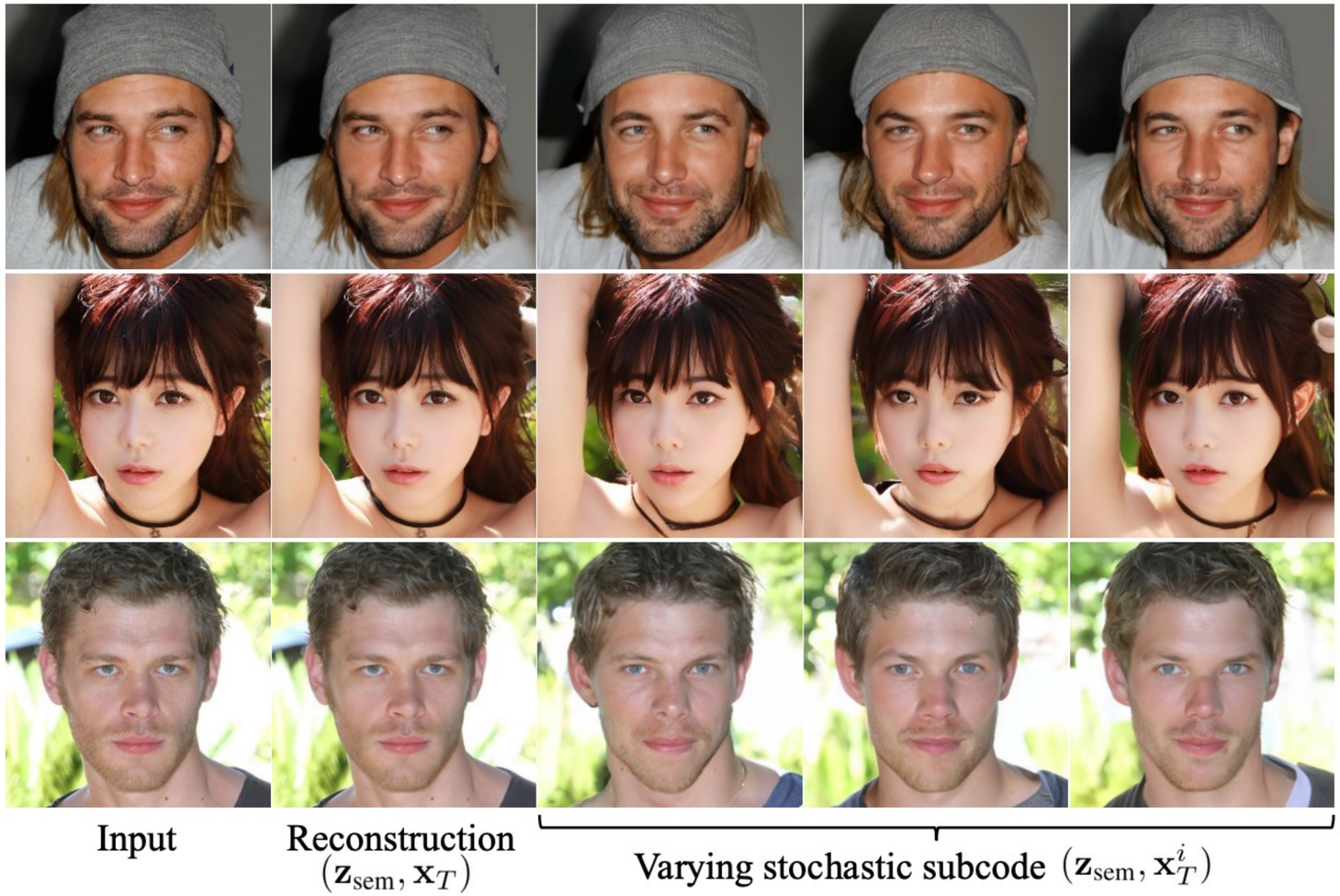


Changing the semantic latent $\mathbf{z}_{\text{sem}}$

Preechakul et al., “Diffusion Autoencoders: Toward a Meaningful and Decodable Representation”, CVPR 2022.

Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models



Preechakul et al., “Diffusion Autoencoders: Toward a Meaningful and Decodable Representation”, CVPR 2022.

Applications (2):
Image Editing, Image-to-Image, Super-resolution, Segmentation

Super-Resolution

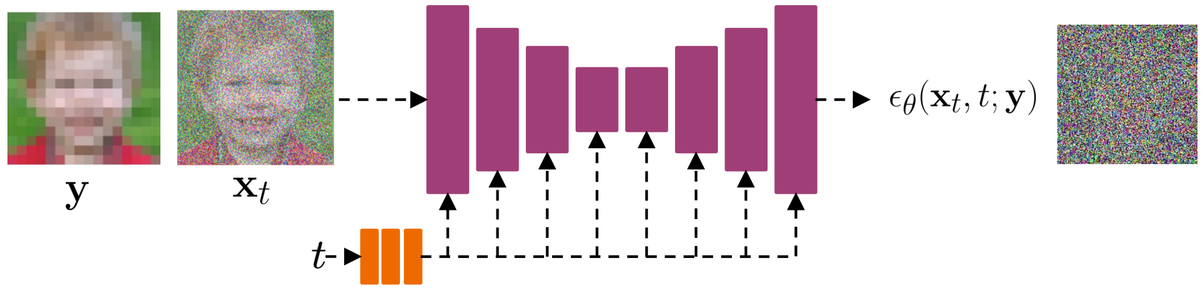
Super-Resolution via Repeated Refinement (SR3)

Image super-resolution can be considered as training $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ is a low-resolution image and $\mathbf{x}$ is the corresponding high-resolution image

Train a score model for $\mathbf{x}$ conditioned on $\mathbf{y}$ using:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\, \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\, \mathbb{E}_t\, ||\epsilon_\theta(\mathbf{x}_t, t; \mathbf{y}) - \epsilon||_p^p$$

The conditional score is simply a U-Net with $\mathbf{x_t}$ and $\mathbf{y}$ (resolution image) concatenated.



Saharia et al., Image Super-Resolution via Iterative Refinement, 2021

Super-Resolution

Super-Resolution via Repeated Refinement (SR3)

Saharia et al., Image Super-Resolution via Iterative Refinement, 2021

Image-to-Image Translation

Palette: Image-to-Image Diffusion Models

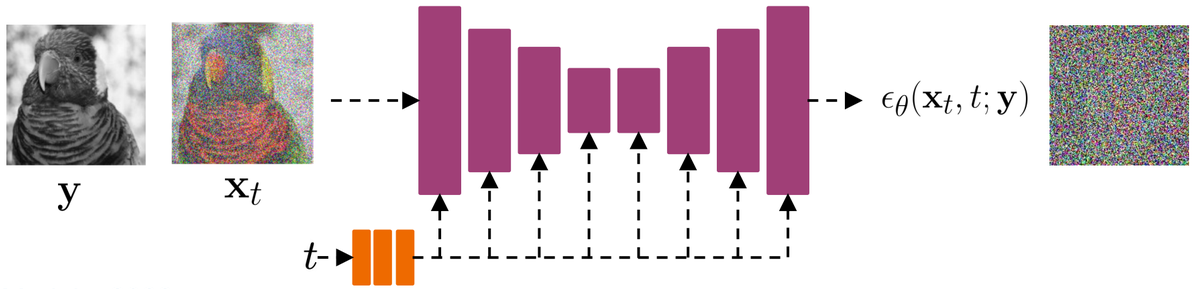
Many image-to-image translation applications can be considered as training $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ is the input image.

For example, for colorization, $\mathbf{x}$ is a colored image and $\mathbf{y}$ is a gray-level image.

Train a score model for $\mathbf{x}$ conditioned on $\mathbf{y}$ using:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\, \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\, \mathbb{E}_t\, ||\epsilon_\theta(\mathbf{x}_t, t; \mathbf{y}) - \epsilon||_p^p$$

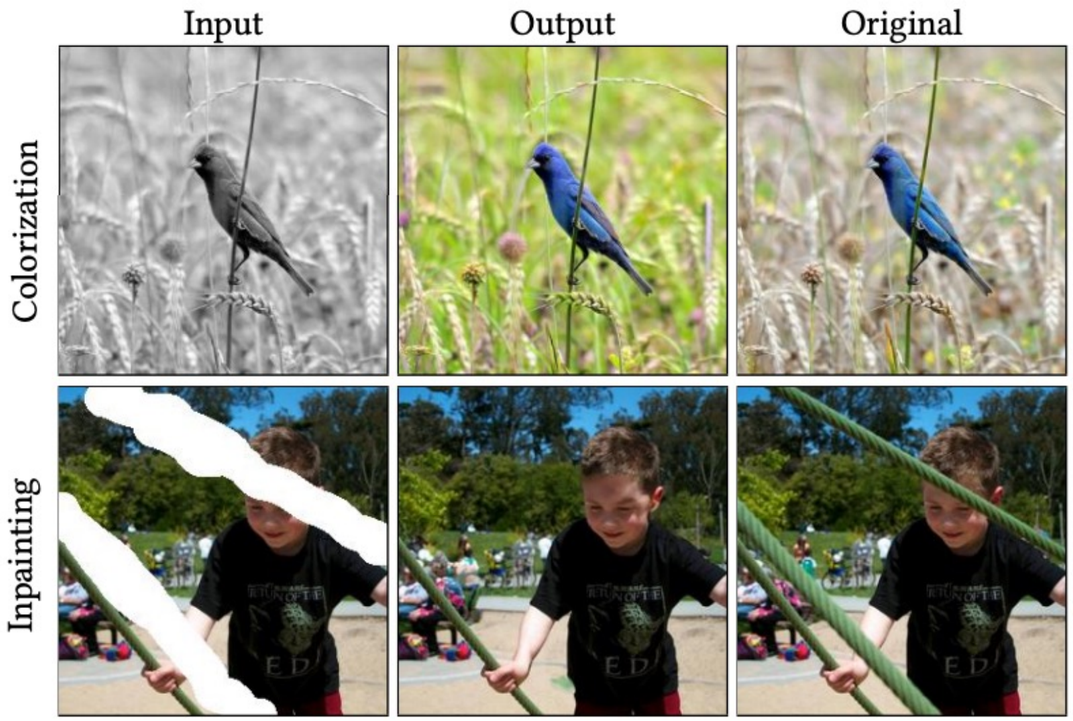
The conditional score is simply a U-Net with $\mathbf{x_t}$ and $\mathbf{y}$ concatenated.

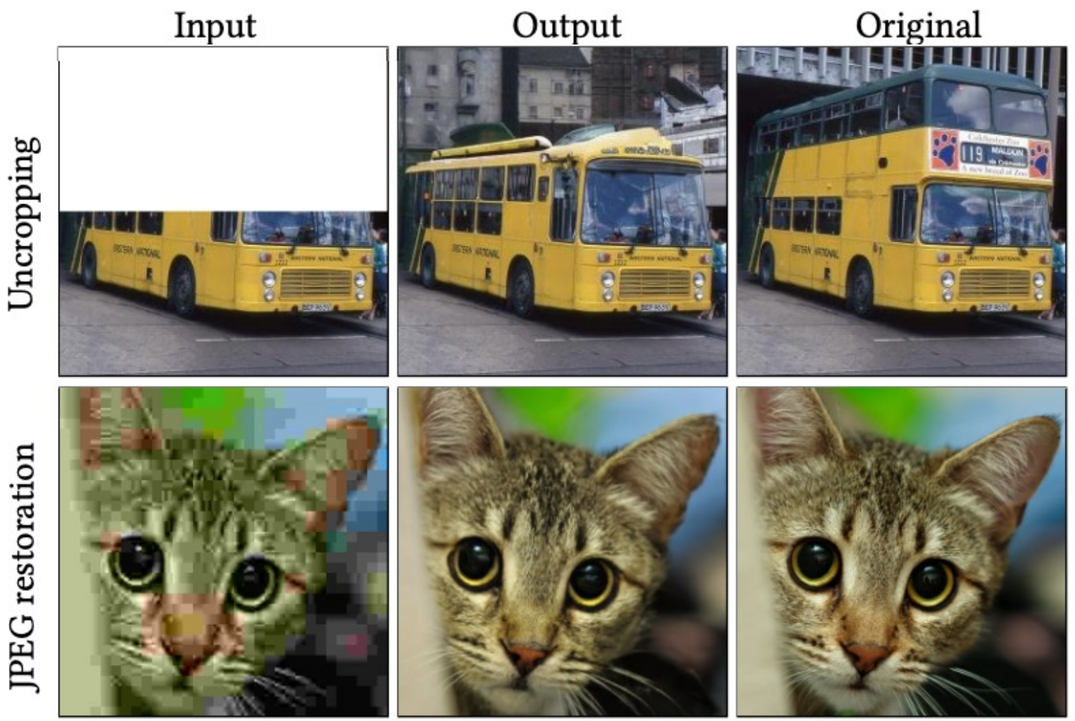


Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

Image-to-Image Translation

Palette: Image-to-Image Diffusion Models





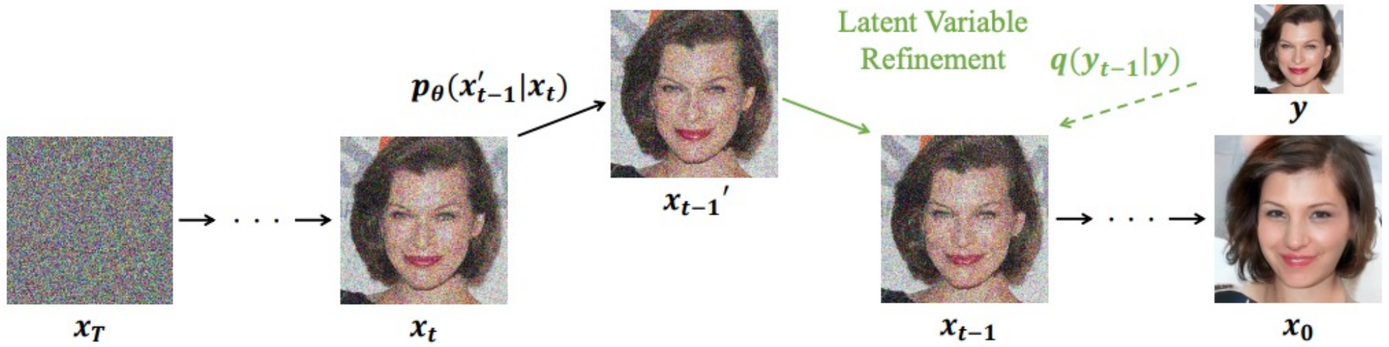
Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

Conditional Generation

Iterative Latent Variable Refinement (ILVR)

A simple technique to guide the generation process of an unconditional diffusion model using a reference image.

Given the conditioning (reference) image y the generation process is modified to pull the samples towards the reference image.

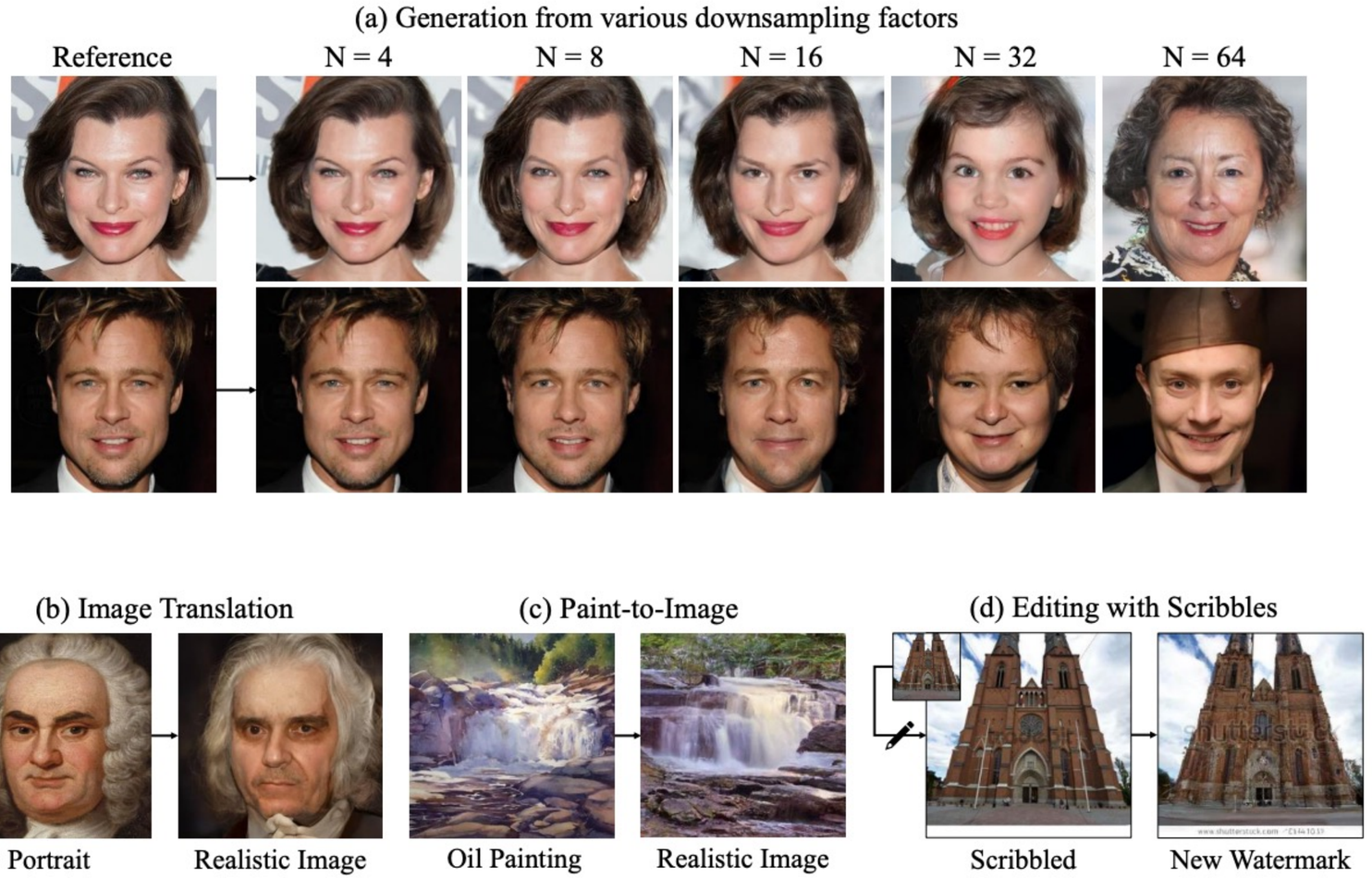


for $t = T, ..., 1$ **do**
  $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$
  $x'_{t-1} \sim p_\theta(x'_{t-1}|x_t)$ ▷ unconditional proposal
  $y_{t-1} \sim q(y_{t-1}|y)$ ▷ condition encoding
  $x_{t-1} \leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})$
end for

Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021

Conditional Generation

Iterative Latent Variable Refinement (ILVR)

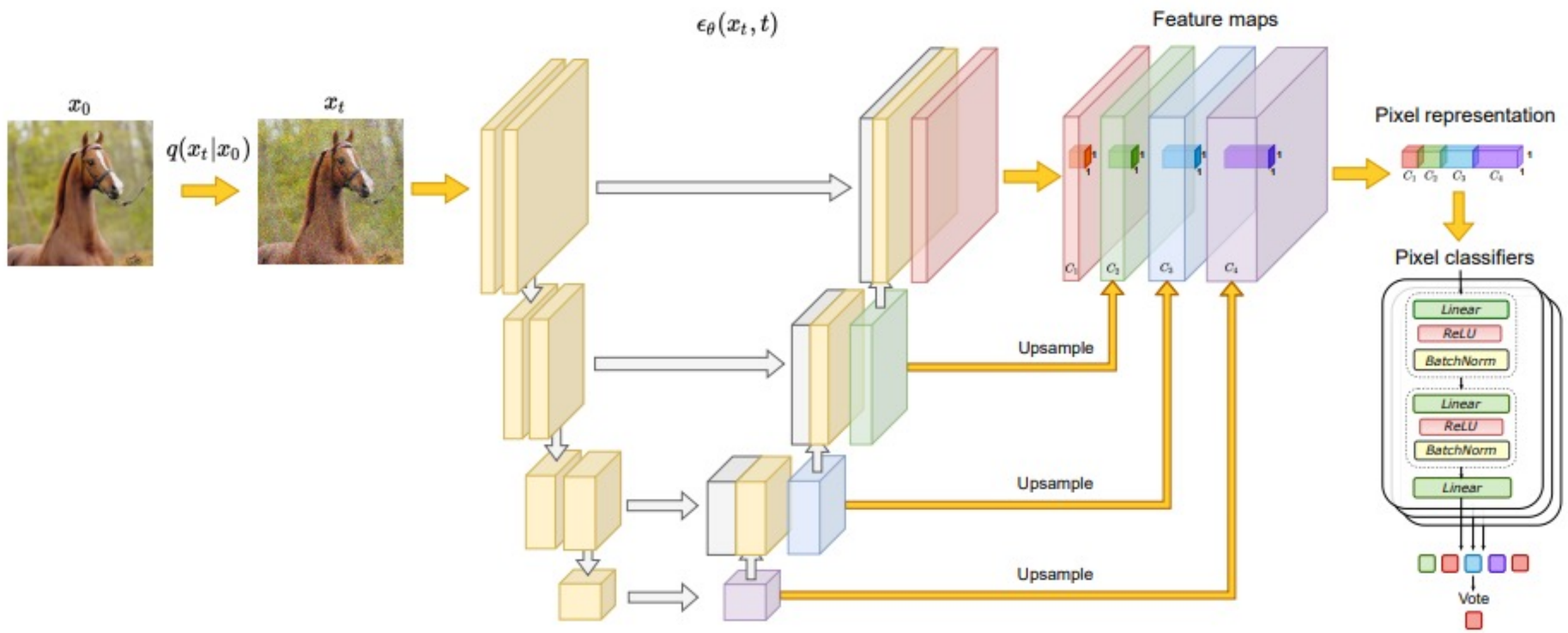


Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021

Semantic Segmentation

Label-efficient semantic segmentation with diffusion models

Can we use representation learned from diffusion models for downstream applications such as semantic segmentation?

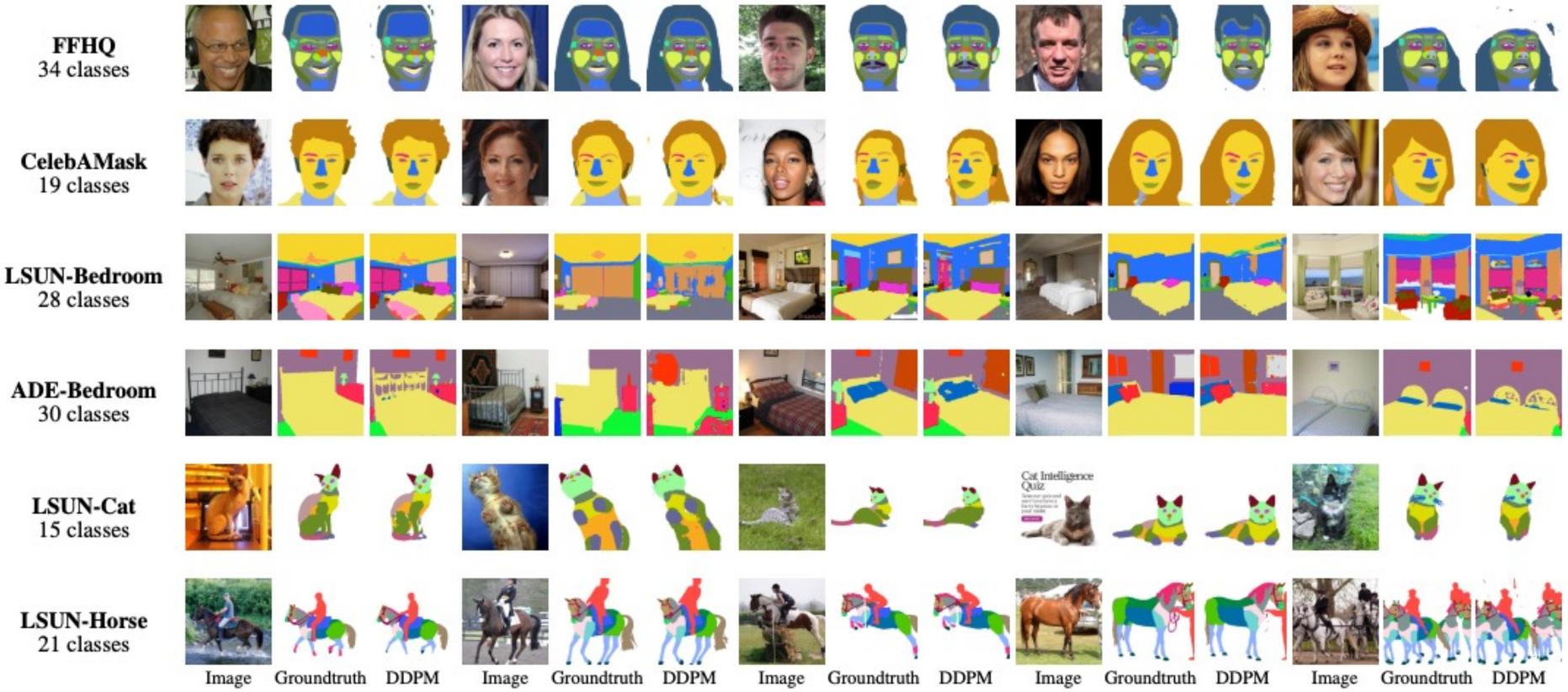


Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022

Semantic Segmentation

Label-efficient semantic segmentation with diffusion models

The experimental results show that the proposed method outperforms Masked Autoencoders, GAN and VAE-based models.

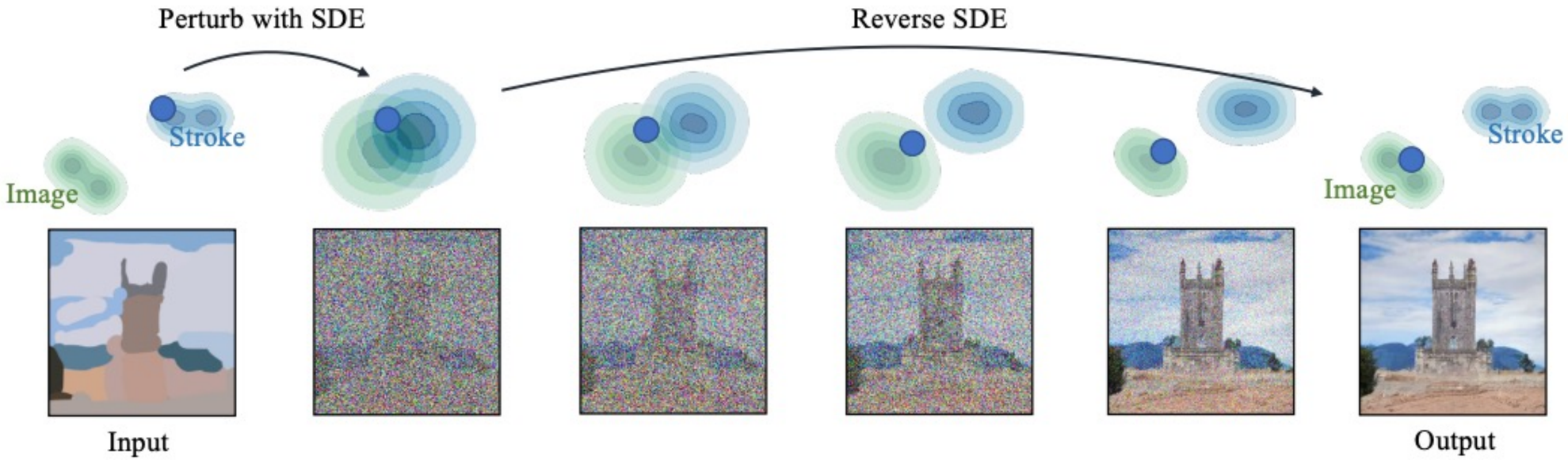


Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022

Image Editing

SDEdit

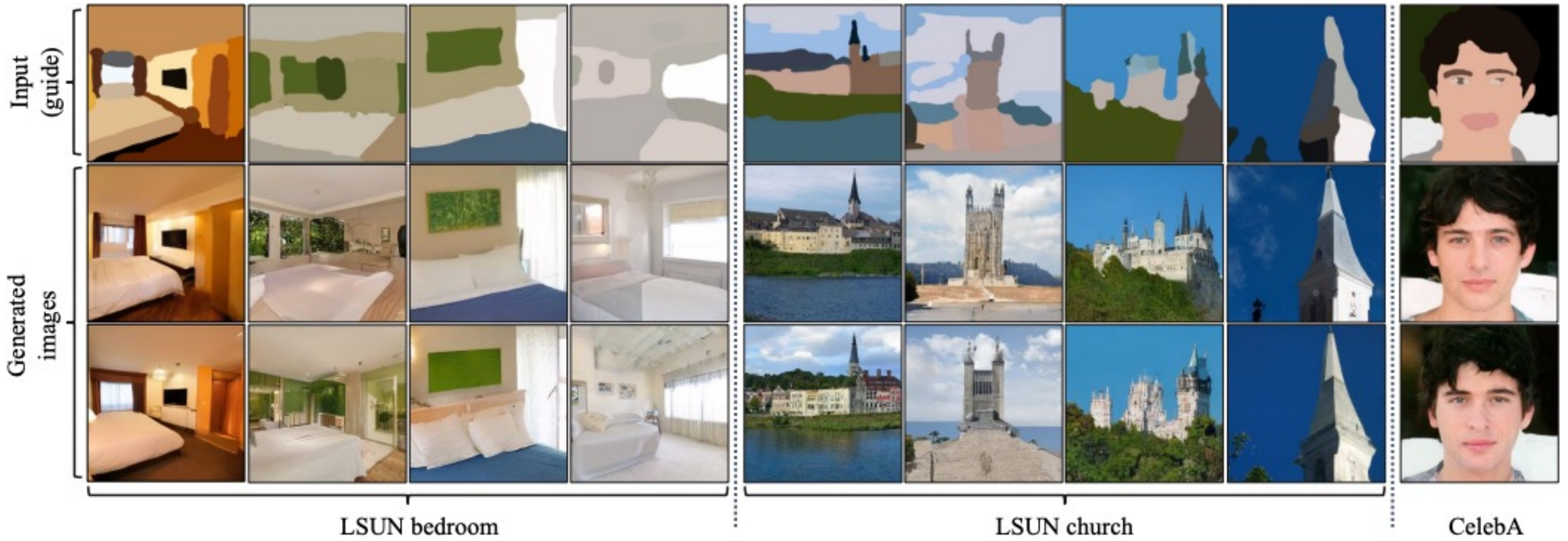
Forward diffusion brings two distributions close to each other



Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022

Image Editing

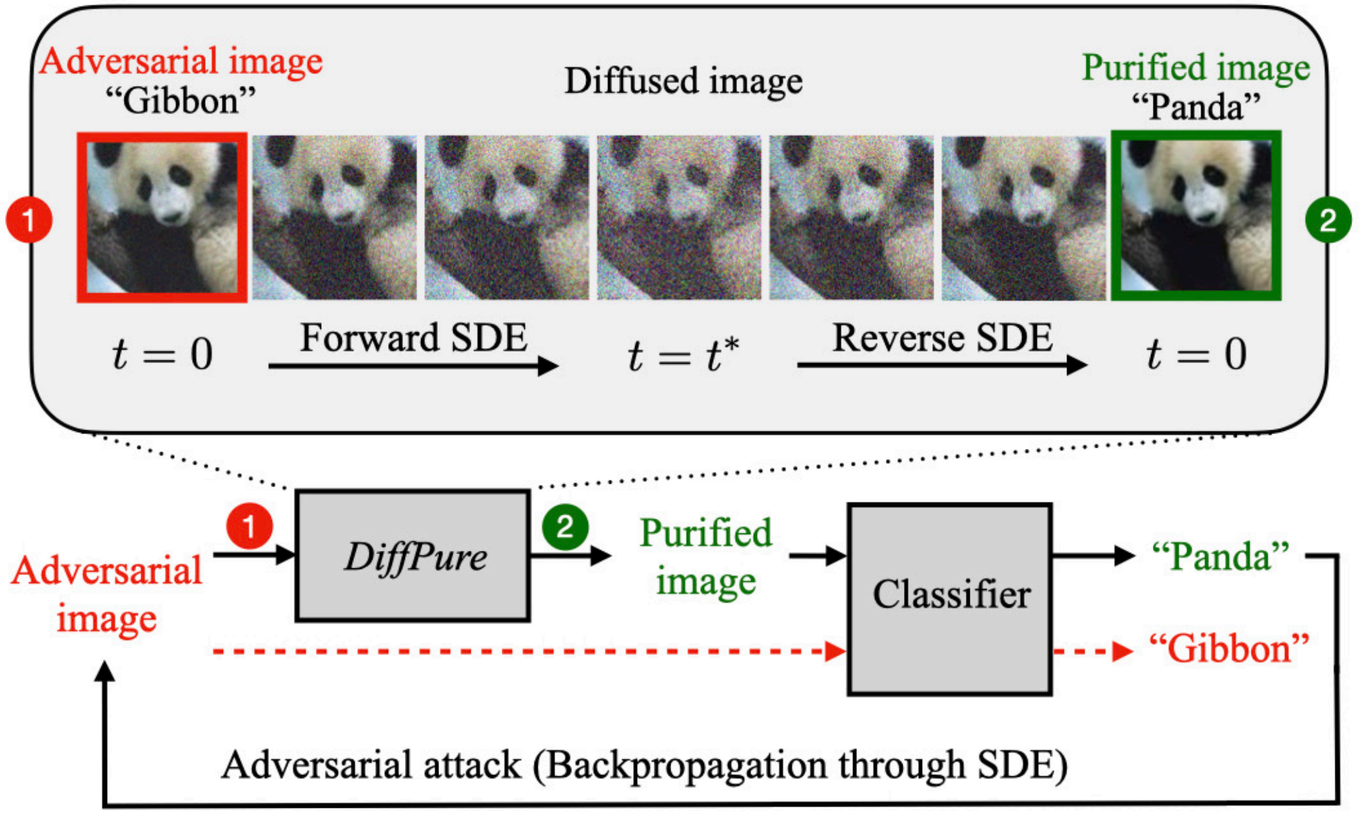
SDEdit



Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022

Adversarial Robustness

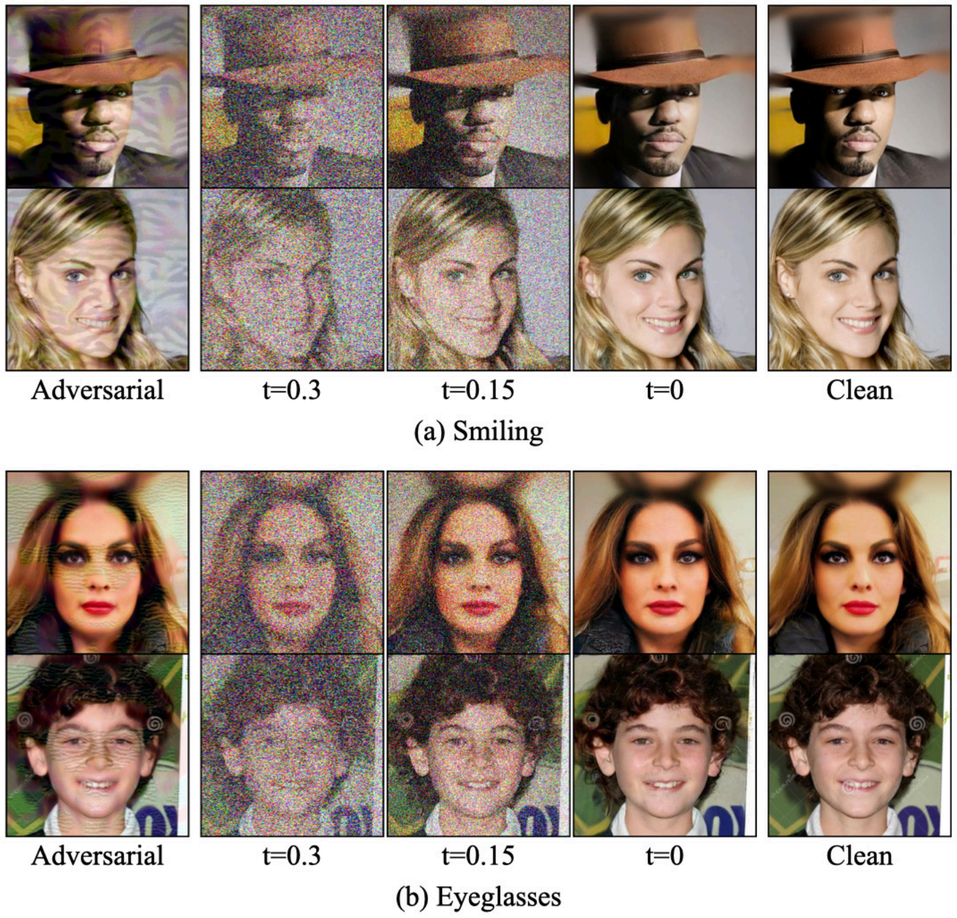
Diffusion Models for Adversarial Purification

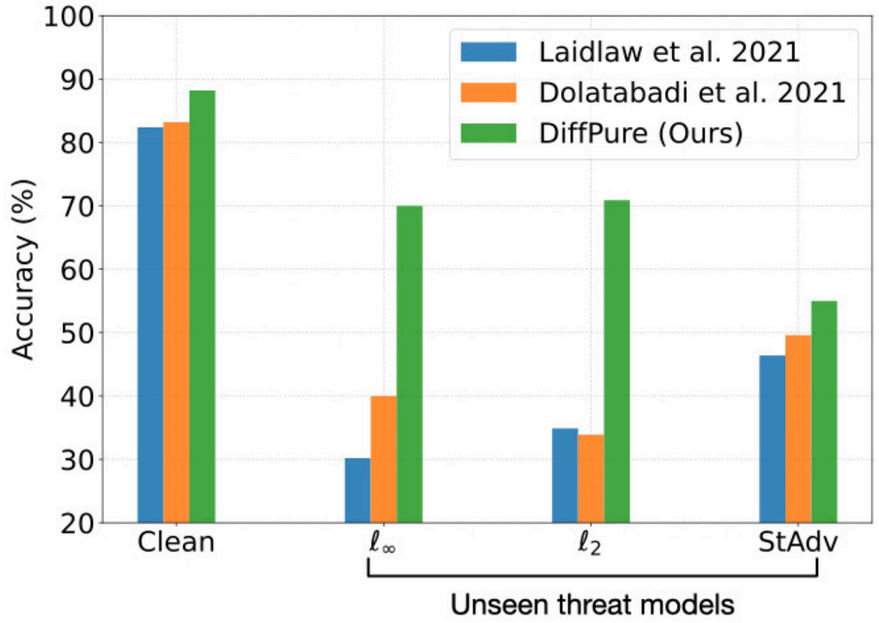


Nie et al., Diffusion Models for Adversarial Purification, ICML 2022

Adversarial Robustness

Diffusion Models for Adversarial Purification





Comparison with state-of-the-art defense methods against unseen threat models (including AutoAttack ℓ_∞, AutoAttack ℓ_2, and StdAdv) on ResNet-50 for CIFAR-10.

Nie et al., Diffusion Models for Adversarial Purification, ICML 2022

Applications (3):
Video Synthesis, Medical Imaging, 3D Generation, Discrete State Models

Video Generation

Samples from a text-conditioned video diffusion model, conditioned on the string *fireworks*.

(video from: Ho et al., “Video Diffusion Models”, *arXiv*, 2022,
https://video-diffusion.github.io/)

Ho et al., “Video Diffusion Models”, *arXiv*, 2022
Harvey et al., “Flexible Diffusion Modeling of Long Videos”, arXiv, 2022
Yang et al., “Diffusion Probabilistic Modeling for Video Generation”, *arXiv*, 2022
Höppe et al., “Diffusion Models for Video Prediction and Infilling”, arXiv, 2022
Voleti et al., “MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”, *arXiv*, 2022

Video Generation

Video Generation Tasks:

- Unconditional Generation (Generate all frames)
- Future Prediction (Generate future from past fames)
- Past Prediction (Generate past from future fames)
- Interpolation (Generate intermediate frames)

➡ Learn a model of the form:

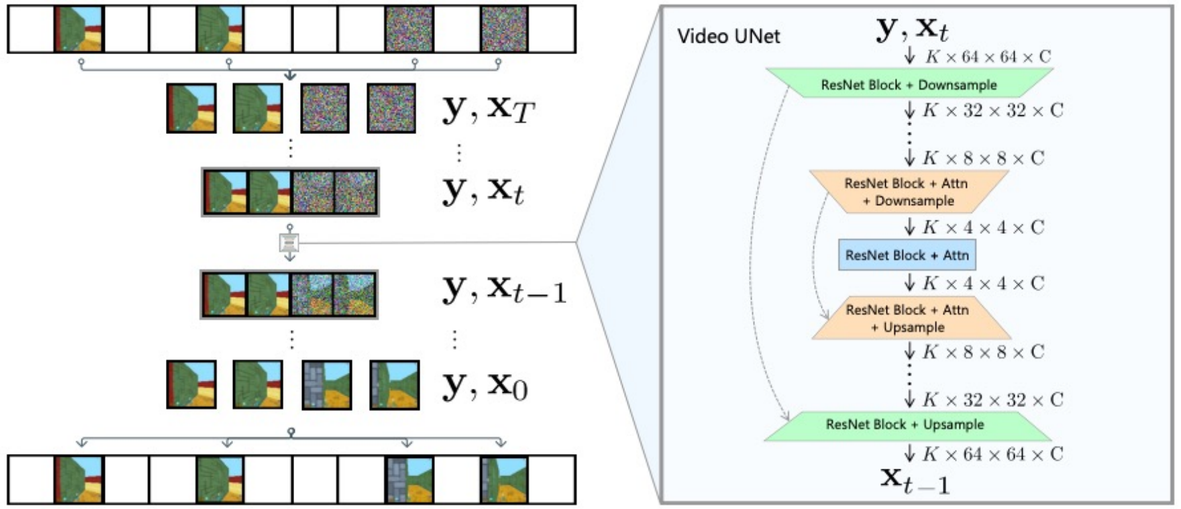
$$p_{\boldsymbol{\theta}}\left(\mathbf{x}^{t_1}, \cdots, \mathbf{x}^{t_K} \middle| \mathbf{x}^{\tau_1}, \cdots, \mathbf{x}^{\tau_M}\right)$$

Given frames: $\mathbf{x}^{\tau_1}, \cdots, \mathbf{x}^{\tau_M}$

Frames to be predicted: $\mathbf{x}^{t_1}, \cdots, \mathbf{x}^{t_K}$

➡ Learn one model for everything:

- Architecture as **one diffusion model** over **all frames concatenated.**
- Mask frames to be predicted; provide conditioning frames; vary applied masking/conditioning for different tasks during training.
- Use **time position encodings** to encode times.



(image from: Harvey et al., "Flexible Diffusion Modeling of Long Videos", *arXiv*, 2022)

Ho et al., "Video Diffusion Models", *arXiv*, 2022
Harvey et al., "Flexible Diffusion Modeling of Long Videos", arXiv, 2022
Yang et al., "Diffusion Probabilistic Modeling for Video Generation", *arXiv*, 2022
Höppe et al., "Diffusion Models for Video Prediction and Infilling", arXiv, 2022
Voleti et al., "MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation", *arXiv*, 2022

Video Generation

Architecture Details

Architecture Details:

Data is 4D (image height, image width, #frames, channels)

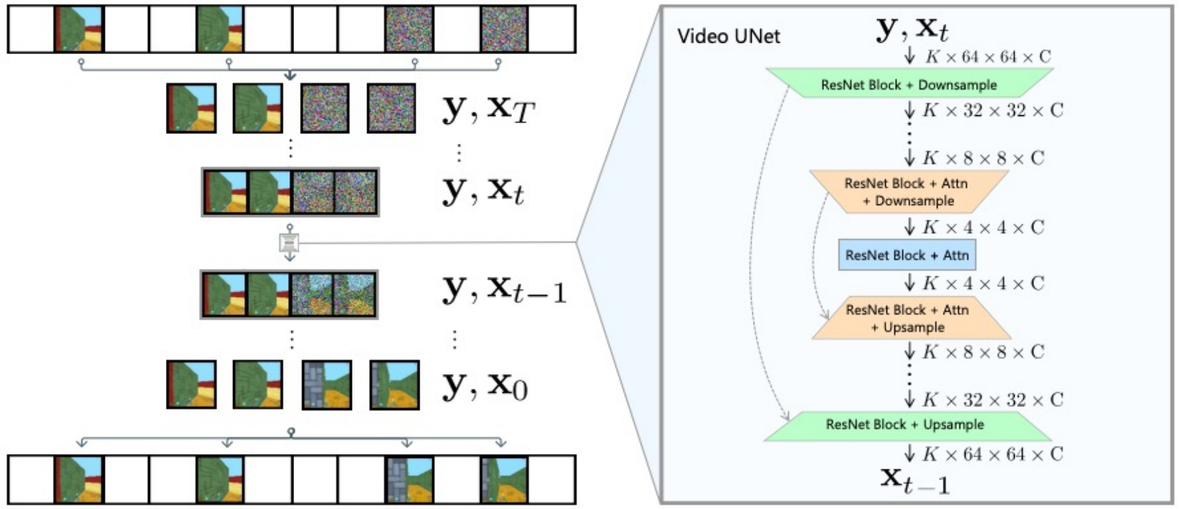
- Option (1): 3D Convolutions. Can be computationally expensive.
- Option (2): Spatial 2D Convolutions + Attention Layers along frame axis.

➡ Additional Advantage:

Ignoring the attention layers, the model can be trained additionally on pure image data!

➡ Learn one model for everything:

- Architecture as **one diffusion model** over **all frames concatenated**.
- Mask frames to be predicted; provide conditioning frames; vary applied masking/conditioning for different tasks during training.
- Use **time position encodings** to encode times.



(image from: Harvey et al., "Flexible Diffusion Modeling of Long Videos", *arXiv*, 2022)

Ho et al., "Video Diffusion Models", *arXiv*, 2022
Harvey et al., "Flexible Diffusion Modeling of Long Videos", arXiv, 2022
Yang et al., "Diffusion Probabilistic Modeling for Video Generation", *arXiv*, 2022
Höppe et al., "Diffusion Models for Video Prediction and Infilling", arXiv, 2022
Voleti et al., "MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation", *arXiv*, 2022

Video Generation

Results

Long term video generation in hierarchical manner:

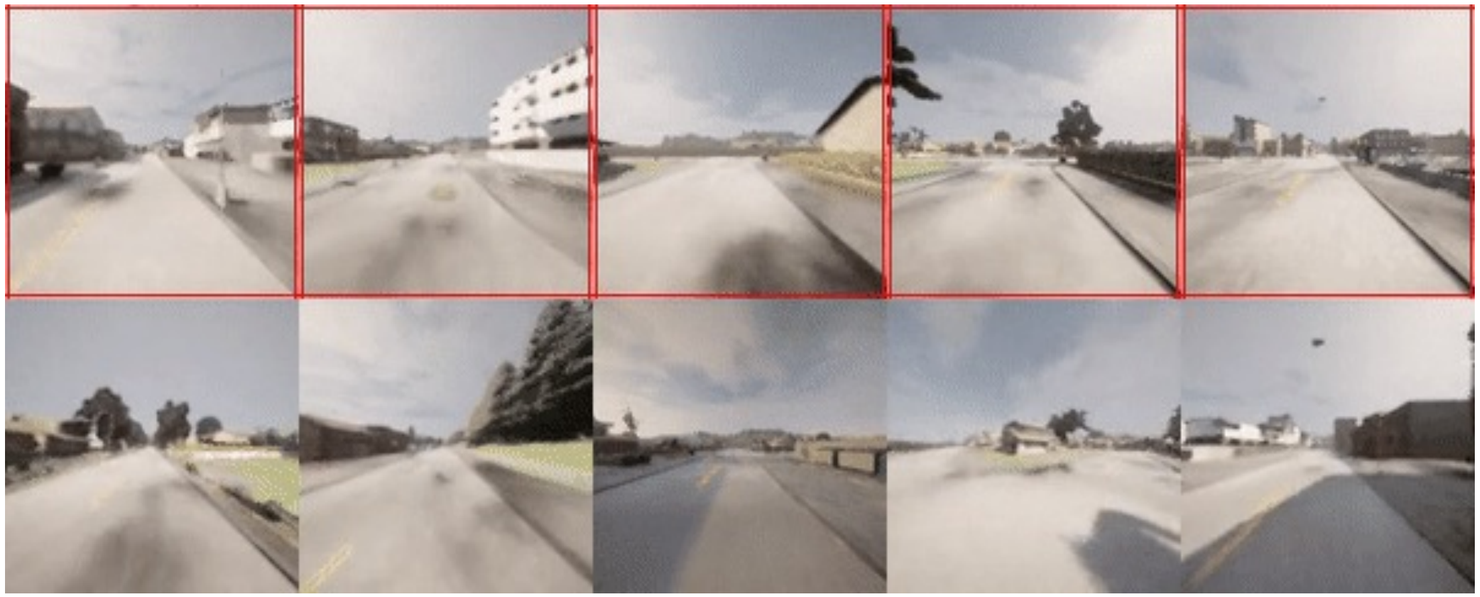
- 1. Generate future frames in sparse manner, conditioning on frames far back
- 2. Interpolate in-between frames

⬇

1+ hour coherent video generation possible!

Test Data:

Generated:

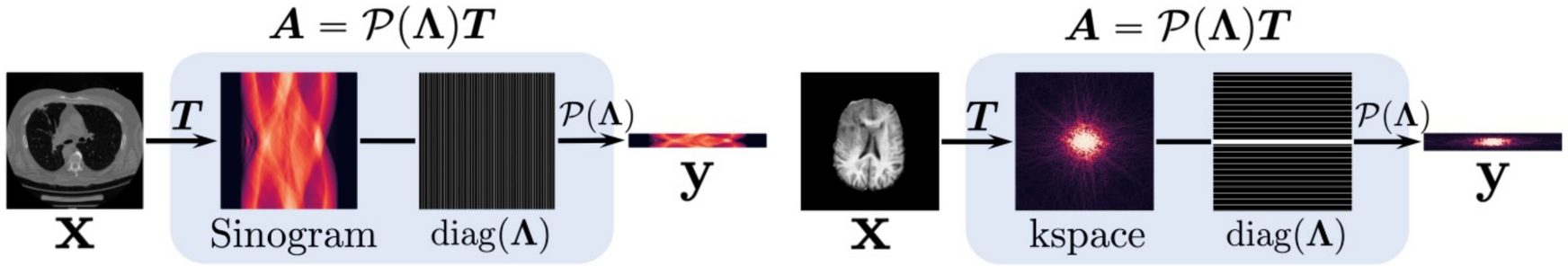


(video from: Harvey et al., “Flexible Diffusion Modeling of Long Videos”, *arXiv*, 2022, https://plai.cs.ubc.ca/2022/05/20/flexible-diffusion-modeling-of-long-videos/)

Ho et al., “Video Diffusion Models”, *arXiv*, 2022
Harvey et al., “Flexible Diffusion Modeling of Long Videos”, arXiv, 2022
Yang et al., “Diffusion Probabilistic Modeling for Video Generation”, *arXiv*, 2022
Höppe et al., “Diffusion Models for Video Prediction and Infilling”, arXiv, 2022
Voleti et al., “MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”, *arXiv*, 2022

Solving Inverse Problems in Medical Imaging

Forward CT or MRI imaging process (simplified):

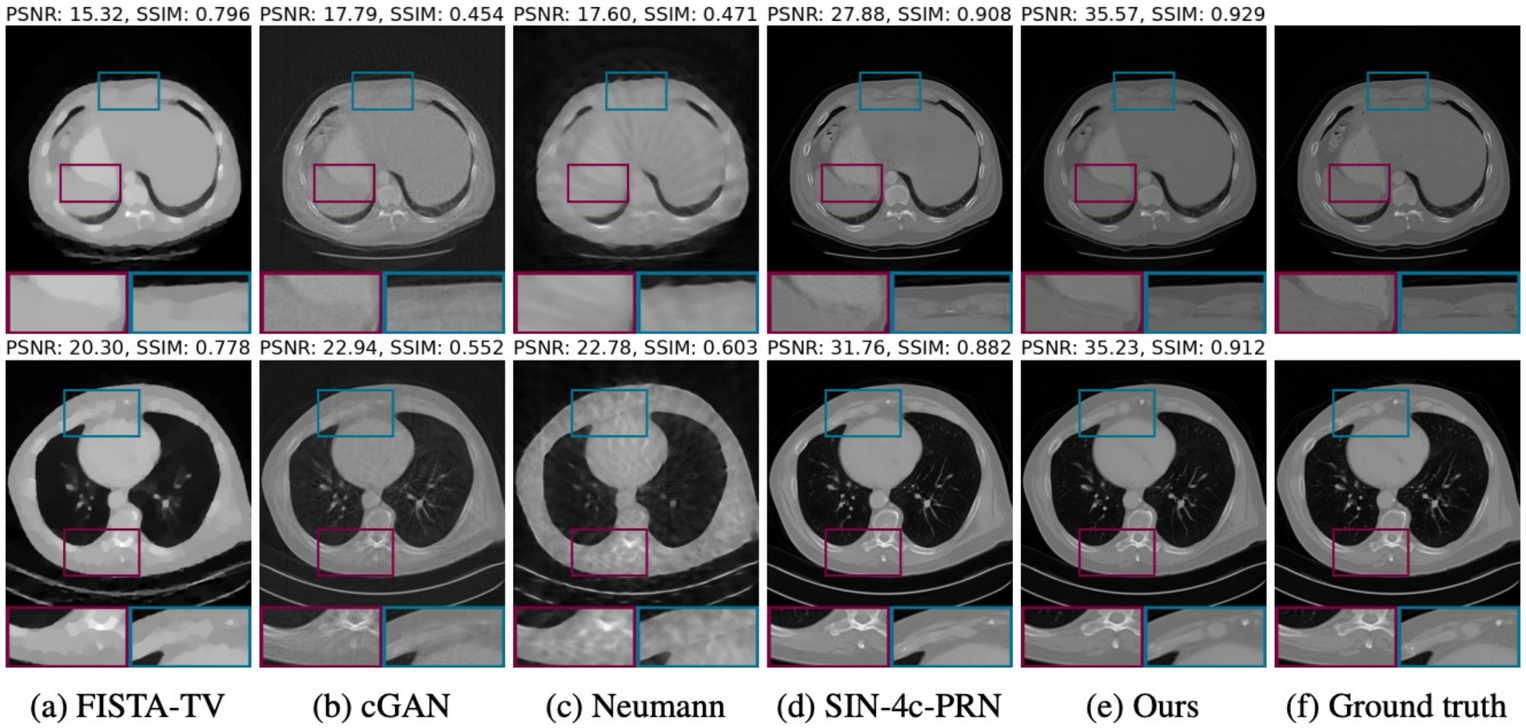


(image from: Song et al., “Solving Inverse Problems in Medical Imaging with Score-Based Generative Models”, *ICLR*, 2022)

➡ *Inverse Problem:*
Reconstruct original image from sparse measurements.

Solving Inverse Problems in Medical Imaging

High-level idea: Learn Generative Diffusion Model as “prior”; then guide synthesis conditioned on sparse observations:



(image from: Song et al., “Solving Inverse Problems in Medical Imaging with Score-Based Generative Models”, *ICLR*, 2022)

➡ *Outperforms even fully-supervised methods.*

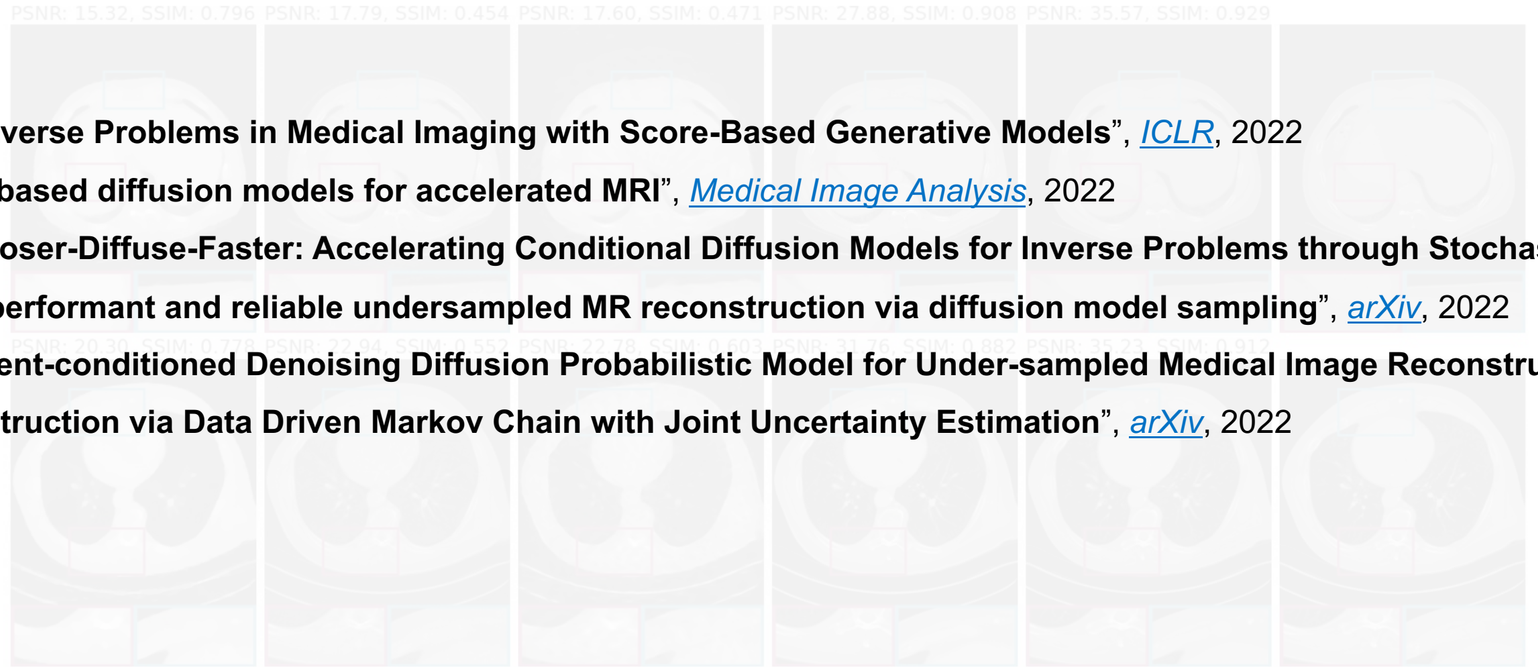
Song et al., “Solving Inverse Problems in Medical Imaging with Score-Based Generative Models”, *ICLR*, 2022

Solving Inverse Problems in Medical Imaging

Lots of Literature

High-level idea: Learn Generative Diffusion Model as "prior"; then guide synthesis conditioned on sparse observations:

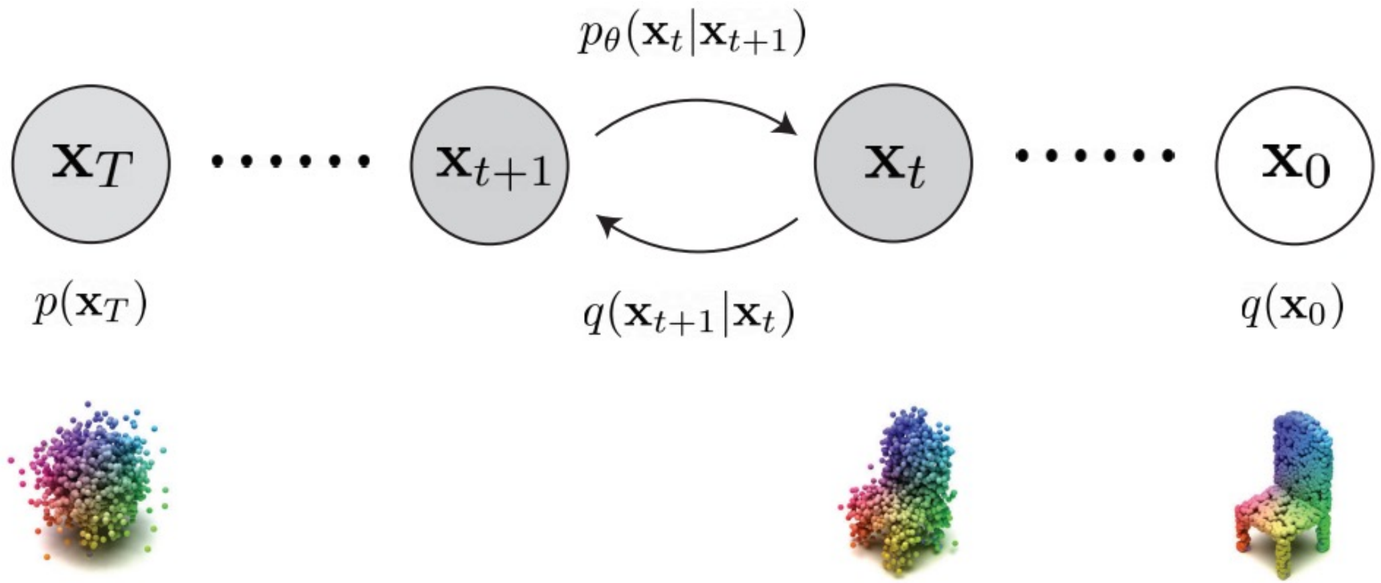
- Song et al., "**Solving Inverse Problems in Medical Imaging with Score-Based Generative Models**", *ICLR*, 2022
- Chung and Ye, "**Score-based diffusion models for accelerated MRI**", *Medical Image Analysis*, 2022
- Chung et al., "**Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction**", *CVPR*, 2022
- Peng et al., "**Towards performant and reliable undersampled MR reconstruction via diffusion model sampling**", *arXiv*, 2022
- Xie and Li, "**Measurement-conditioned Denoising Diffusion Probabilistic Model for Under-sampled Medical Image Reconstruction**", *arXiv*, 2022
- Luo et al, "**MRI Reconstruction via Data Driven Markov Chain with Joint Uncertainty Estimation**", *arXiv*, 2022
- …



(Song et al., "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models", *ICLR*, 2022)

3D Shape Generation

- Point clouds as 3D shape representation can be diffused easily and intuitively
- Denoiser implemented based on modern point cloud-processing networks (PointNets & Point-VoxelCNNs)

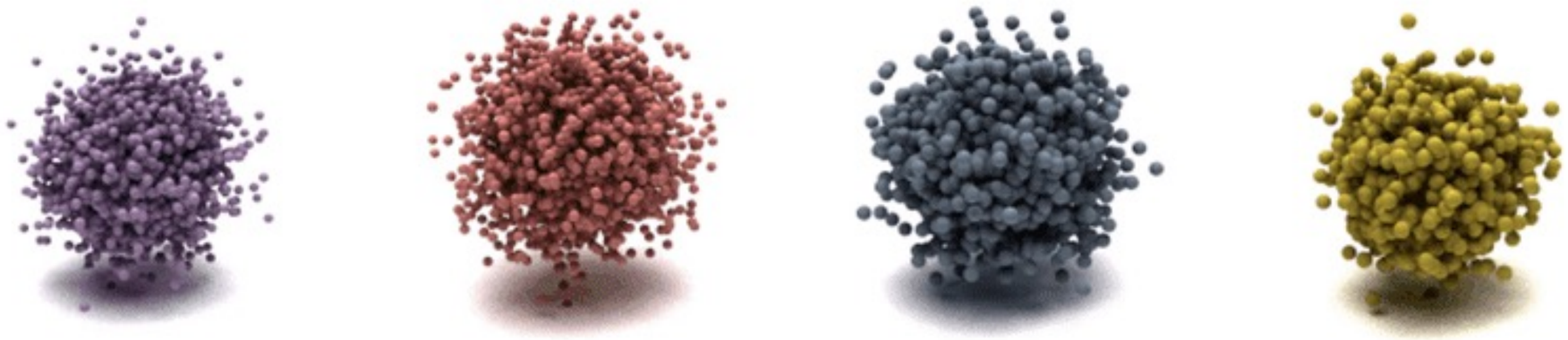


(image from: Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021)

Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021
Luo and Hu, “Diffusion Probabilistic Models for 3D Point Cloud Generation”, *CVPR*, 2021

3D Shape Generation

- Point clouds as 3D shape representation can be diffused easily and intuitively
- Denoiser implemented based on modern point cloud-processing networks (PointNets & Point-VoxelCNNs)



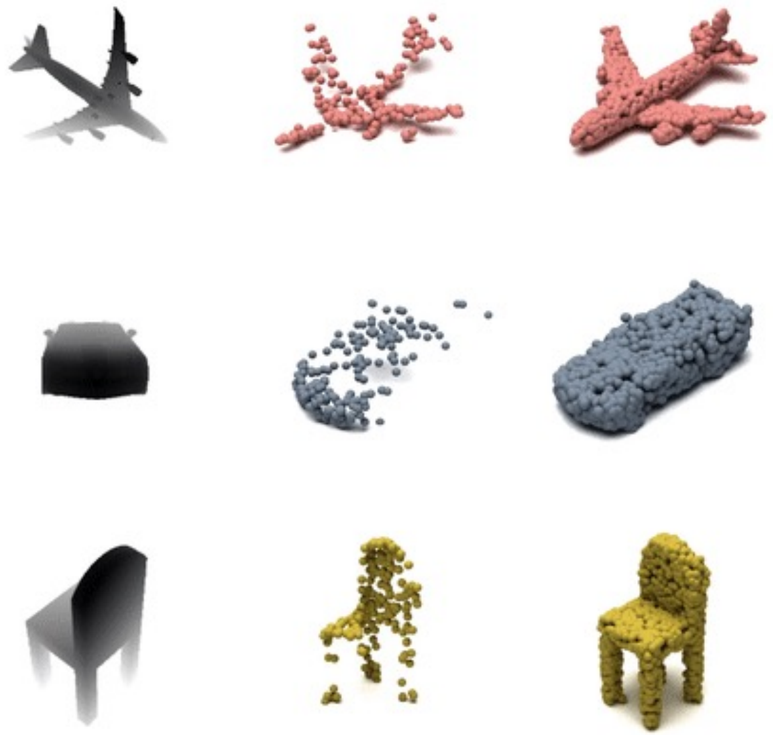
(video from: Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021, https://alexzhou907.github.io/pvd)

Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021

3D Shape Generation

Shape Completion

- Can train conditional shape completion diffusion model (subset of points fixed to given conditioning points):



(video from: Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021, https://alexzhou907.github.io/pvd)

Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021

3D Shape Generation

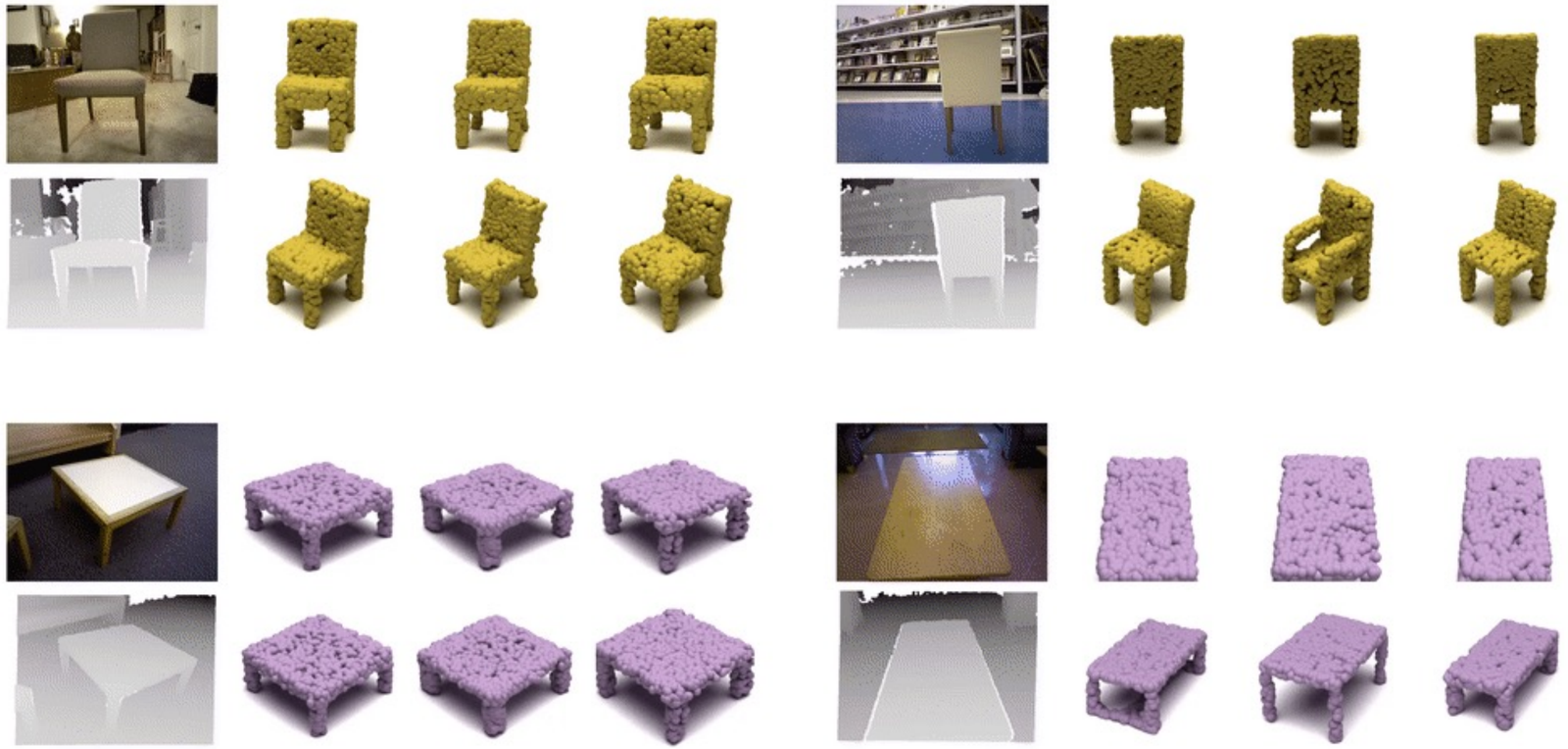
Shape Completion – Multimodality

(video from: Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021, https://alexzhou907.github.io/pvd)

Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021

3D Shape Generation

Shape Completion – Multimodality – On Real Data



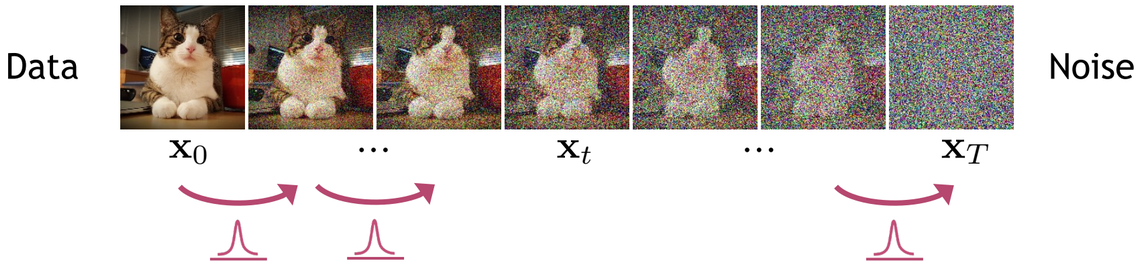
(video from: Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021, https://alexzhou907.github.io/pvd)

Zhou et al., “3D Shape Generation and Completion through Point-Voxel Diffusion”, *ICCV*, 2021

Towards Discrete State Diffusion Models

- So far:

  Continuous diffusion and denoising processes.



Fixed forward diffusion process: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$

Reverse generative process: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$

➡ *But what if data is discrete? Categorical?*
Continuous perturbations are not possible!

(Text, Pixel-wise Segmentation Labels,
Discrete Image Encodings, etc.)

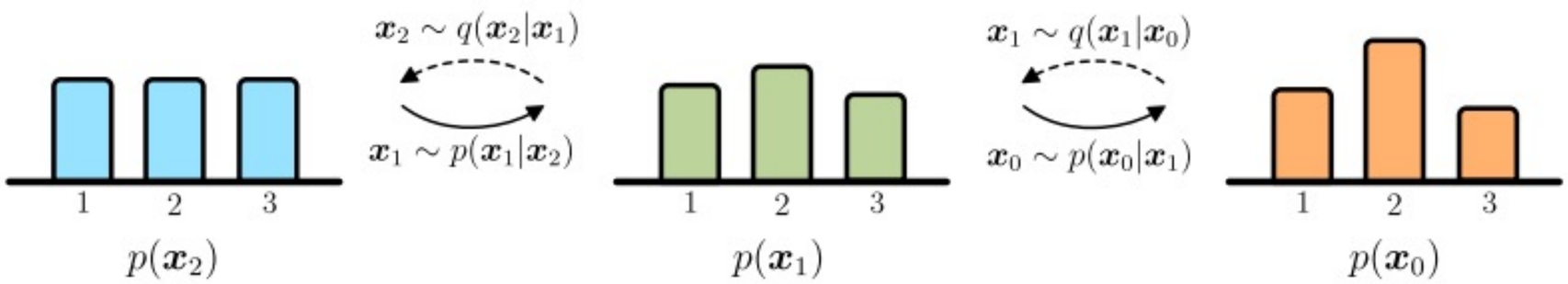
Discrete State Diffusion Models

- **Categorical diffusion:** $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathrm{Cat}(\mathbf{x}_t; \mathbf{p} = \mathbf{x}_{t-1}\mathbf{Q}_t)$

  $\mathbf{x}_t$: one-hot state vector

  $\mathbf{Q}_t$: transition matrix $[\mathbf{Q}_t]_{ij} = q(x_t = j|x_{t-1} = i)$

- Reverse process can be parametrized categorical distribution.



(image from: Hoogeboom et al., “Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions”, *NeurIPS*, 2022)

Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021
Hoogeboom et al., “Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions”, *NeurIPS*, 2022

Discrete State Diffusion Models

Options for forward process:

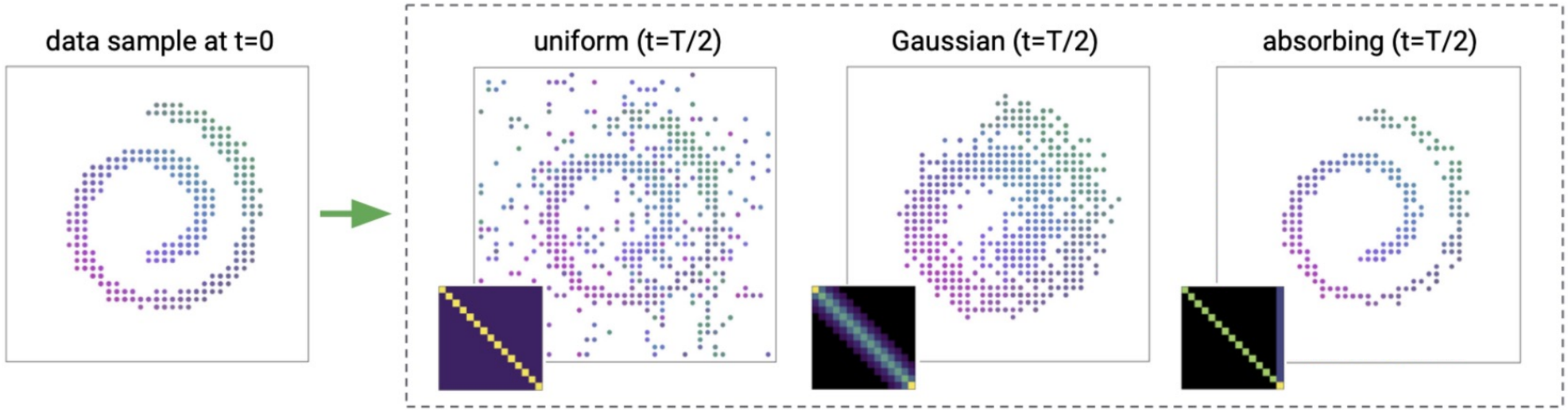
- Uniform categorical diffusion:

$$\mathbf{Q}_t = (1-\beta_t)\mathbf{I} + \frac{\beta_t}{K}\mathbb{1}\mathbb{1}^\top$$

- Progressive masking out of data (generation is “de-masking”)
- Tailored to ordinal data (e.g. discretized Gaussian)

$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

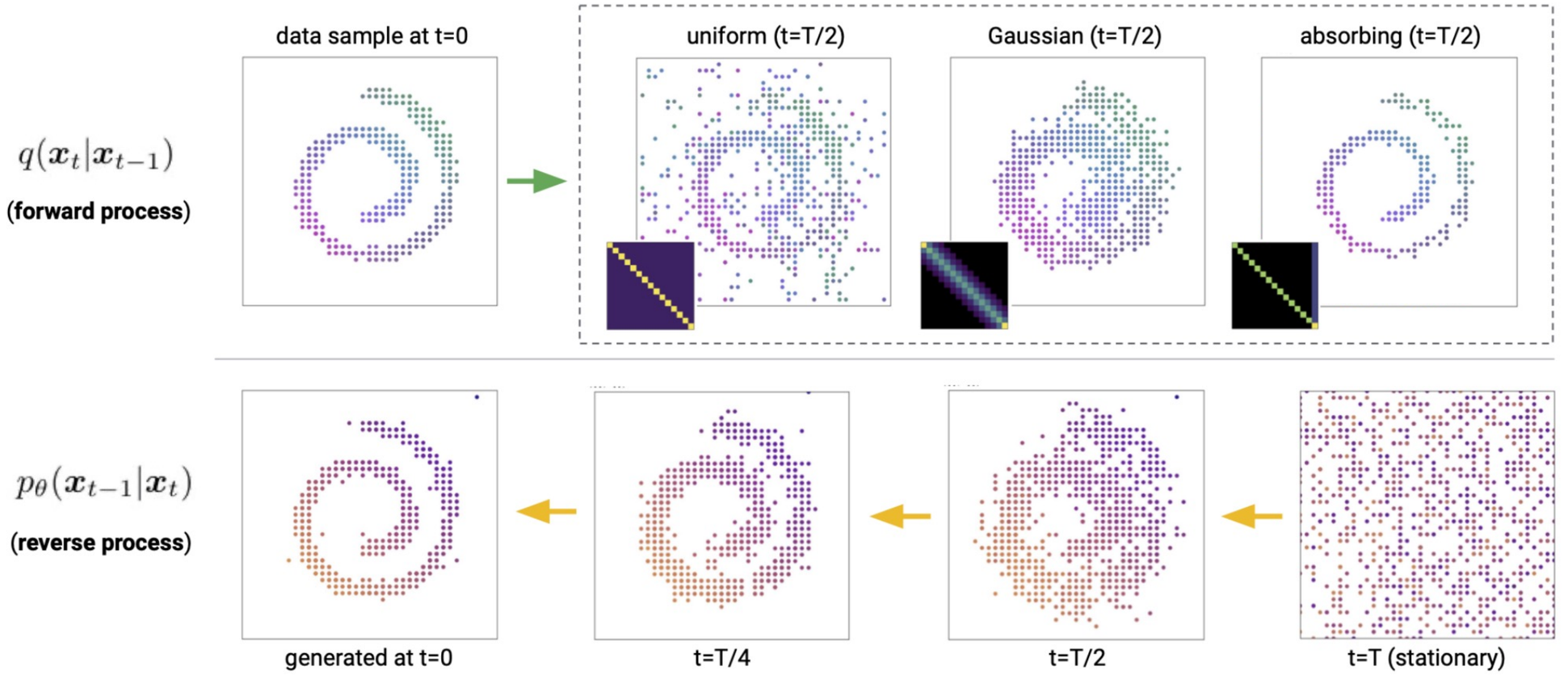
(forward process)



(image from: Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021)

Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021

Discrete State Diffusion Models



(image from: Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021)

Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021

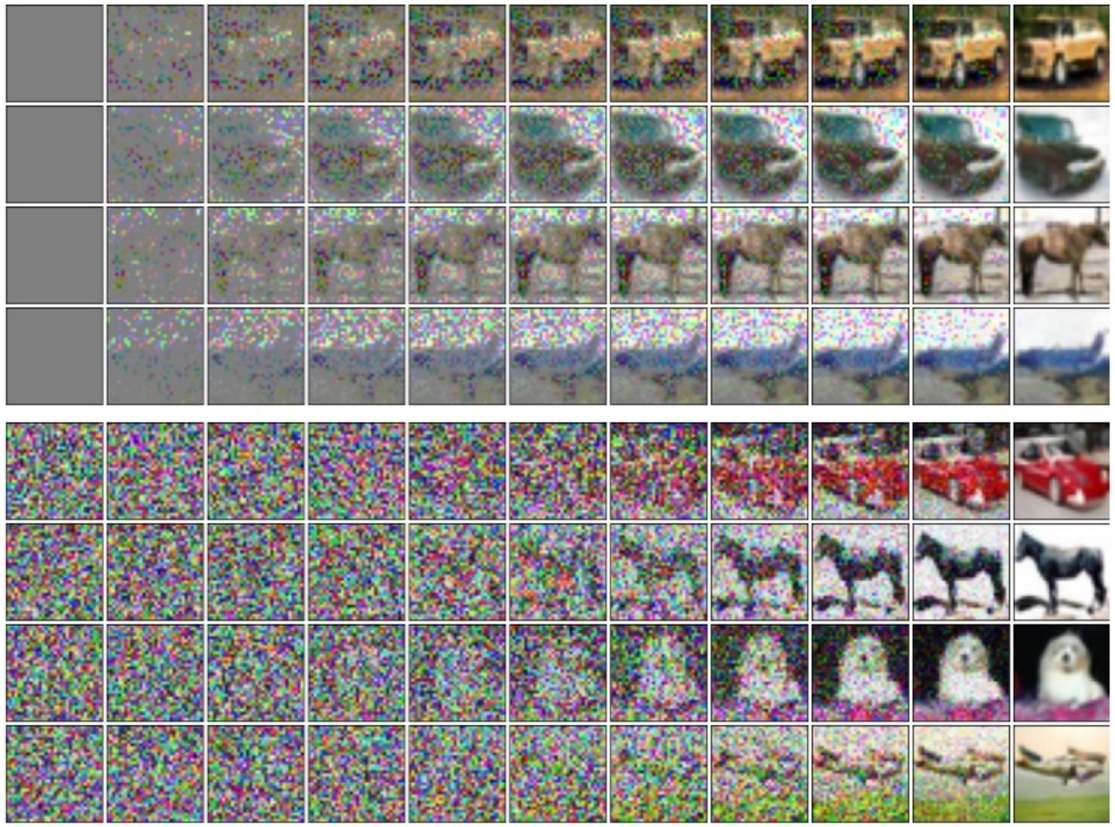
Discrete State Diffusion Models

Modeling Categorical Image Pixel Values

Progressive denoising starting from all-masked state.

Progressive denoising starting from random uniform state.

(with discretized Gaussian denoising model)

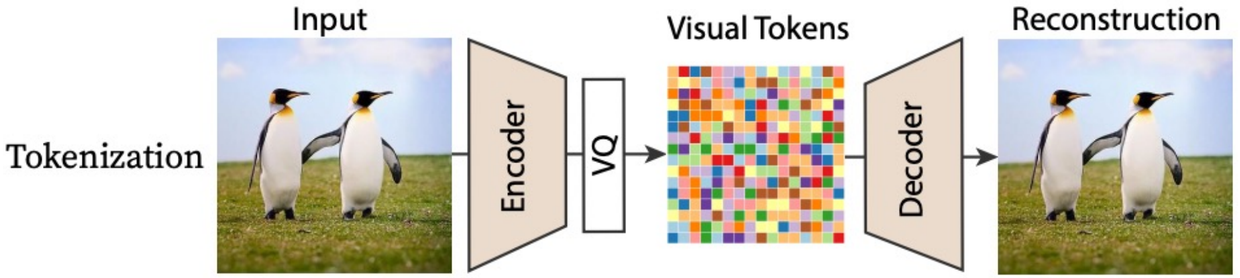


(image from: Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021)

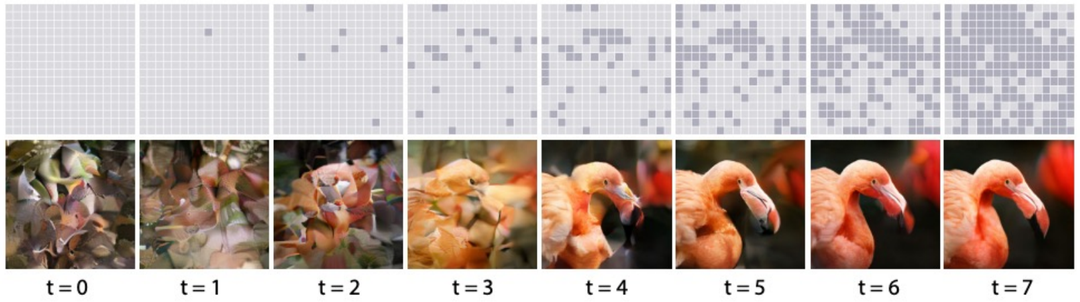
Austin et al., “Structured Denoising Diffusion Models in Discrete State-Spaces”, *NeurIPS*, 2021

Discrete State Diffusion Models

Modeling Discrete Image Encodings



Encoding images into latent space with discrete tokens, and modeling discrete token distribution



Iterative generation

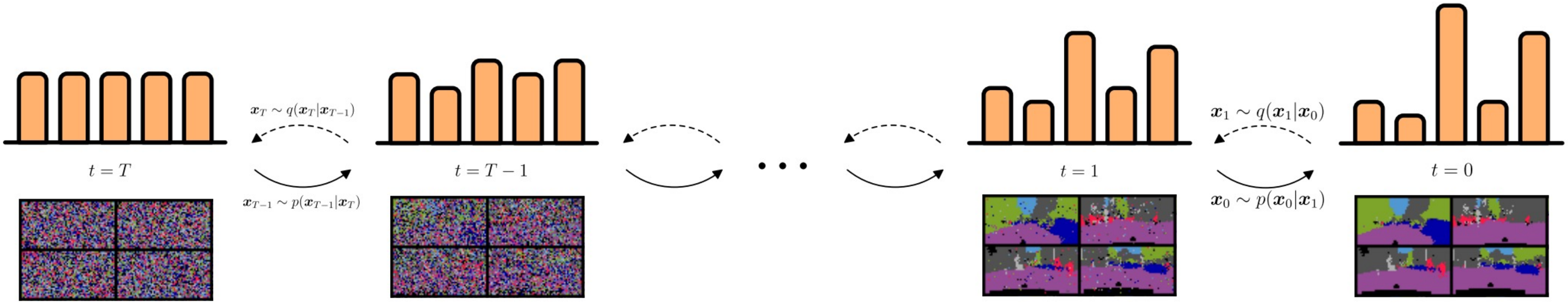
Class-conditional model samples

(images from: Chang et al., “MaskGIT: Masked Generative Image Transformer”, *CVPR*, 2022)

Chang et al., “MaskGIT: Masked Generative Image Transformer”, *CVPR*, 2022
Esser et al., “ImageBART: Bidirectional Context with Multinomial Diffusion for Autoregressive Image Synthesis”, *NeurIPS*, 2021

Discrete State Diffusion Models

Modeling Pixel-wise Segmentations



(image from: Hoogeboom et al., “Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions”, *NeurIPS*, 2022)

Hoogeboom et al., “Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions”, *NeurIPS*, 2022