Towards *Scalable* and Knowledgeable Generative Intelligence

Jiatao Gu
Machine Learning Researcher
Apple

Human Brain is a Prediction Machine

NeuroscienceNews.com

Your Brain Is a Prediction Machine That Is Always Active

Featured Neuroscience · August 4, 2022

Summary: *The brain constantly acts as a prediction machine, continuously comparing sensory information with internal predictions.*

Source: *Max Planck Institute*

This is in line with a recent theory on how our brain works: it is a prediction machine, which continuously compares sensory information that we pick up (such as images, sounds and language) with internal predictions.

Computer Science Topics Archive

NEUROSCIENCE

To Be Energy-Efficient, Brains Predict Their Perceptions

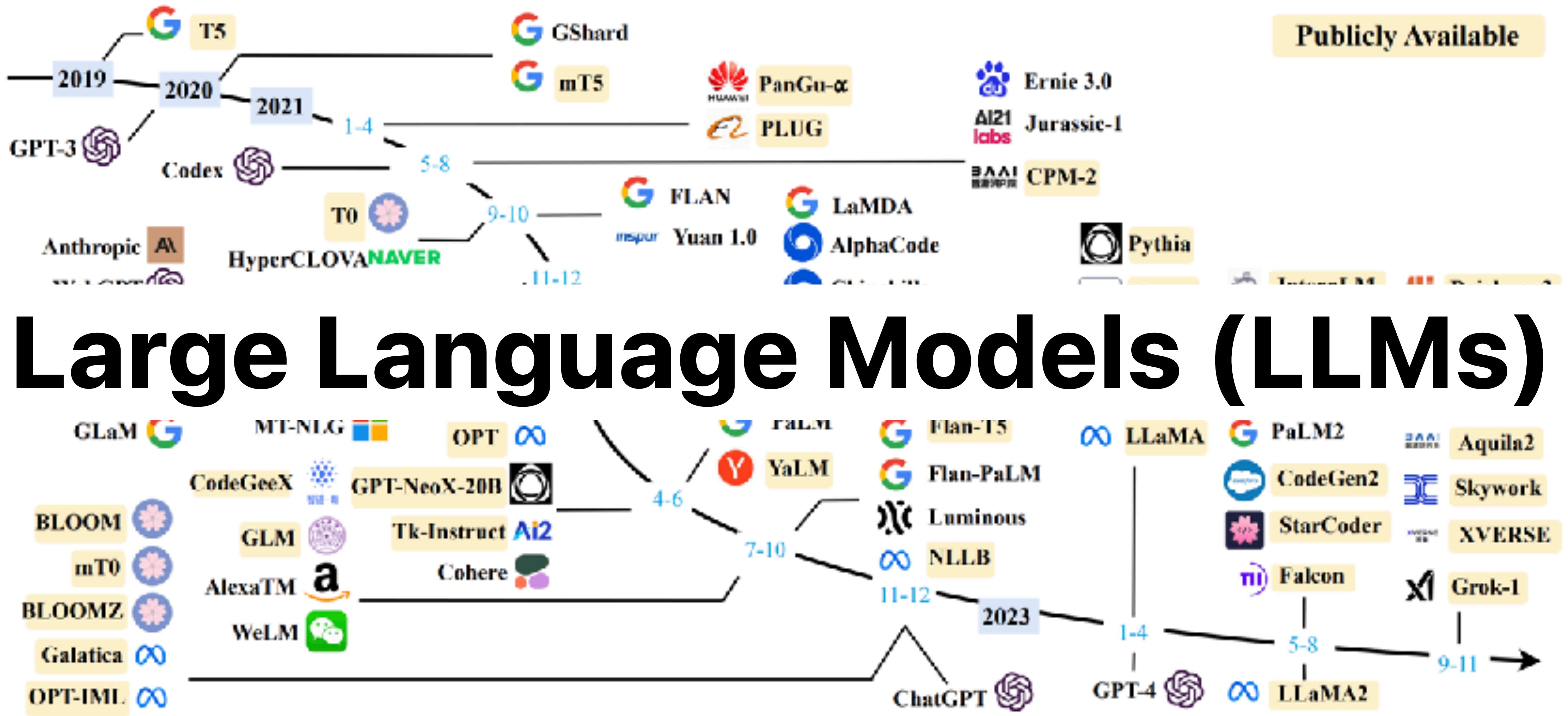
22

Results from neural networks support the idea that brains are "prediction machines" — and that they work that way to conserve energy.

Generative AI

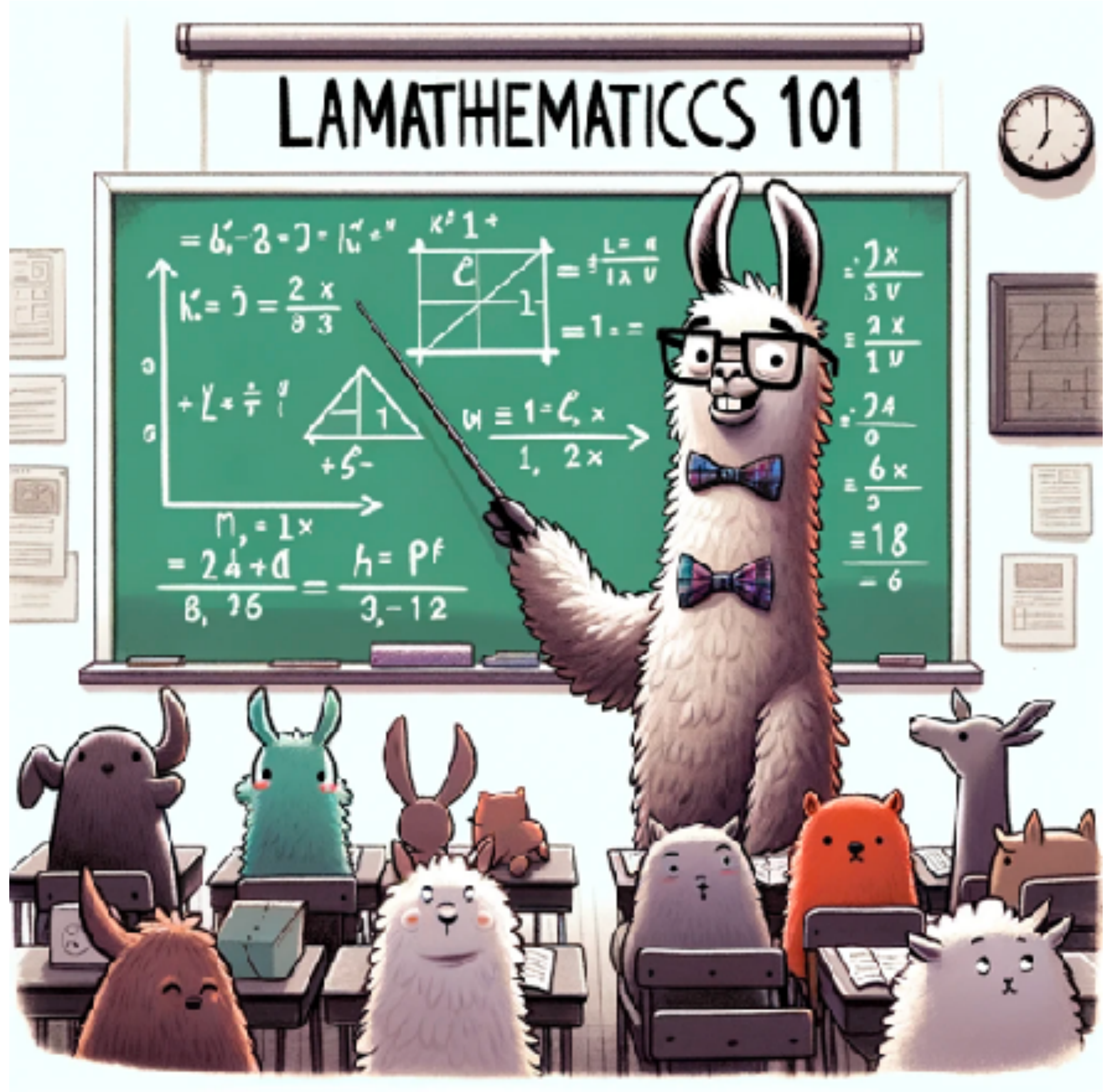
Make the "prediction" of the real world inside a computer.

Success of Generative AI



Large Language Models (LLMs)

Generative Models for Text, Image, Video, 3D, and Multimodal Generation



Prompt Question: What is the dog carrying?

Model Generation: Stick

Prompt: Describe the given image in very fine detail.

Model Generation: In this image, there is a dog holding a stick in its mouth. There is grass on the surface. In the background of the image, there are trees.

Generative AI has huge impacts

Robotics

Biomedicine

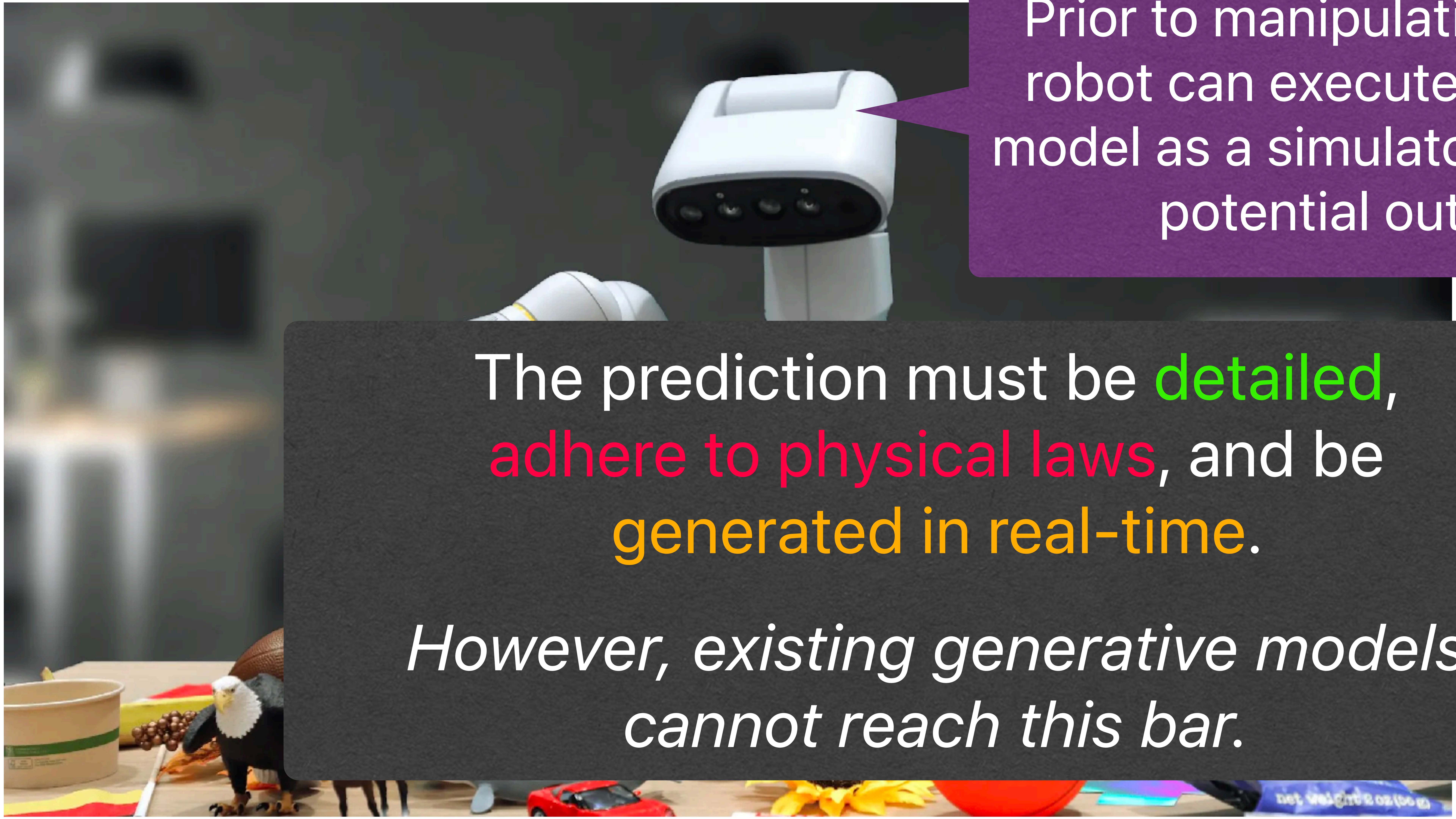
AI Agents

Education

Entertainment

Healthcare

Expectation of Generative AI



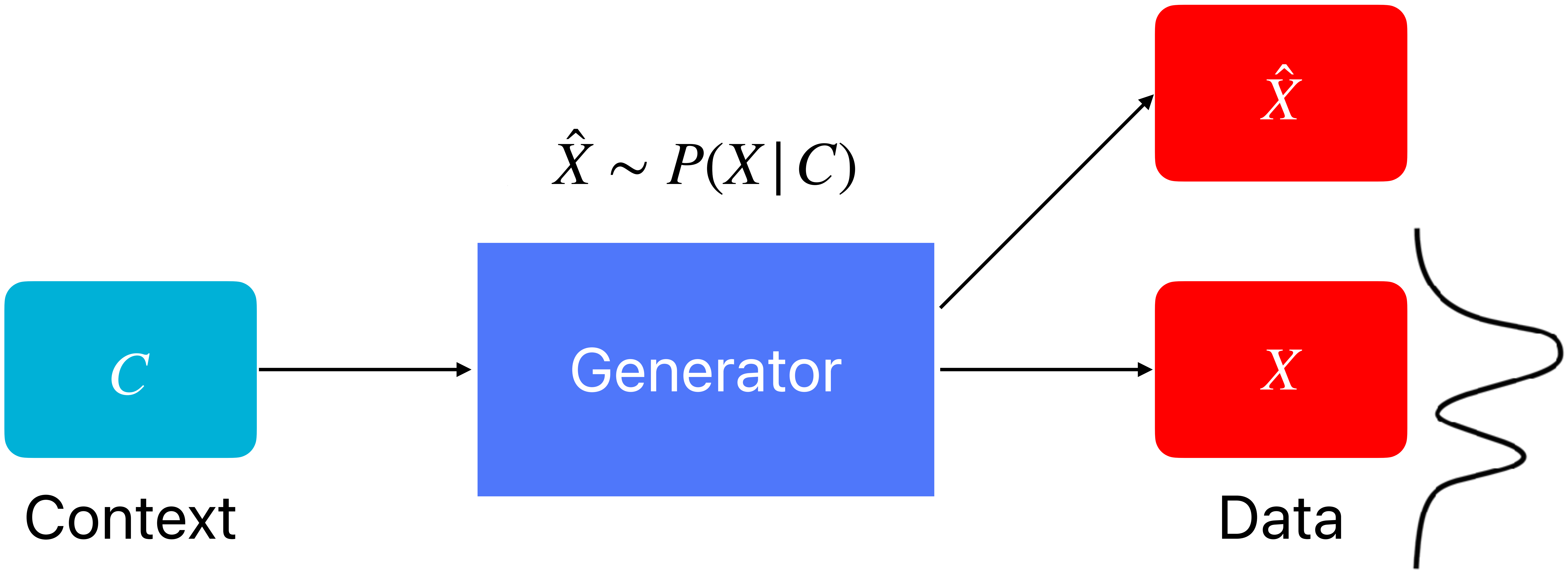
Prior to manipulating objects, a robot can execute a generative model as a simulator to anticipate potential outcomes.

The prediction must be detailed, adhere to physical laws, and be generated in real-time.

However, existing generative models cannot reach this bar.

What are the Challenges?

Formulation



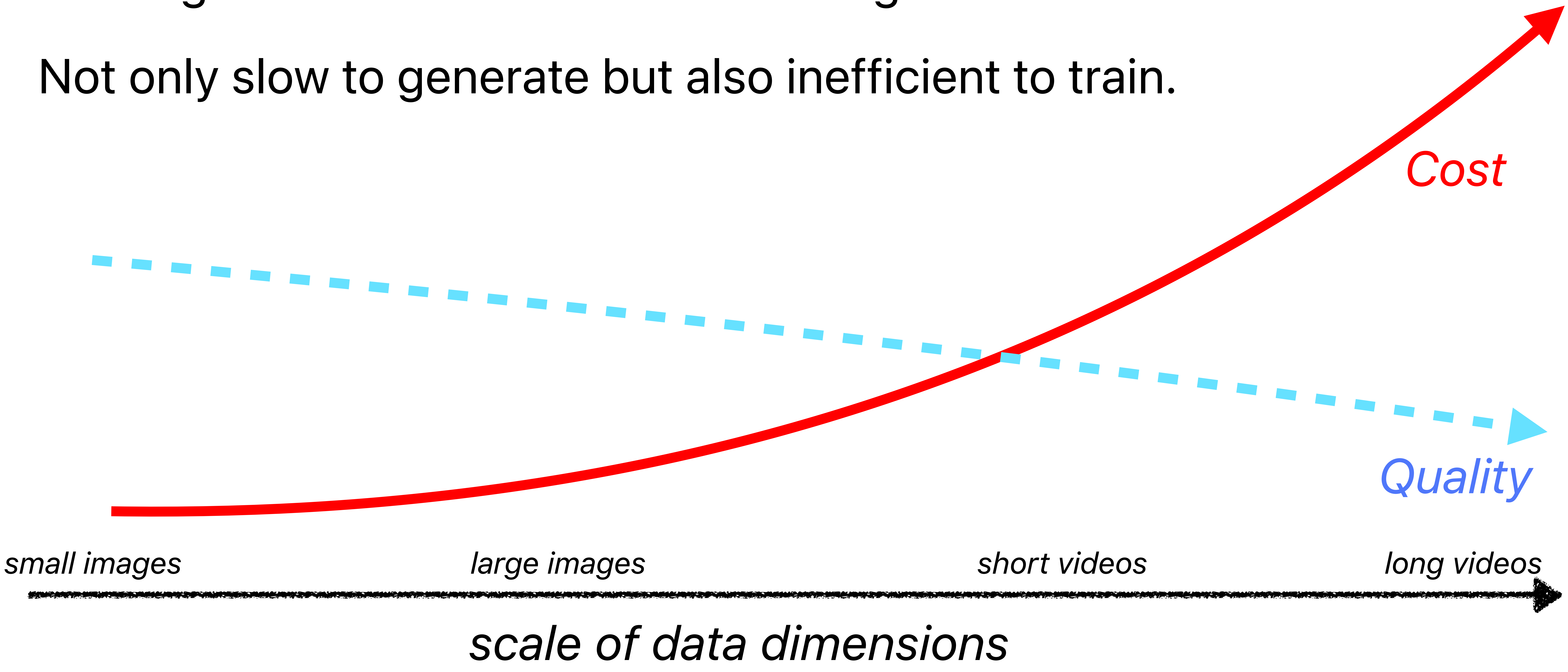
Class labels
Text prompts
Source images
Camera poses
...

Text
Graph
Image/Video
...

Not Scale Well

Existing models do not scale well for high-dimension data:

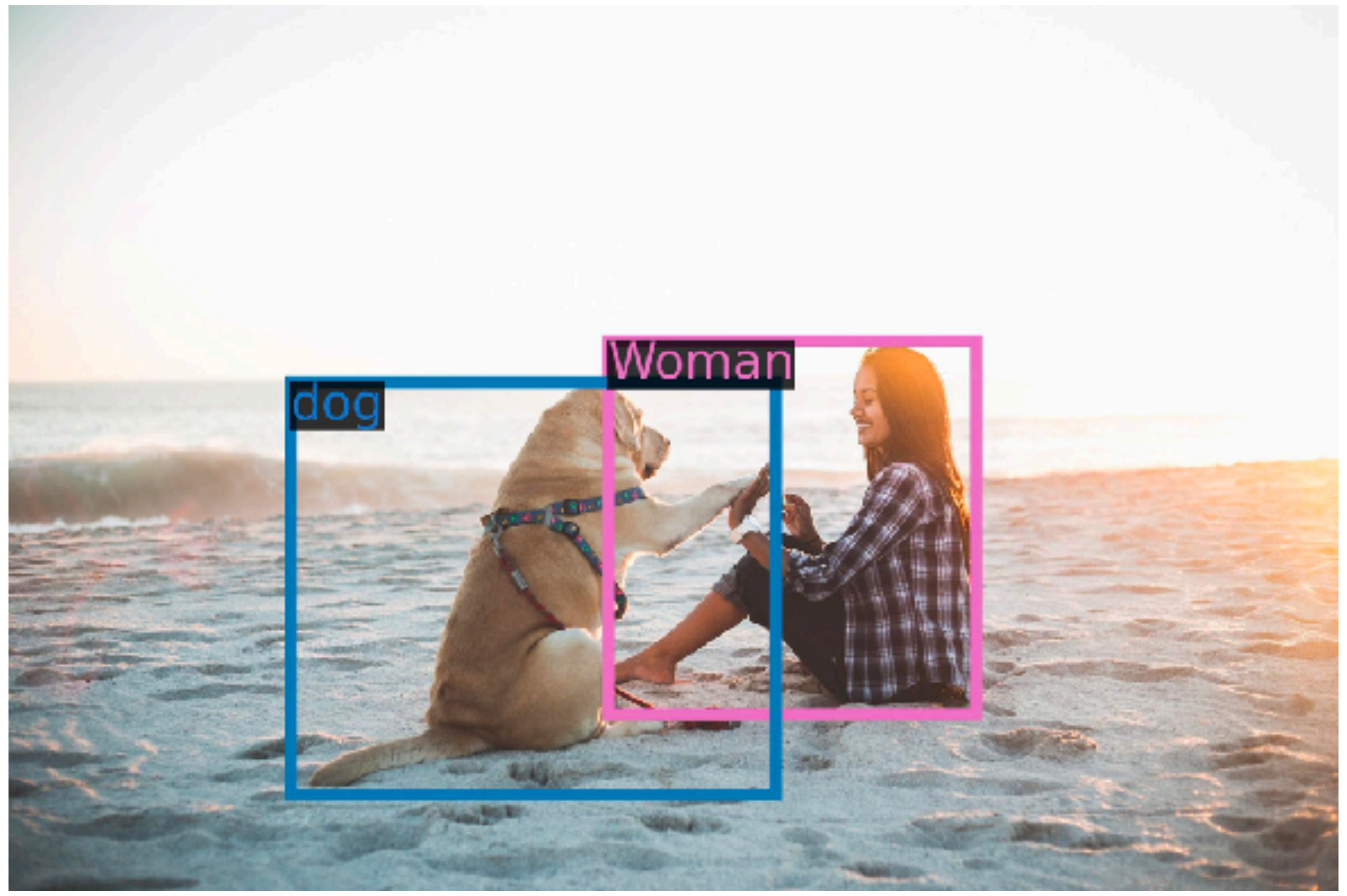
Not only slow to generate but also inefficient to train.



Not Scale Well

High-dimensional data contains useful structures that can greatly improve scalability. However, they are not studied adequately.

Hierarchical structures



Semantic structures

Research Goal

Build Future Generative Models

Scalable

- *Efficient Learning on High-dim data*

We live in a 3D world.

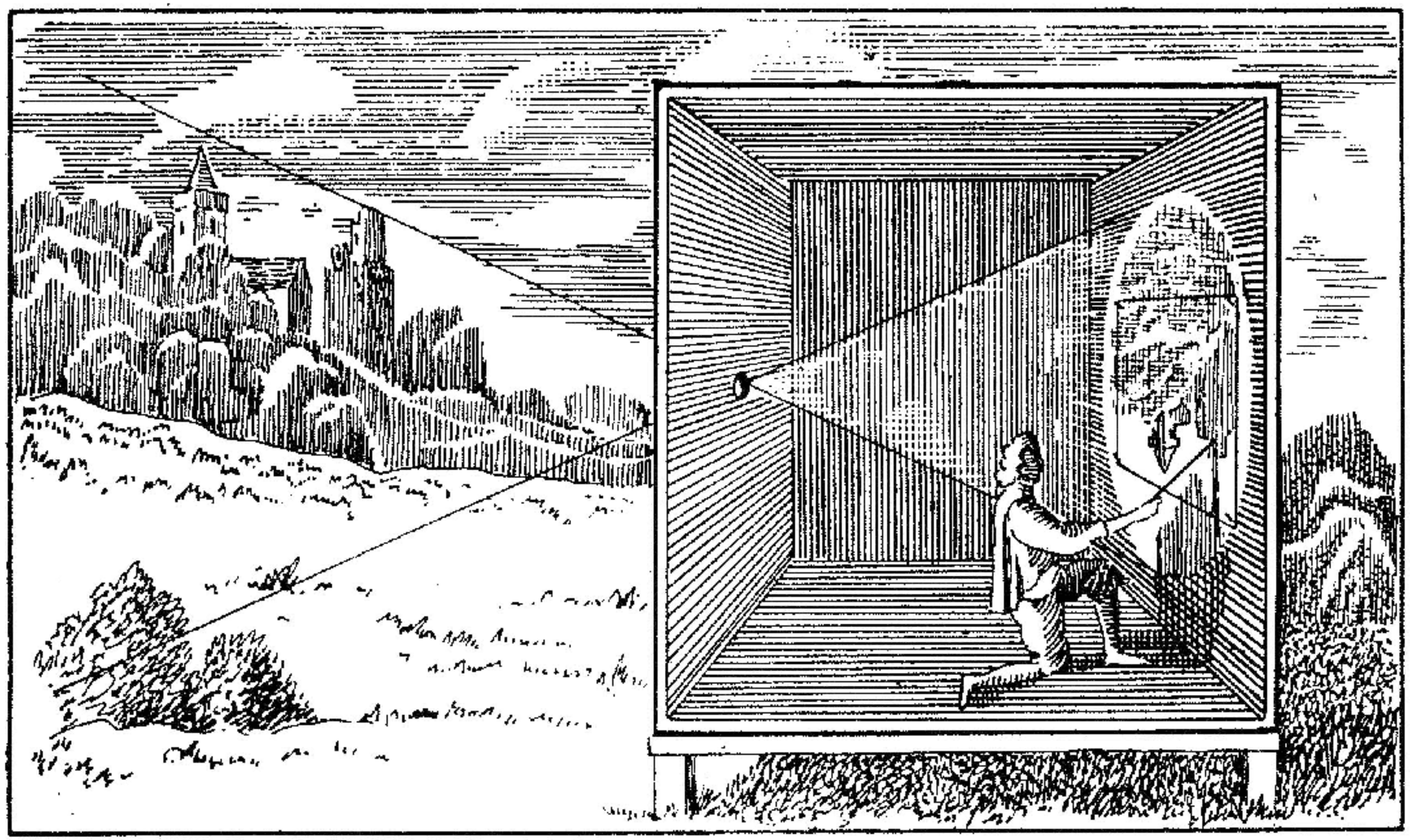
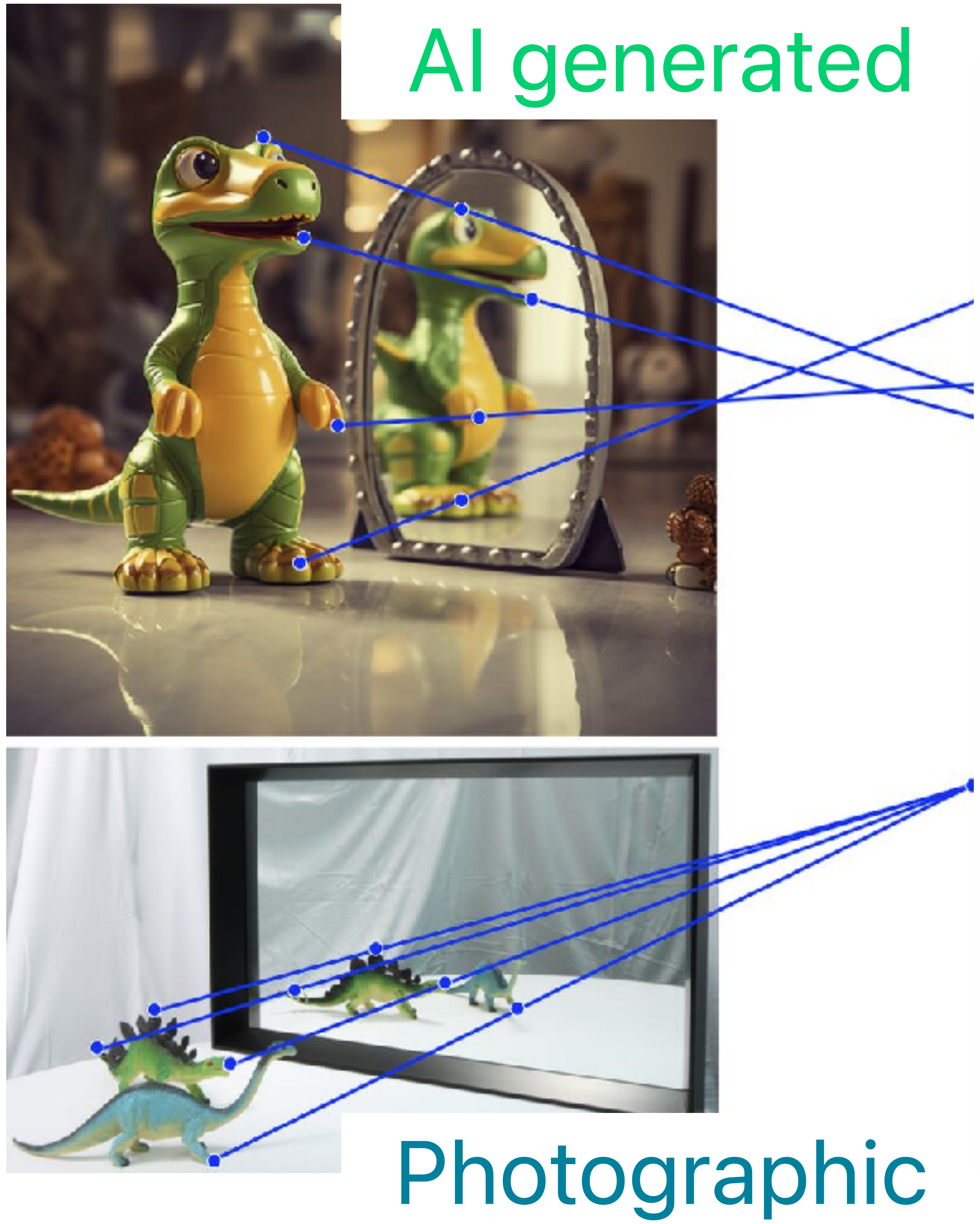


Image and video are 2D representations of a **3D world**.

No World Knowledge

Existing models ignore the underlying world knowledge, e.g., 3D projective geometry.



Research Goal

Build Future Generative Models

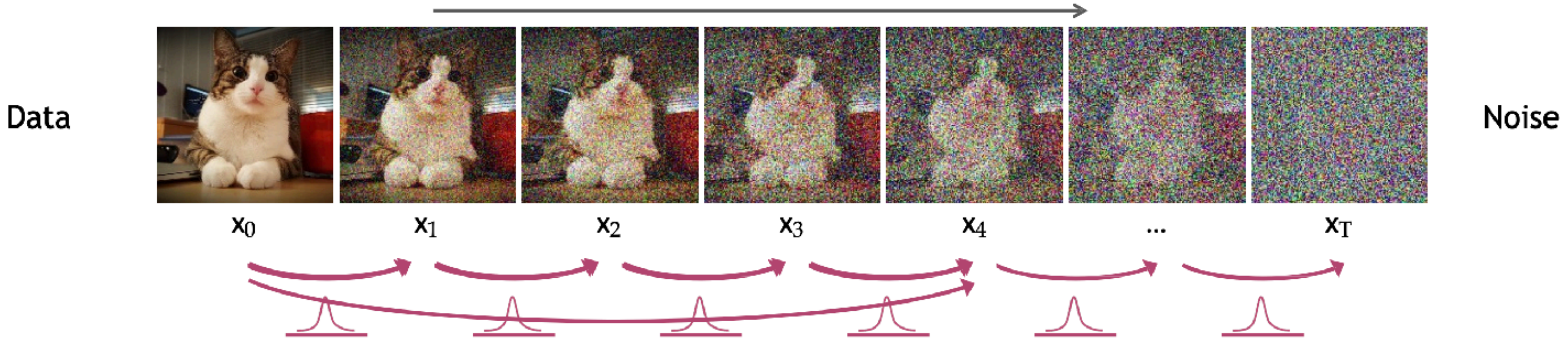
Scalable

Knowledgeable

- *Better generalization with world knowledge*

Brief Introduction of Diffusion Models

Forward Diffusion Process



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \underbrace{\sqrt{1-\beta_t}\mathbf{x}_{t-1}}_{\text{mean}}, \underbrace{\beta_t\mathbf{I}}_{\text{variance}})$$

➡ Sample: $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}$

where, $\epsilon_{t-1} \sim \mathcal{N}(0, \mathbf{I})$

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 $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Brief Introduction of Diffusion Models

Generative Learning by Denoising

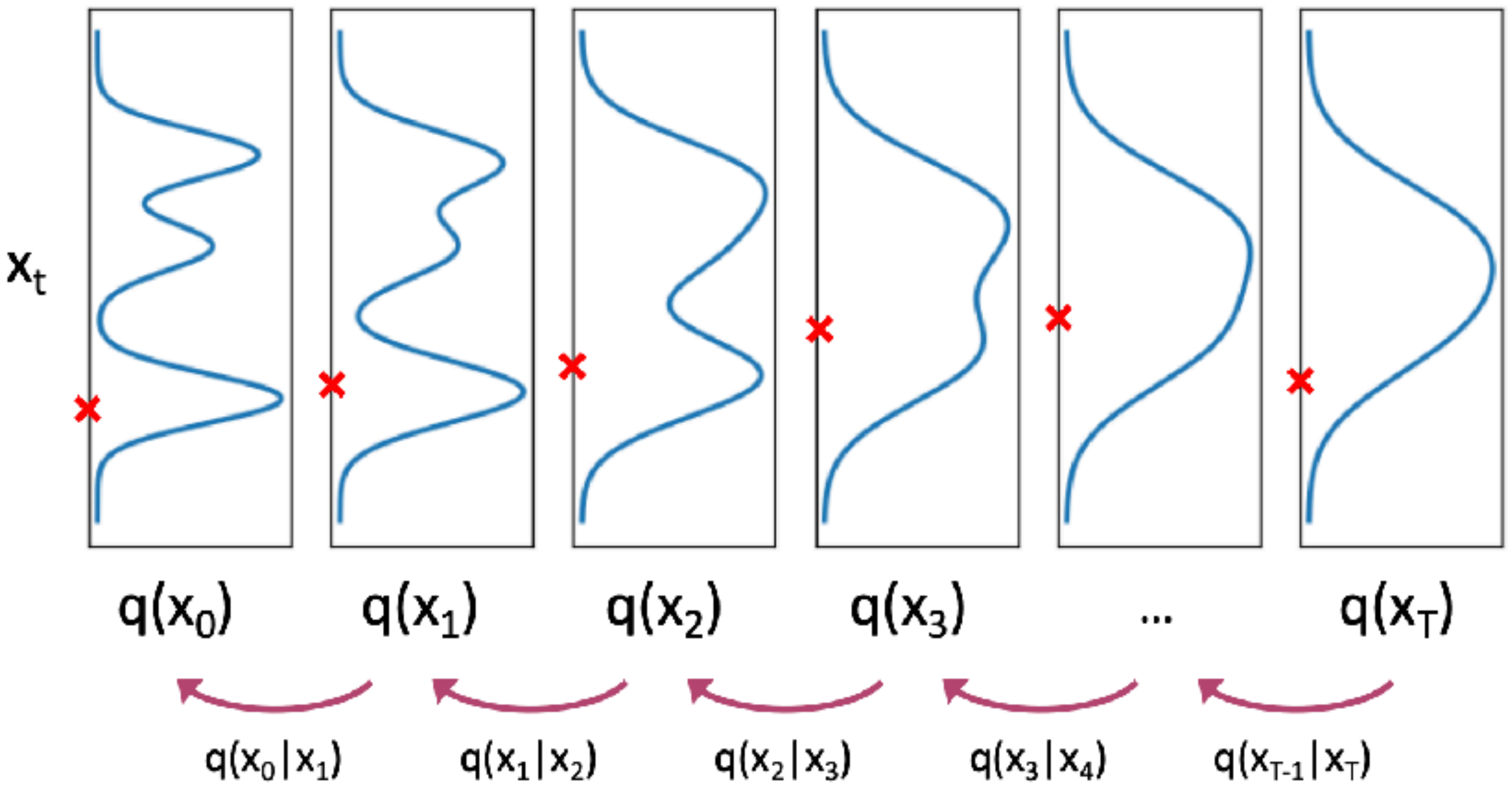
Recall, that the diffusion parameters are designed such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

Generation:

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

Iteratively sample $\mathbf{x}_{t-1} \sim \underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{True Denoising Dist.}}$

Diffused Data Distributions



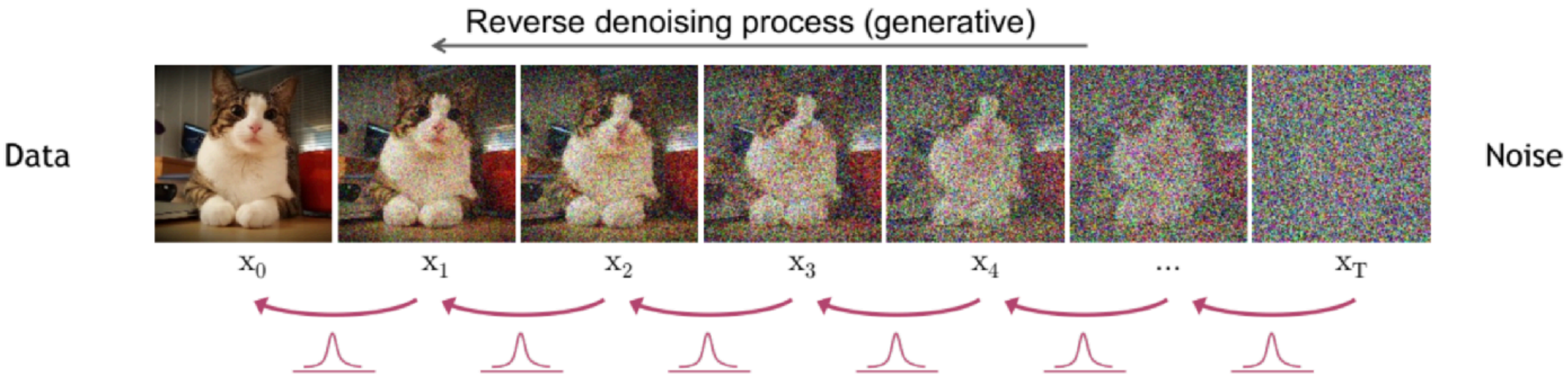
In general, $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \propto q(\mathbf{x}_{t-1})q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is intractable.

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

Brief Introduction of Diffusion Models

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



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

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\substack{\text{Trainable network} \\ \text{(U-net, Denoising Autoencoder)}}}, \sigma_t^2\mathbf{I})$$

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

Brief Introduction of Diffusion Models

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:

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

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

Brief Introduction of Diffusion Models

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

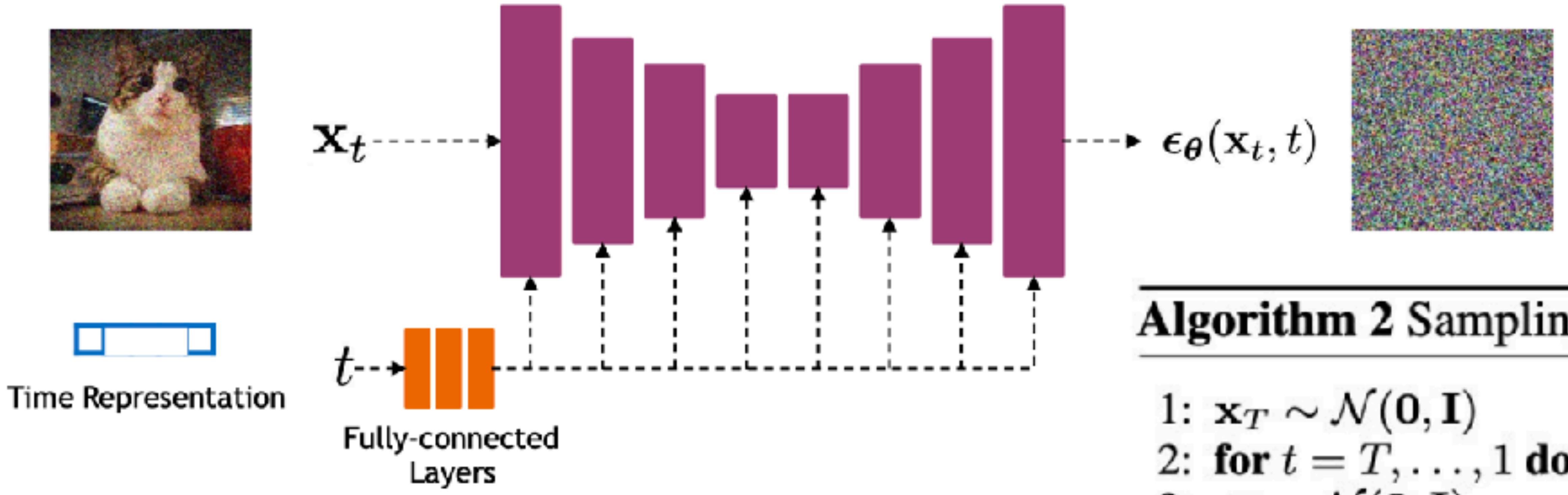
$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\ \epsilon}_{\mathbf{x}_t}, t)||^2\right]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

Brief Introduction of Diffusion Models

Reverse Diffusion Process

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

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

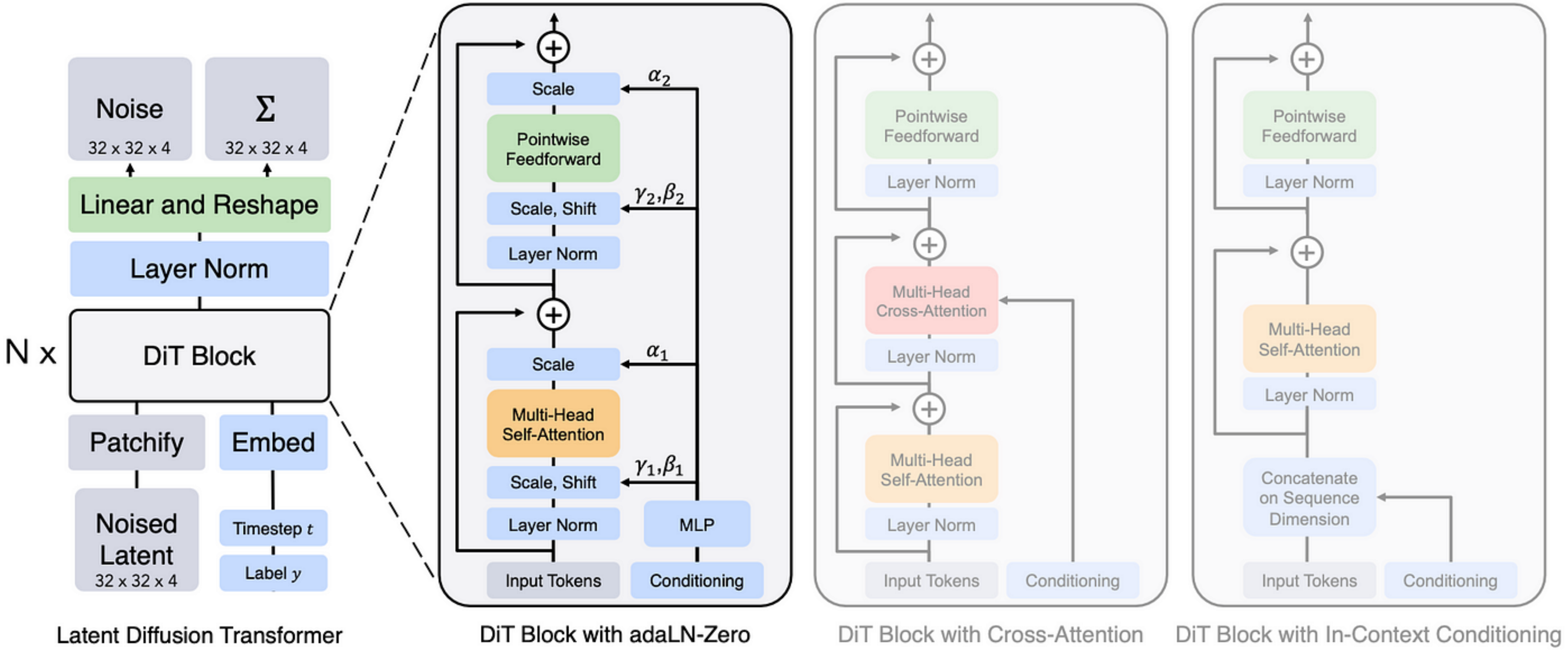


Algorithm 2 Sampling

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

Brief Introduction of Diffusion Models

Diffusion Transformers



Brief Introduction of Diffusion Models

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!

Brief Introduction of Diffusion Models

Classifier guidance

Using the gradient of a trained classifier as guidance

Applying Bayes rule to obtain conditional score function $\nabla_{x_t} log\ q_t(x_t/y)$

$$p(x \mid y) = \frac{p(y \mid x) \cdot p(x)}{p(y)}$$

$$\implies \log p(x \mid y) = \log p(y \mid x) + \log p(x) - \log p(y)$$

$$\implies \nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p(x),$$

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Classifier

Guidance scale: value >1 amplifies the influence of classifier signal.

$$p_\gamma(x \mid y) \propto p(x) \cdot p(y \mid x)^\gamma.$$

Brief Introduction of Diffusion Models

Classifier guidance

Using the gradient of a trained classifier as guidance

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Samples from an unconditional diffusion model with classifier guidance, for guidance scales 1.0 (left) and 10.0 (right), taken from Dhariwal & Nichol (2021).

Brief Introduction of Diffusion Models

Classifier-free guidance

Get guidance by Bayes' rule on conditional diffusion models

$$p(y \mid x) = \frac{p(x \mid y) \cdot p(y)}{p(x)}$$

$$\Longrightarrow \log p(y \mid x) = \log p(x \mid y) + \log p(y) - \log p(x)$$

$$\Longrightarrow \boxed{\nabla_x \log p(y \mid x) = \nabla_x \log p(x \mid y) - \nabla_x \log p(x).}$$

We proved this in classifier guidance.

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \left(\nabla_x \log p(x \mid y) - \nabla_x \log p(x)\right),$$

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusion model

Brief Introduction of Diffusion Models

Classifier-free guidance

Get guidance by Bayes' rule on conditional diffusion models

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusio model

This is a barycentric combination of the conditional and the unconditional score function. For $\gamma = 0$, we recover the unconditional model, and for $\gamma = 1$ we get the standard conditional model. But $\gamma > 1$ is where the magic happens. Below are some examples from OpenAI's GLIDE model[8], obtained using classifier-free guidance.

Two sets of samples from OpenAI's GLIDE model, for the prompt *'A stained glass window of a panda eating bamboo.'*, taken from their paper. Guidance scale 1 (no guidance) on the left, guidance scale 3 on the right.

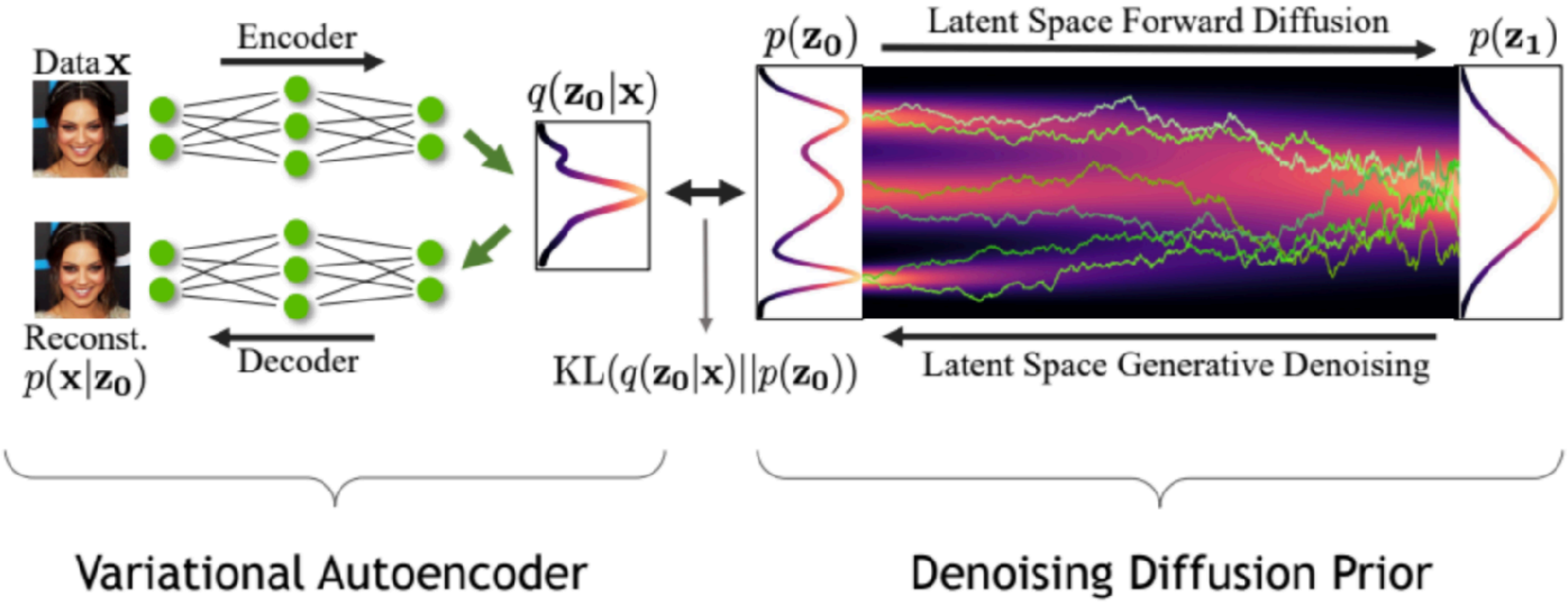
In practice

$$\hat{\epsilon} = (1+\omega)\epsilon_\theta(x_t, y) - \omega\epsilon_\theta(x_t)$$

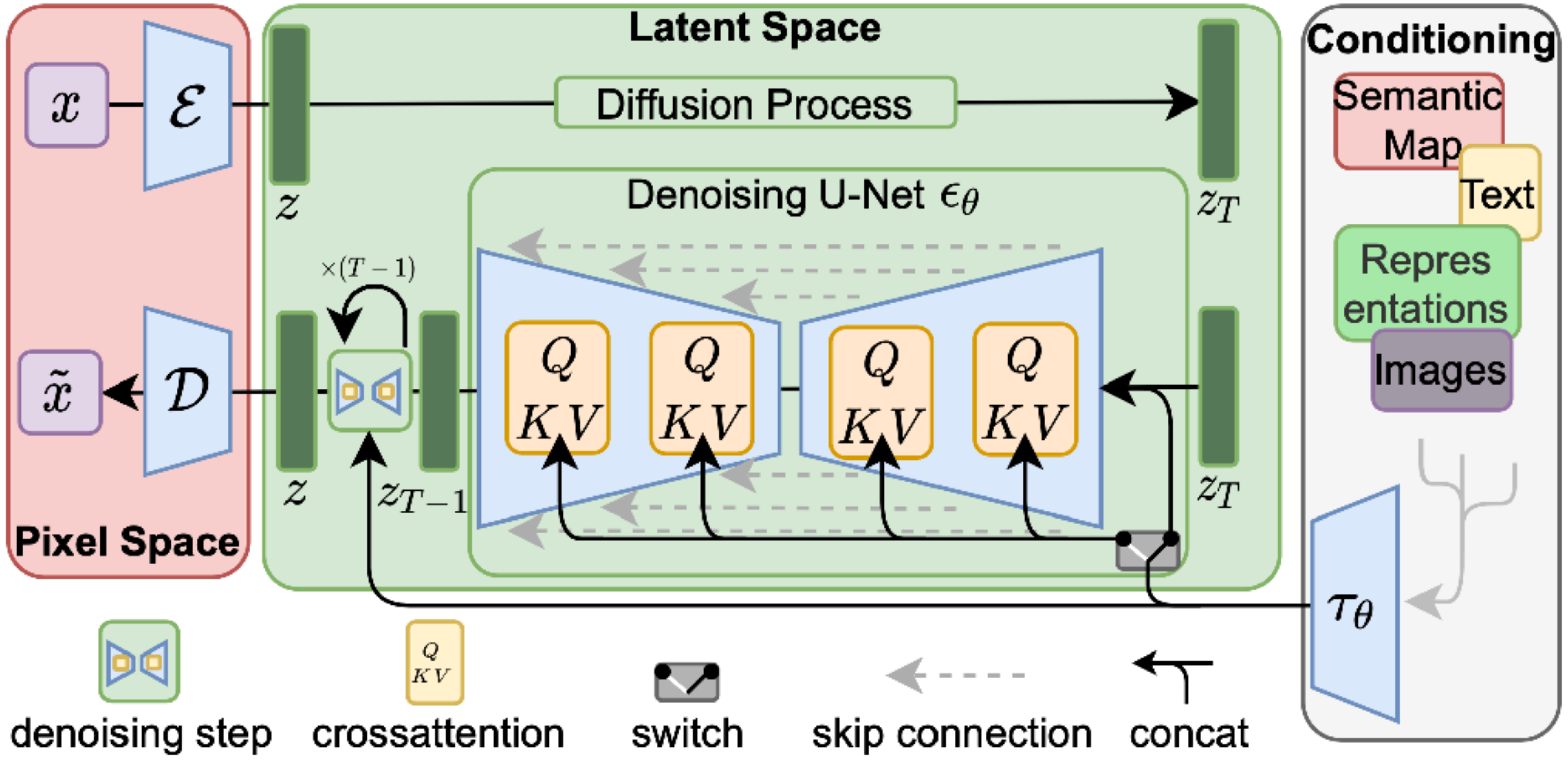
Brief Introduction of Diffusion Models

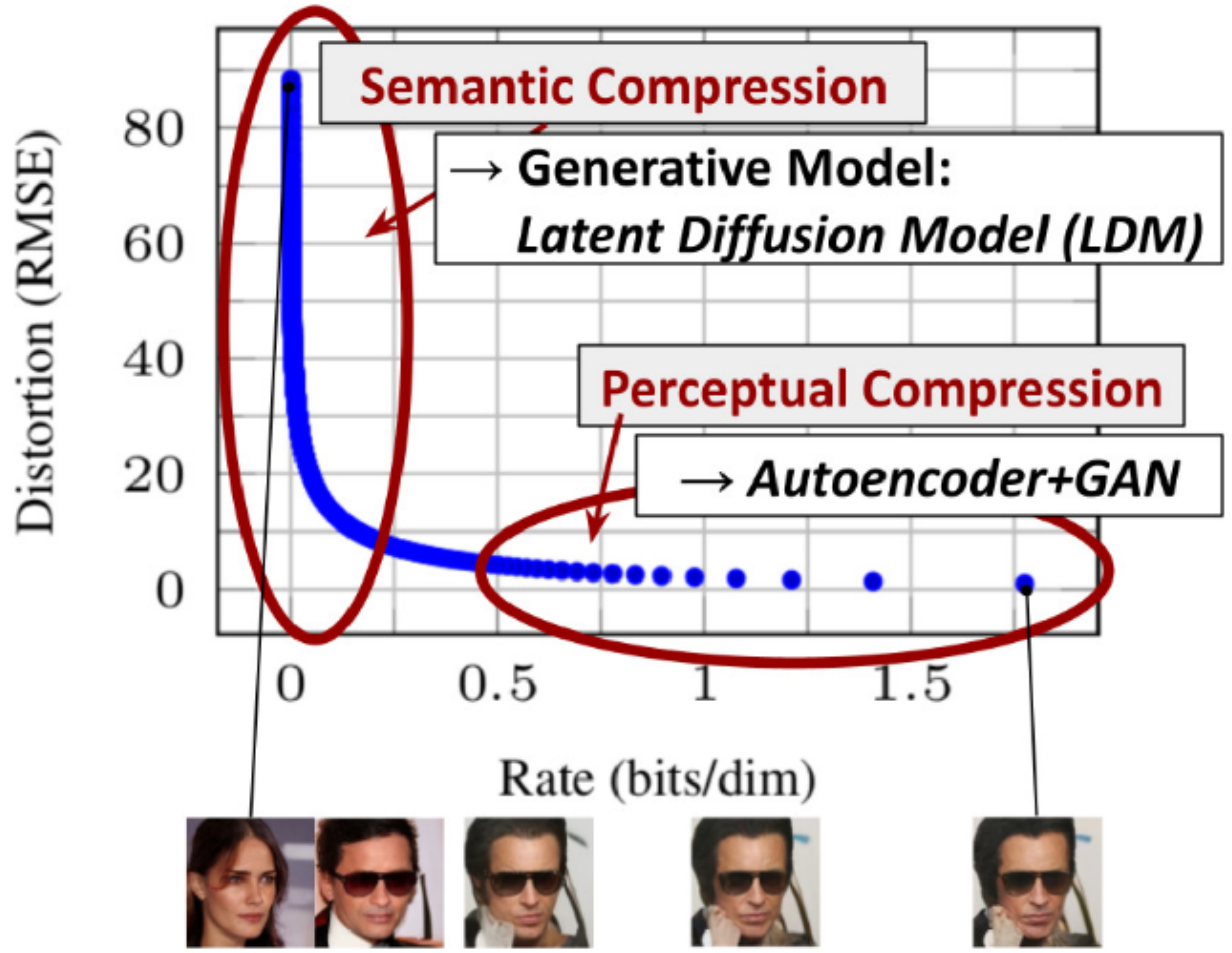
Latent-space diffusion models

Variational autoencoder + score-based prior



Brief Introduction of Diffusion Models





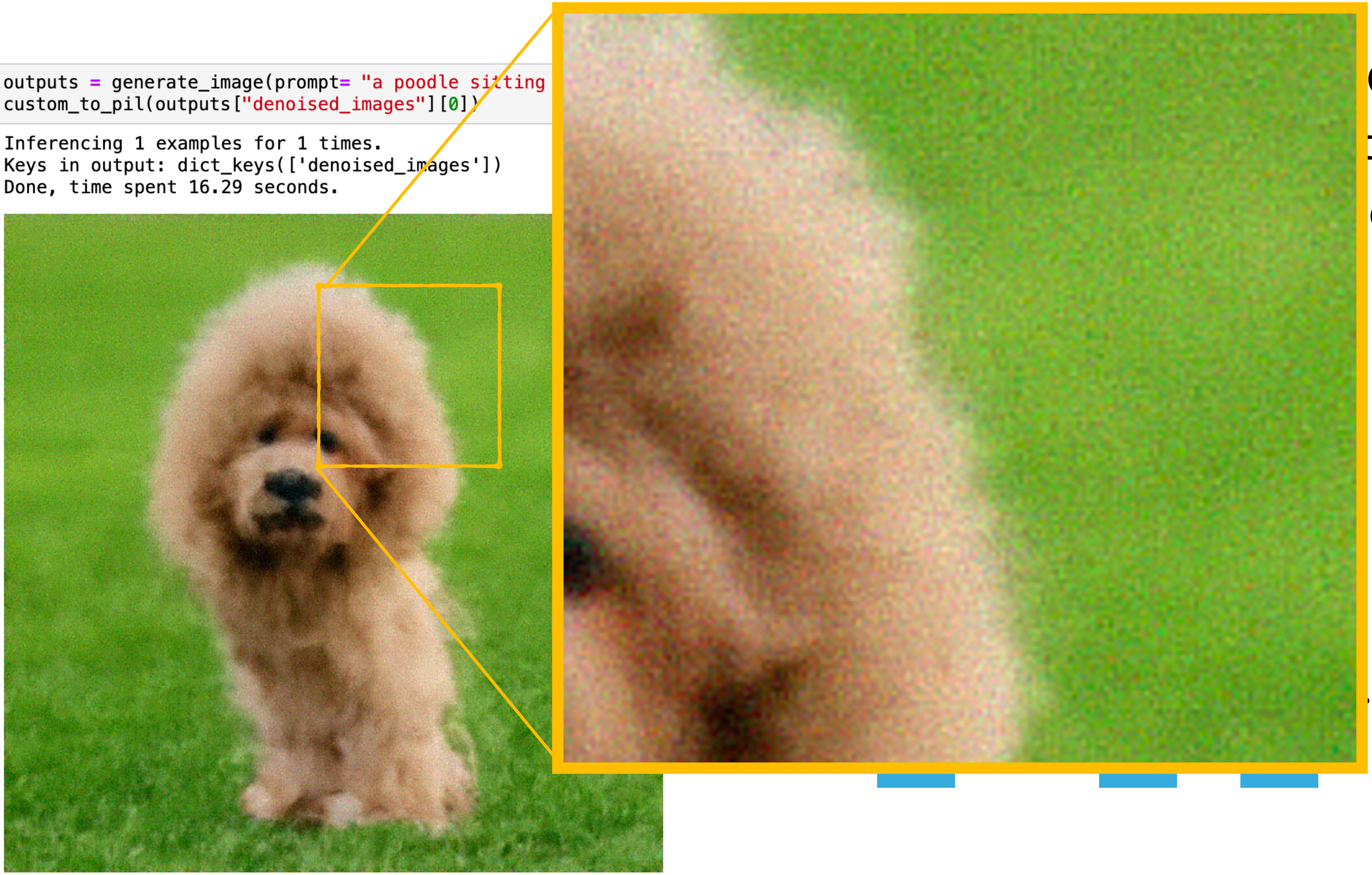
Brief Introduction of Diffusion Models

Scalable

Scaling to High-dimensional Data

```
outputs = generate_image(prompt= "a poodle sitting
custom_to_pil(outputs["denoised_images"][0])

Inferencing 1 examples for 1 times.
Keys in output: dict_keys(['denoised_images'])
Done, time spent 16.29 seconds.
```

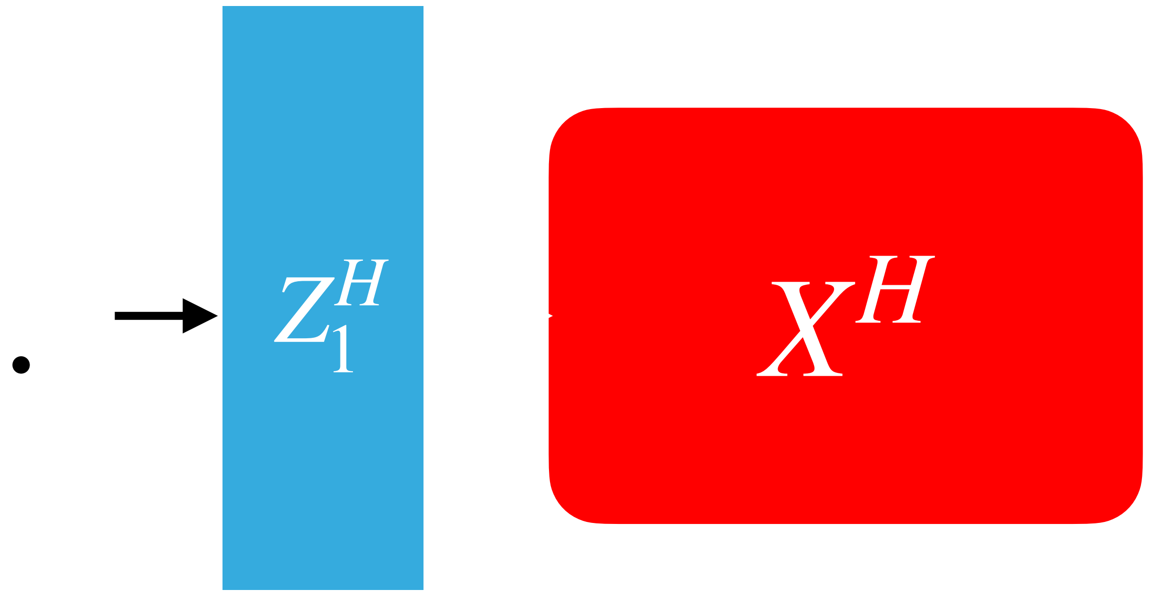


512x512

on is difficult:

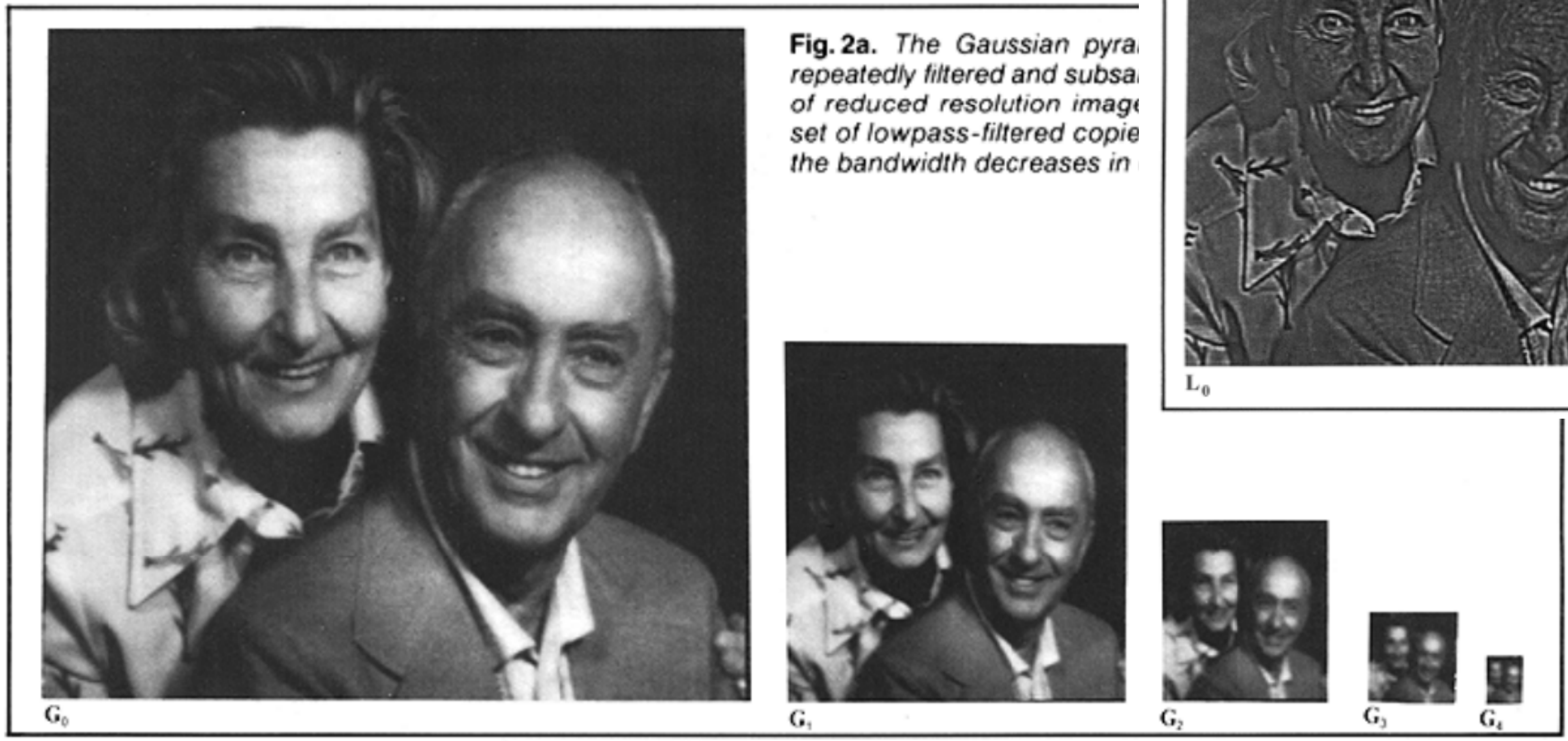
quality.

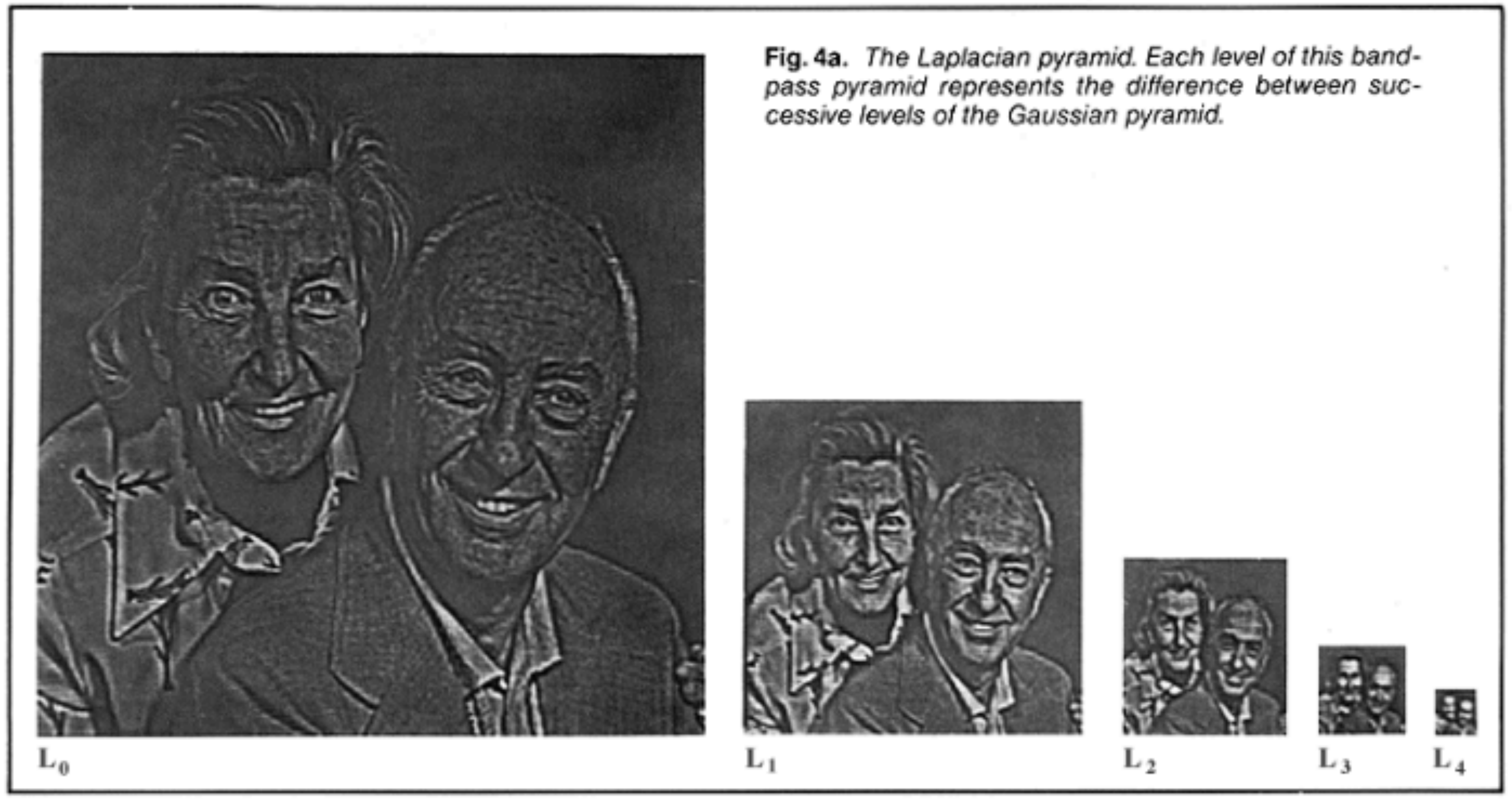
ciency.



Data

Pyramid Representations

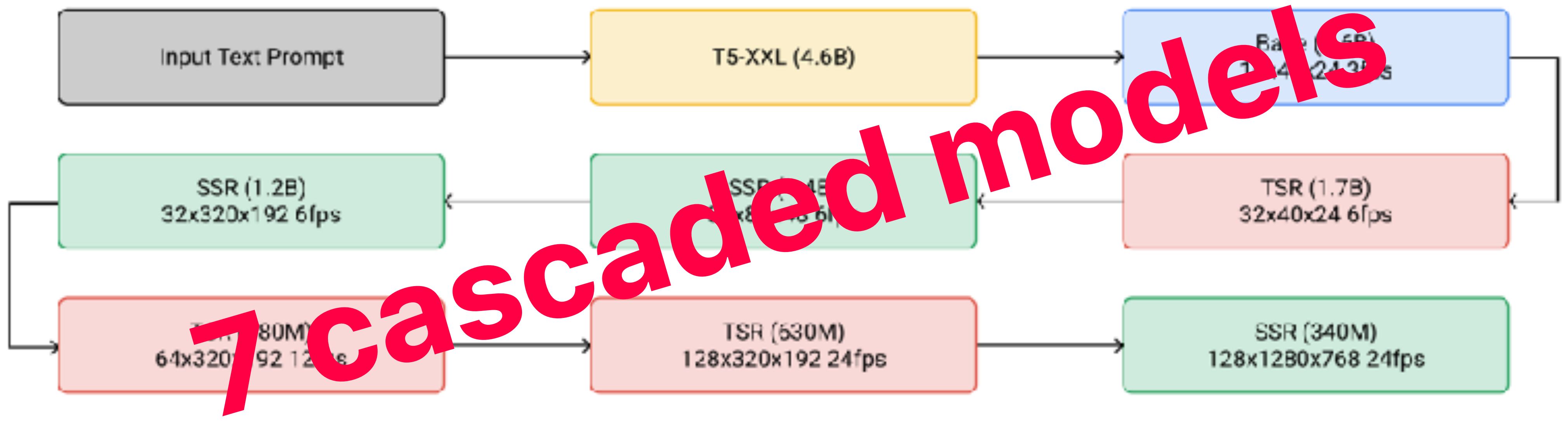


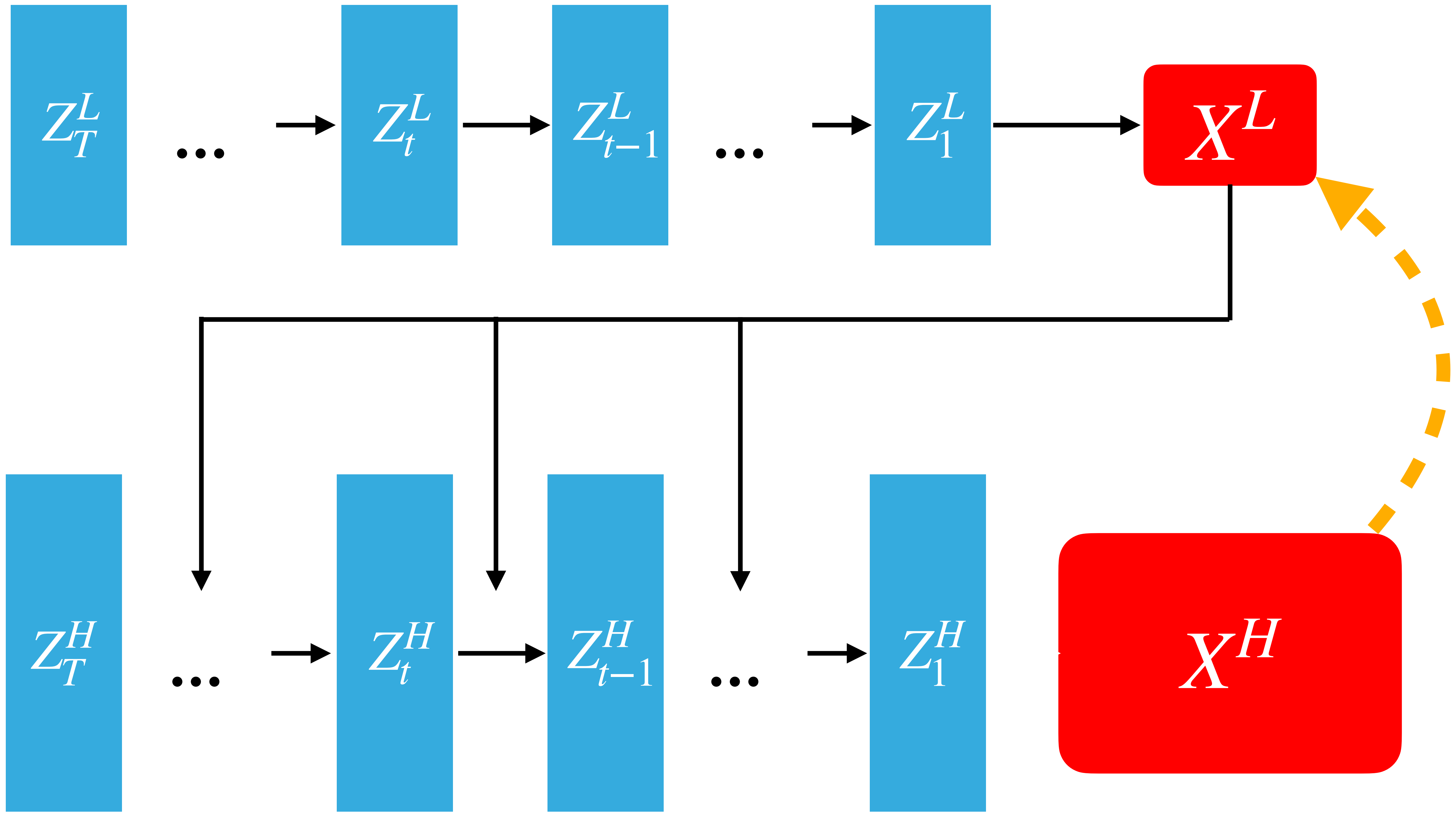


E.H. Andelson and C.H. Anderson and J.R. Bergen and P.J. Burt and J.M. Ogden.
“Pyramid methods in image processing". 1984.

Cascaded Diffusion Models

(1) Slow inference process



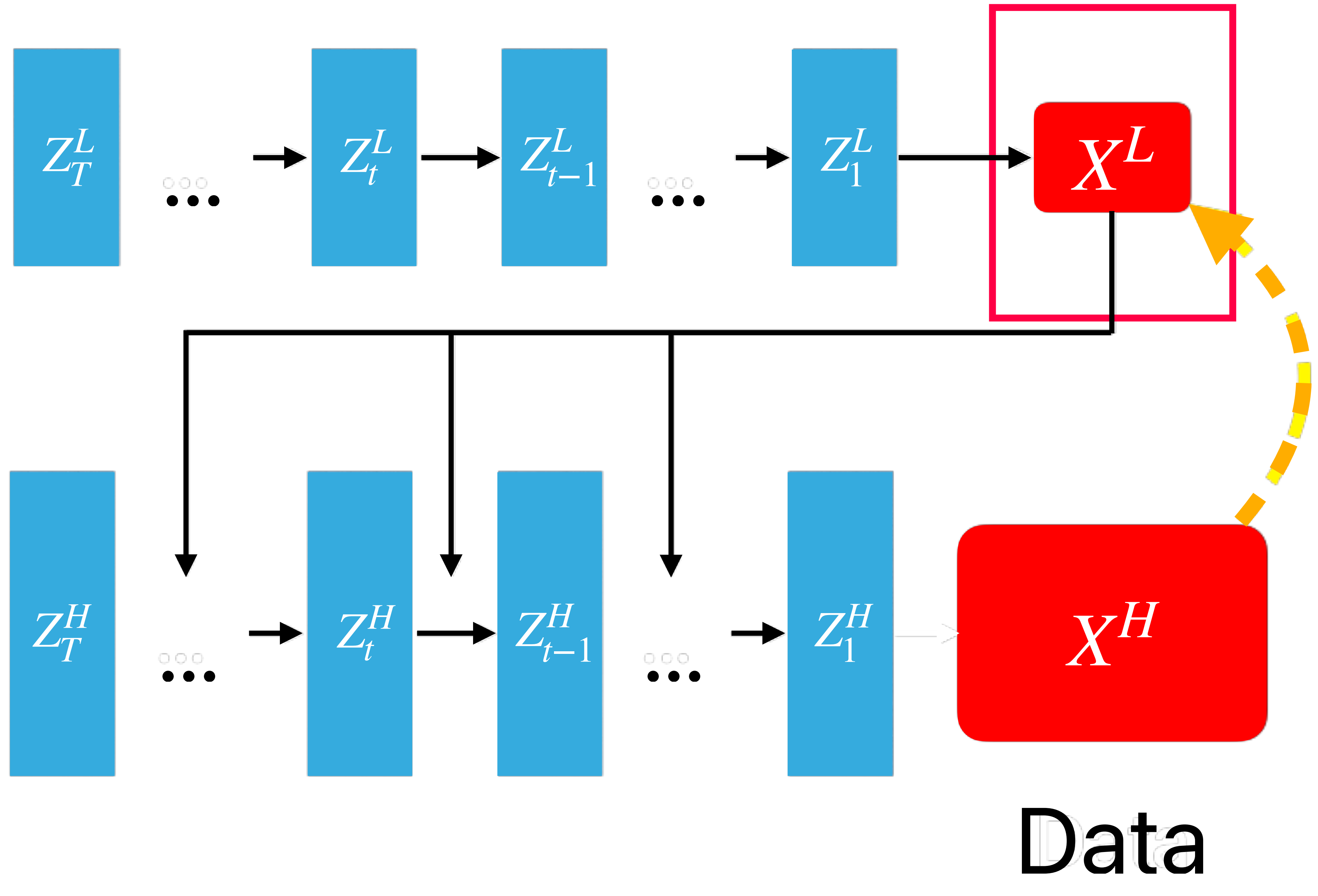


Data

Ho, Jonathan, et al. "Cascaded diffusion models for high fidelity image generation." The Journal of Machine Learning Research 23.1 (2022): 2249-2281.

Cascaded Diffusion Models

Can we leverage the multi-scale information in a single generative model?



16x16 32x32 64x64 128x128 256x256

Learning process with latents

Latents

stru Incorpora

Predict these structures while sharing computations with the final prediction. This can greatly improve learning efficiency.

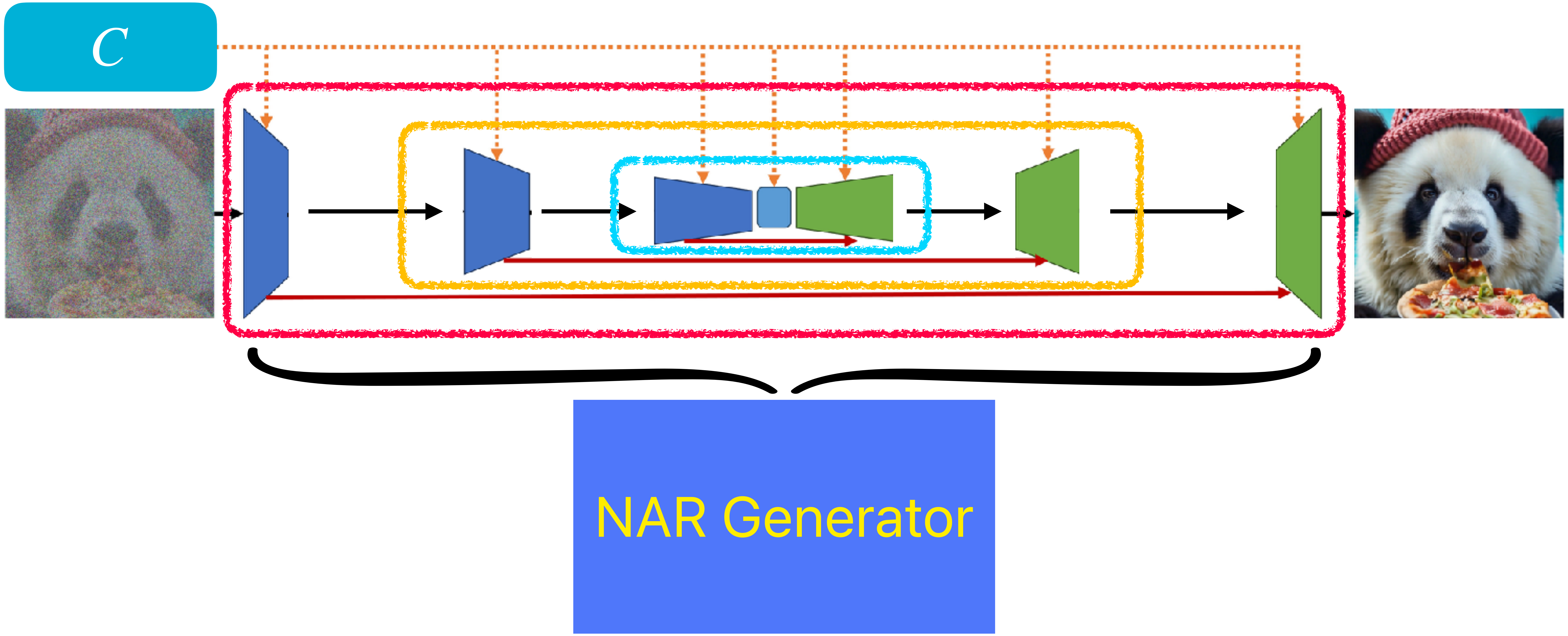
Context

Data

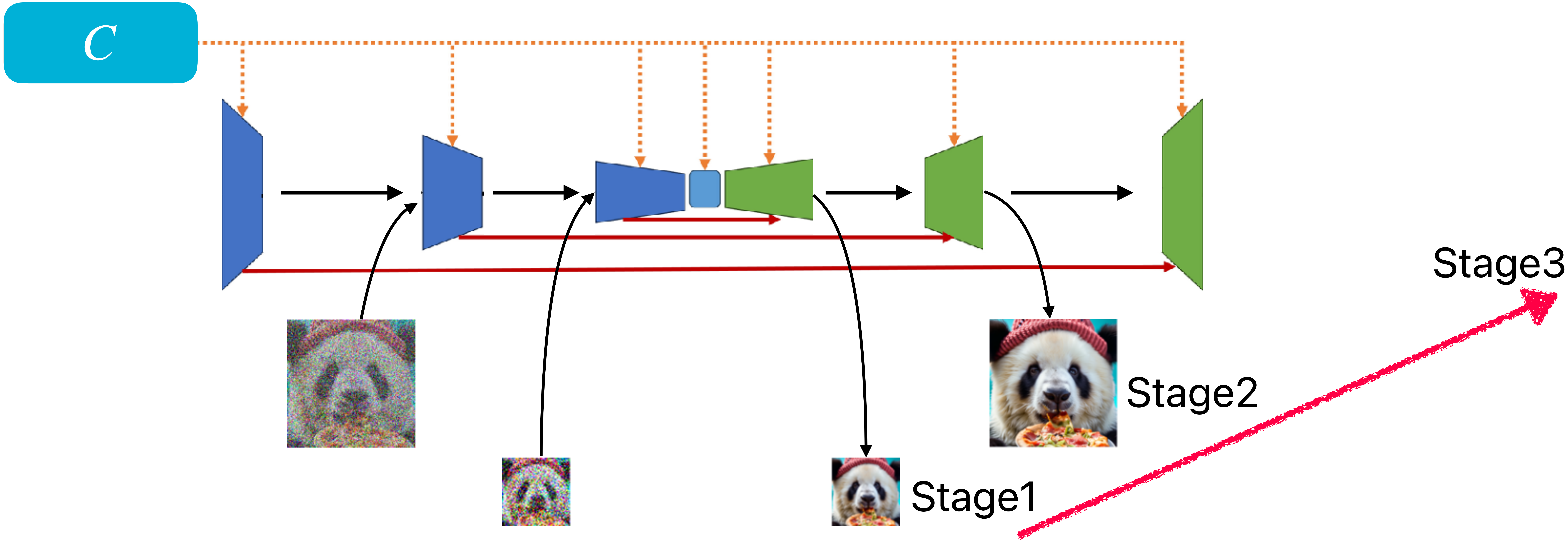
Sharing computation

Sharing Multi-scale Computations

Standard diffusion architecture contains multi-scale computation.

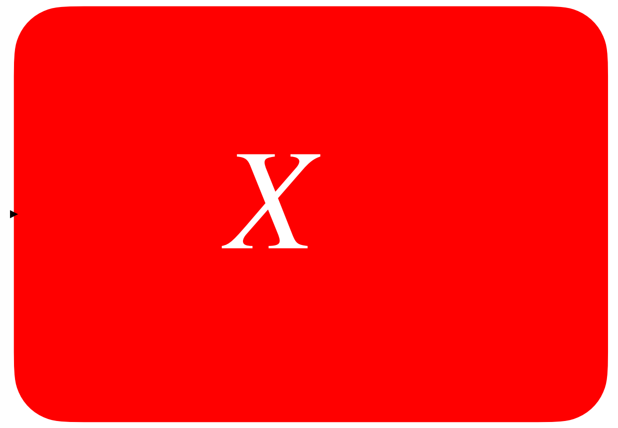


Diffusion via Transformation (f-DM)



Gu, J., Zhai, S., Zhang, Y., Bautista, M. A., & Susskind, J.,
"f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation," ICLR 2023

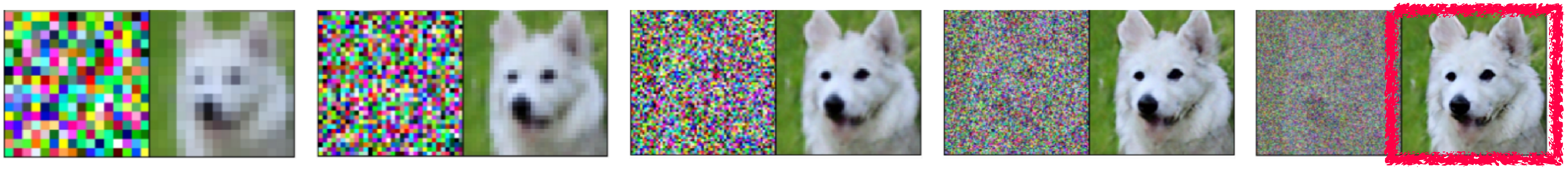
Diffusion via Transformation (f-DM)



Gu, J., Zhai, S., Zhang, Y., Bautista, M. A., & Susskind, J.,
"f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation," ICLR 2023

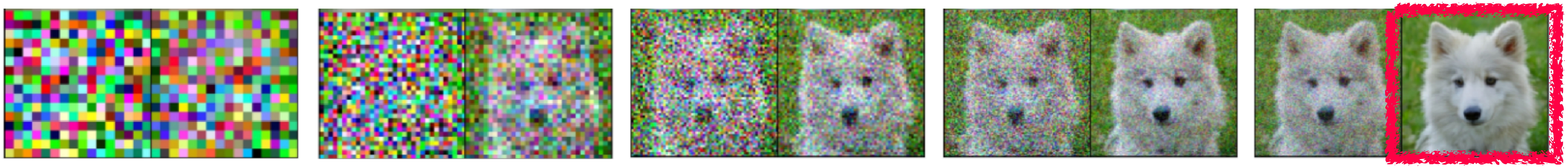
Comparison to Cascaded Models

Cascaded Diffusion



16x16 32x32 64x64 128x128 256x256

f-DM (Ours)



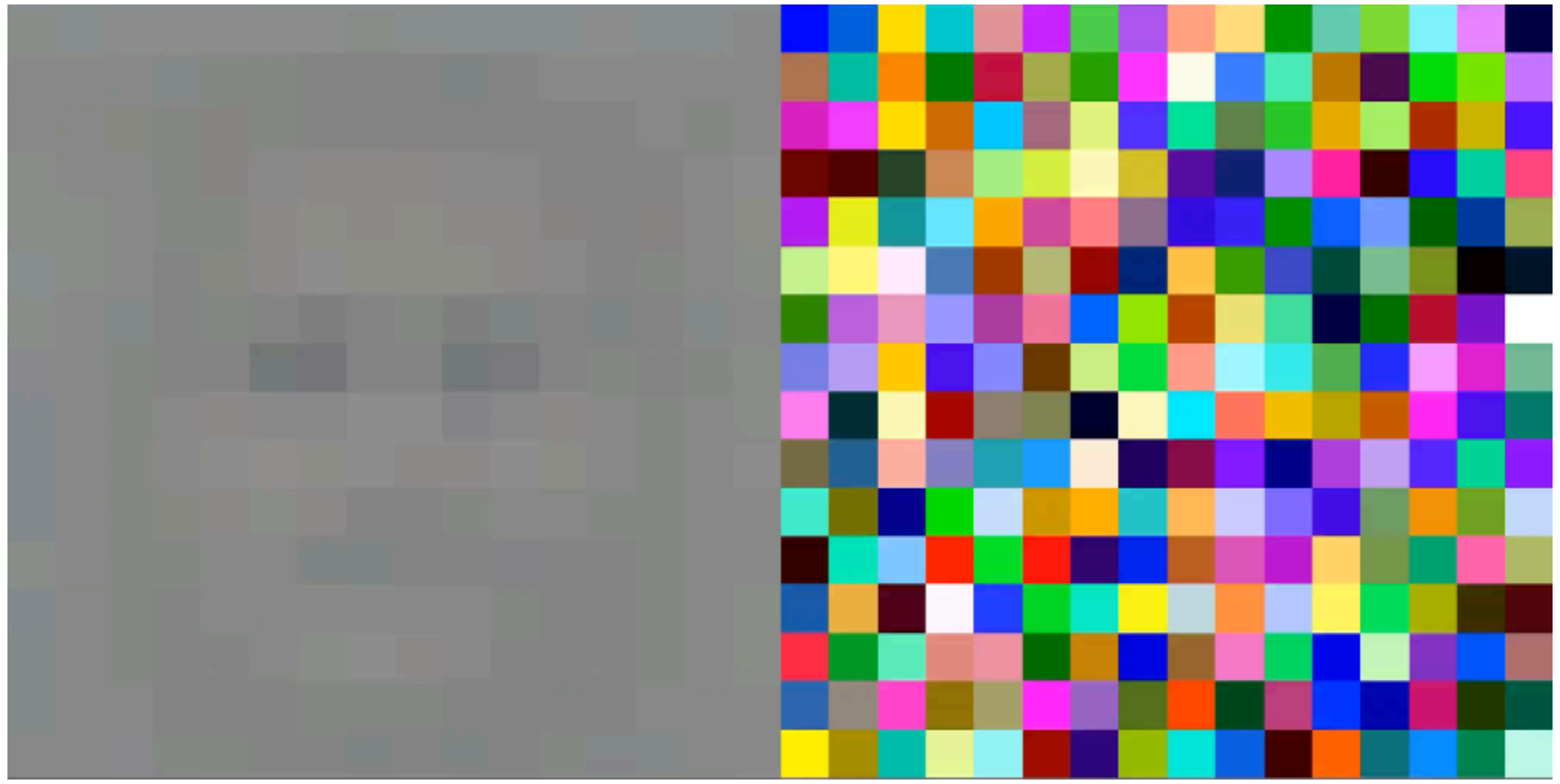
16x16 32x32 64x64 128x128 256x256

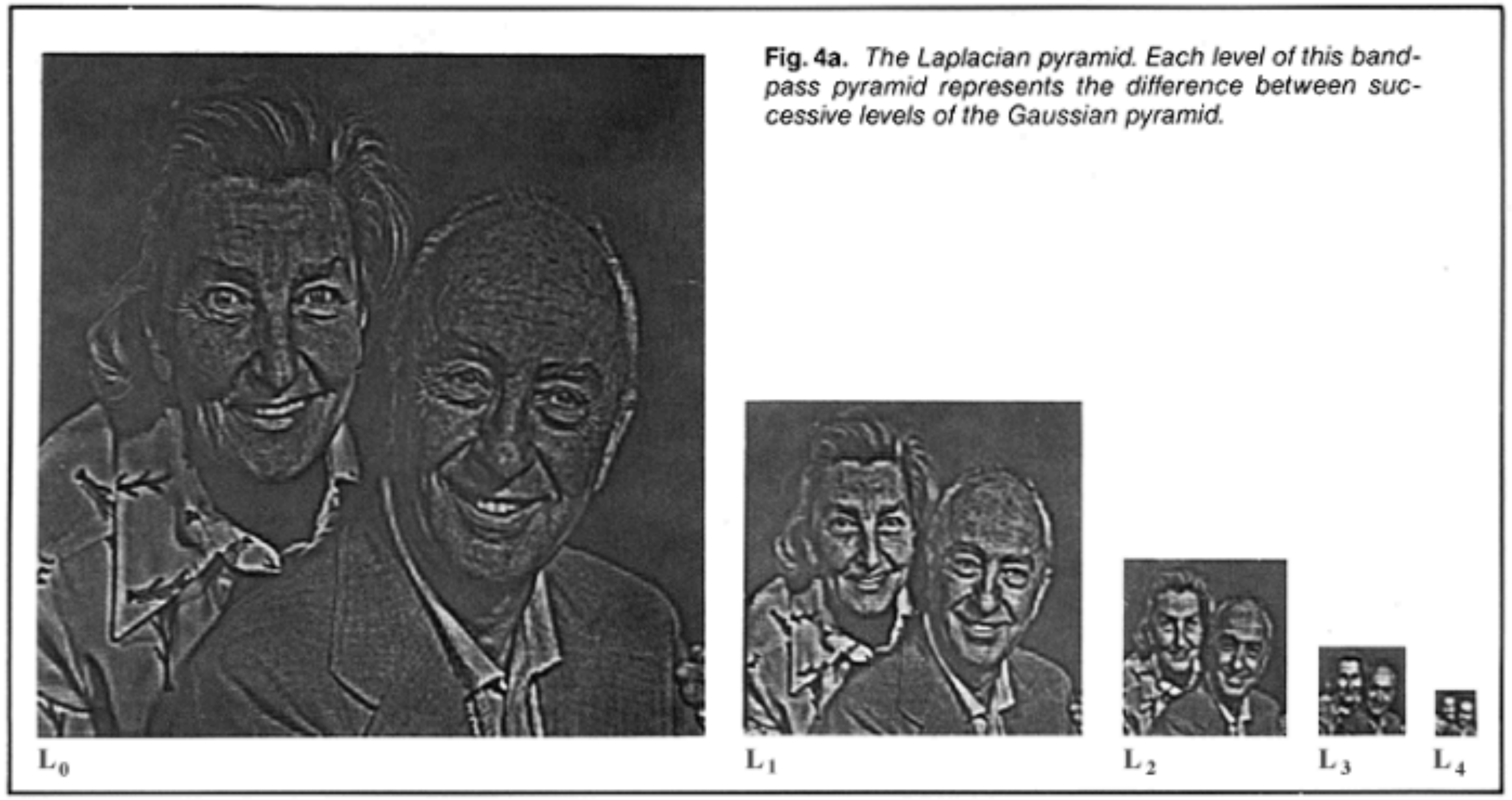
Gu, J., Zhai, S., Zhang, Y., Bautista, M. A., & Susskind, J.,
"f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation," ICLR 2023

Progress of Generation

Predicted "difference" from the target.

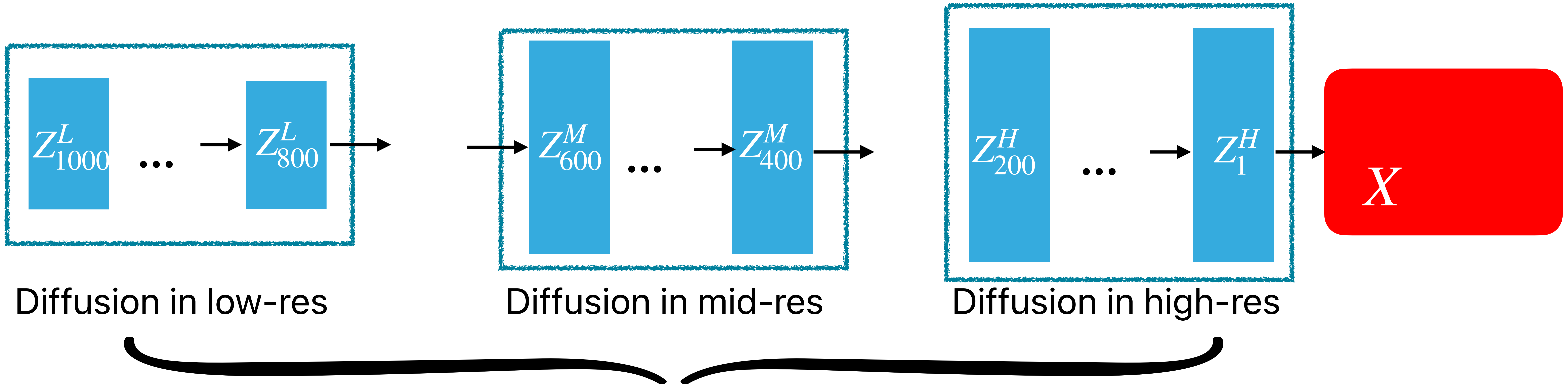
Diffusion Latents





Gu, J., Zhai, S., Zhang, Y., Bautista, M. A., & Susskind, J.,
"f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation," ICLR 2023

Potential Issues



Non-trivial to determine the best schedule for each stage

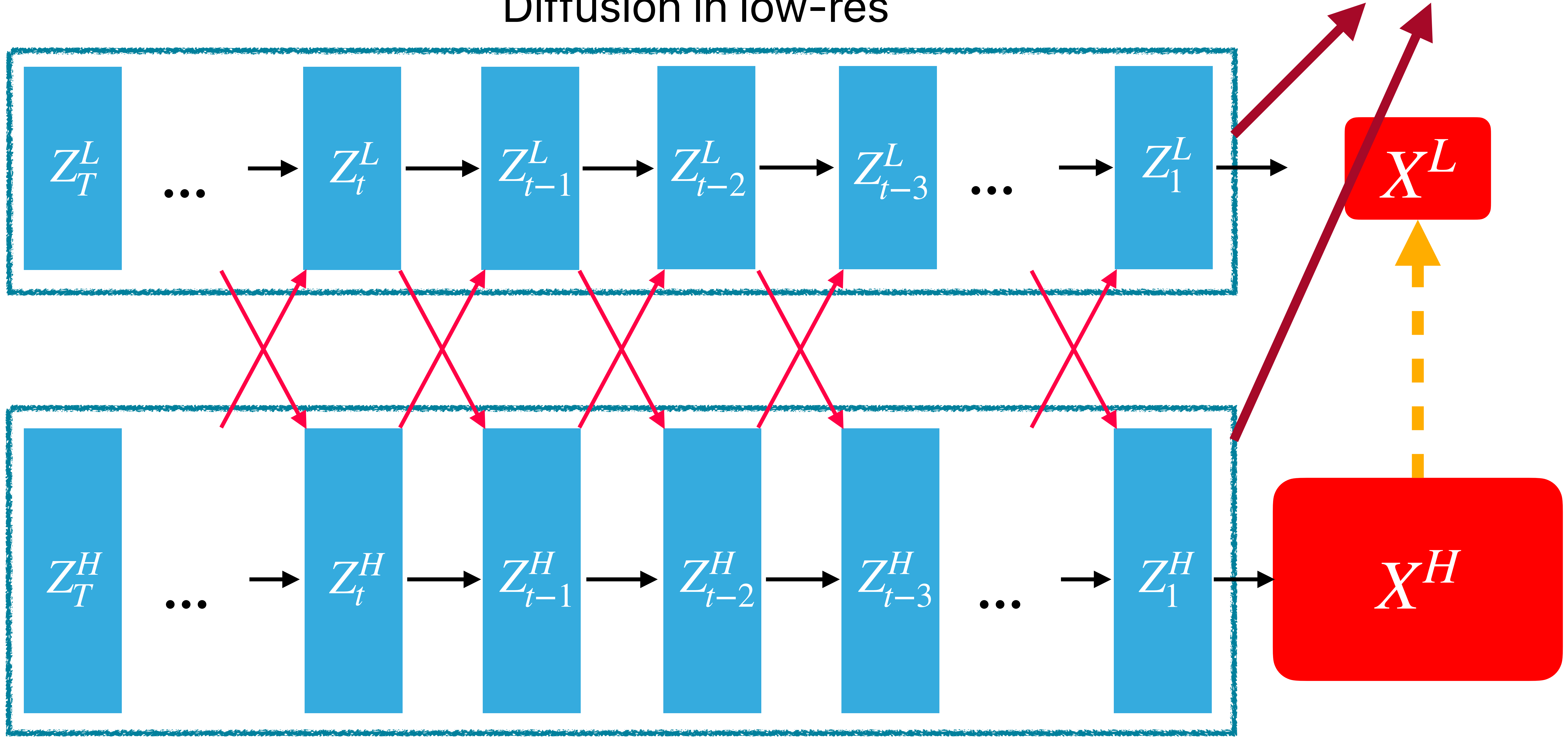
Gu, J., Zhai, S., Zhang, Y., Bautista, M. A., & Susskind, J.,
"f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation," ICLR 2023

Matryoshka Diffusion (MDM)

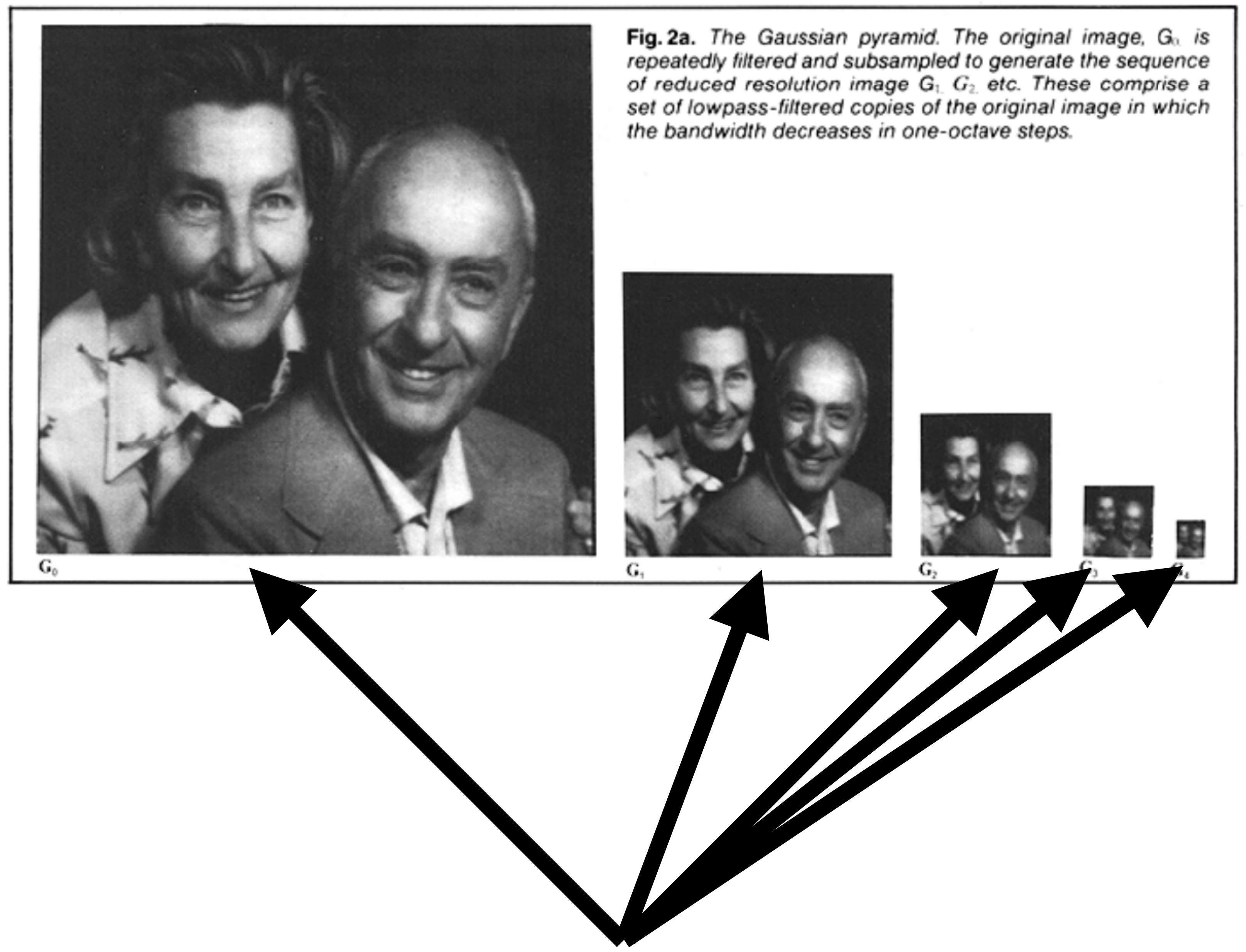
We make diffusion happen at both low and high resolutions.

NOTE: Noise schedule can be different

Diffusion in low-res



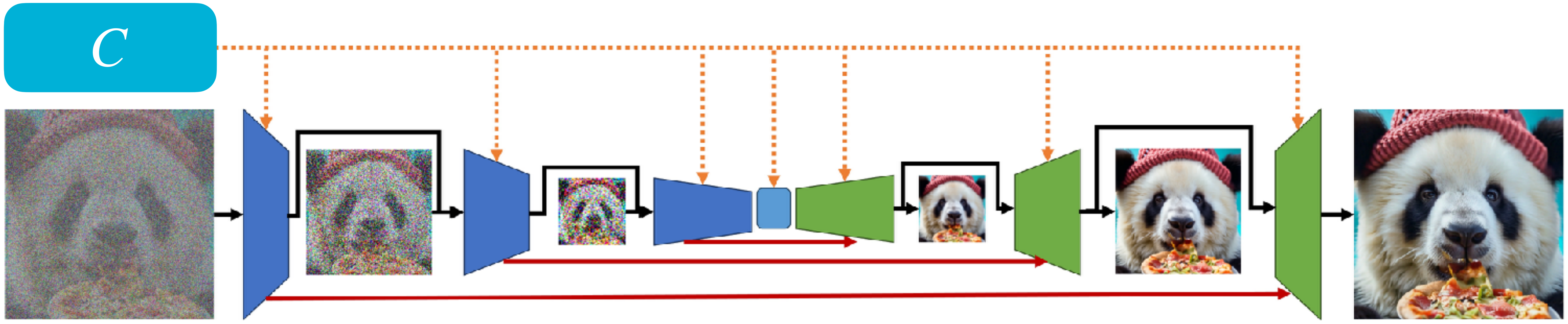
Diffusion in high-res



Noise function

Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., "Matryoshka Diffusion Models," ICLR 2024

Matryoshka Diffusion (MDM)



Multi-scale inputs

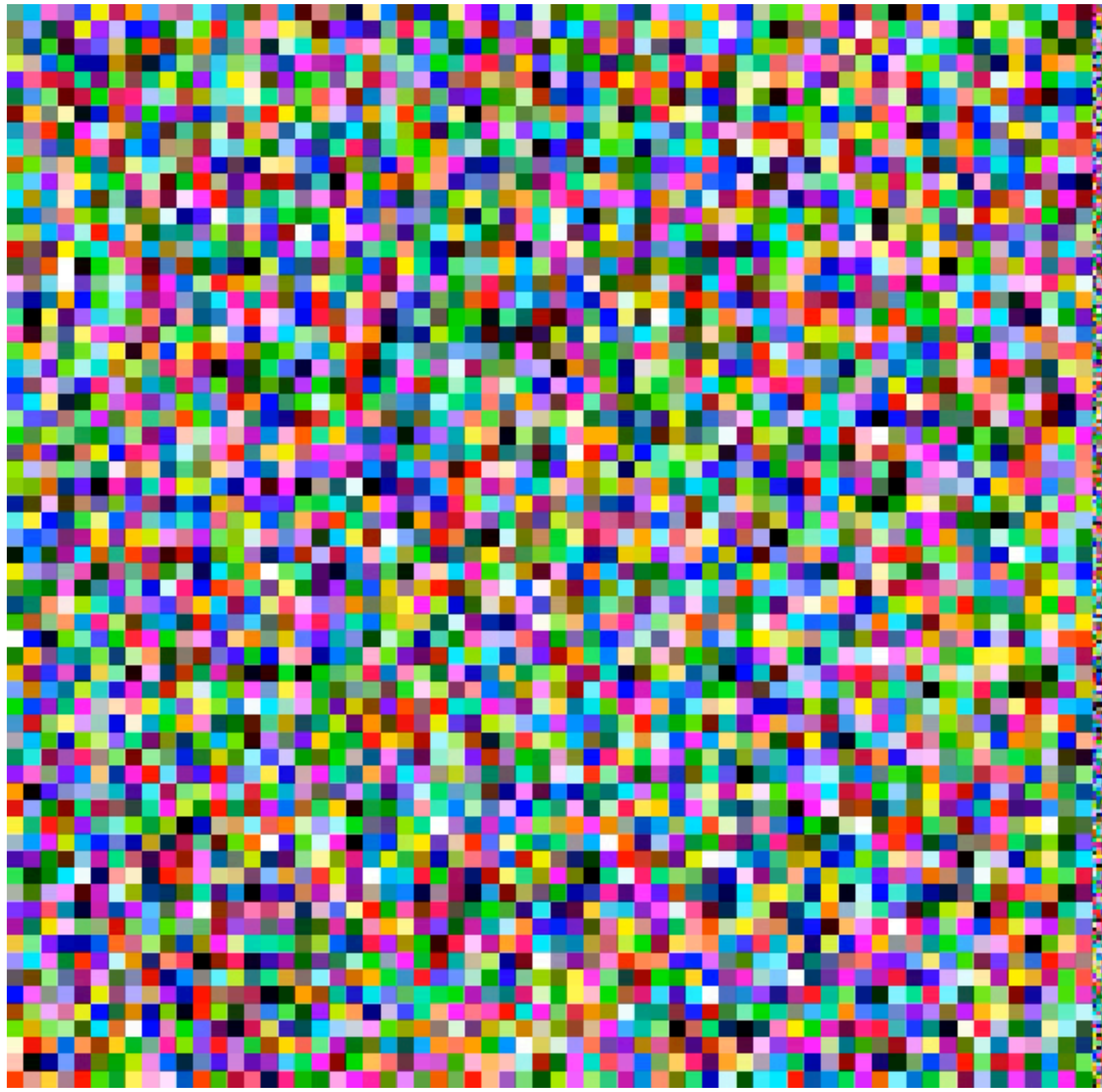
Multi-scale targets

$$64^2 \rightarrow \underbrace{(64^2, 256^2) \rightarrow (64^2, 256^2, 1024^2)}_{\text{Progressive Training}}$$

Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., "Matryoshka Diffusion Models," ICLR 2024

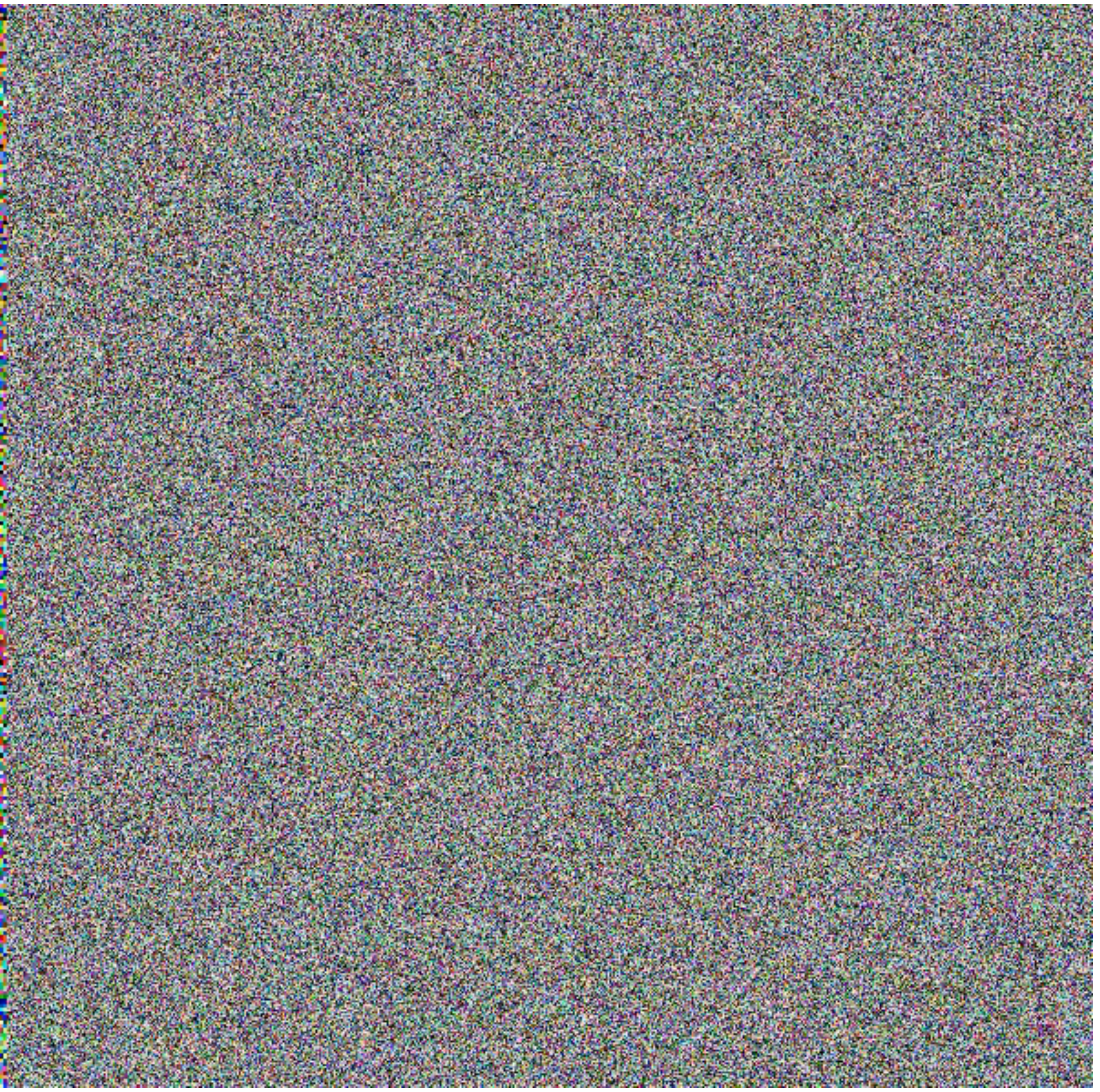
Progress of Generation

64x64



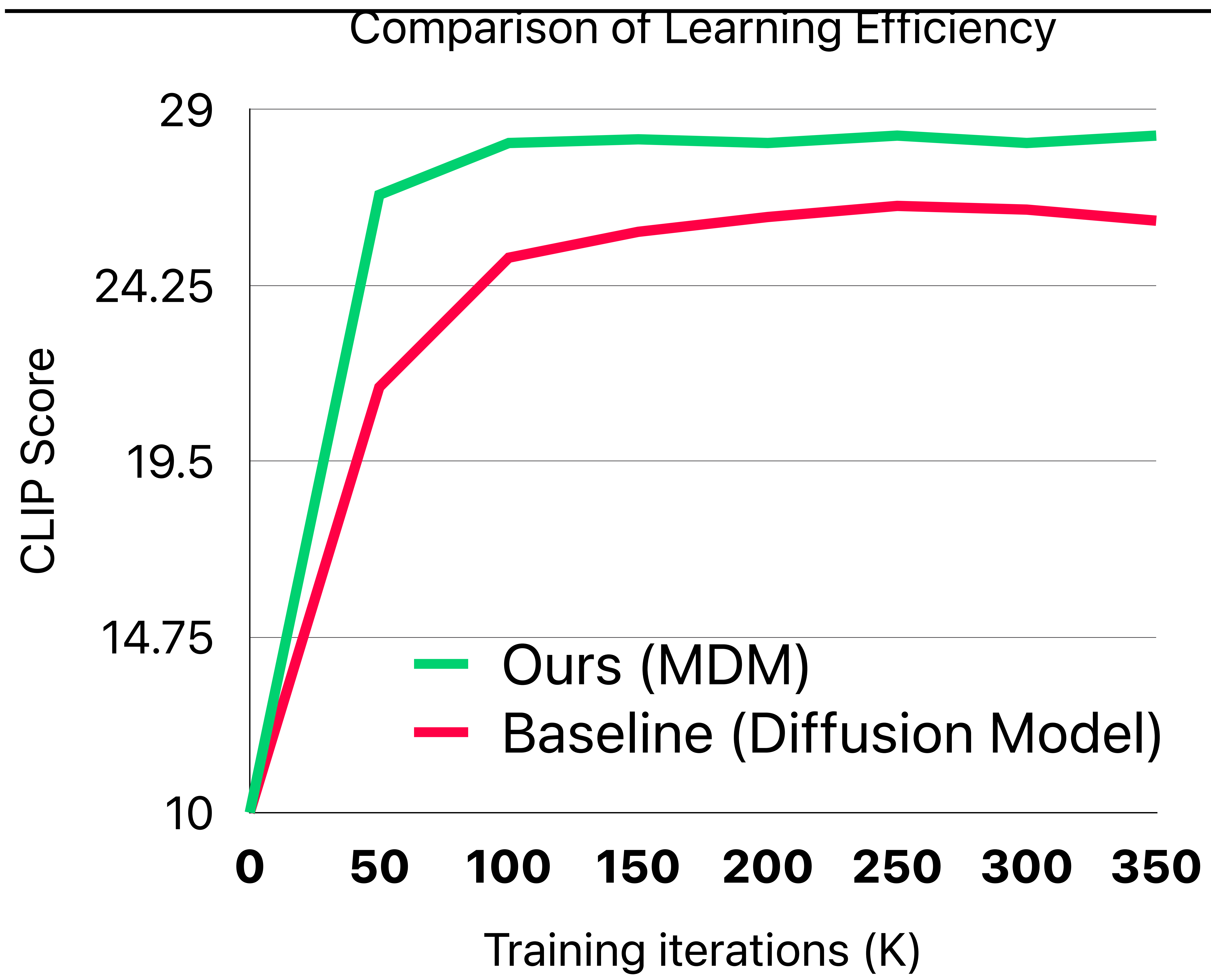
256x256

1024x1024



Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., “Matryoshka Diffusion Models,” ICLR 2024

Multi-scale Scales Better Than Single-scale



Incorporating a multi-scale structure learns diffusion much more efficiently than baseline.

Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., "Matryoshka Diffusion Models," ICLR 2024

Multi-scale Scales Better Than Single-scale

```
outputs = generate_image(prompt= "a poodle sitting on grass.",
custom_to_pil(outputs["denoised_images"][0])
```

```
Inferencing 1 examples for 1 times.
Keys in output: dict_keys(['denoised_images'])
Done, time spent 16.29 seconds.
```

Single-scale (512x512)

Ours (512x512)

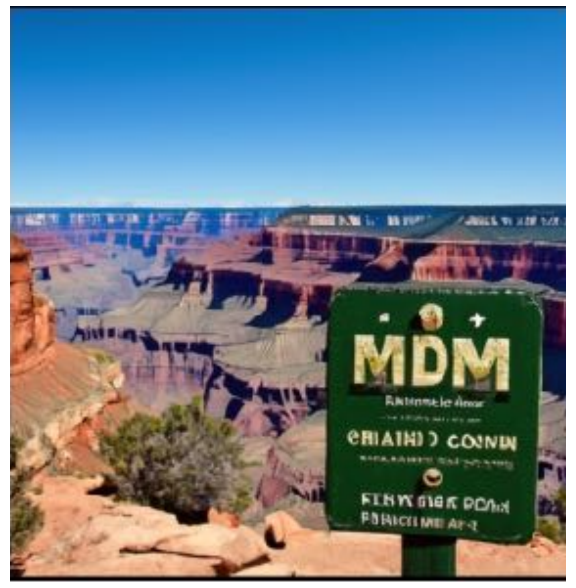
Results

MDM is the first single model at 1024px for text-to-image generation. Only 12M data.

A chromeplated cat sculpture placed on a Persian rug

A traditional Chinese garden in summer, oil paining by Claude Monet

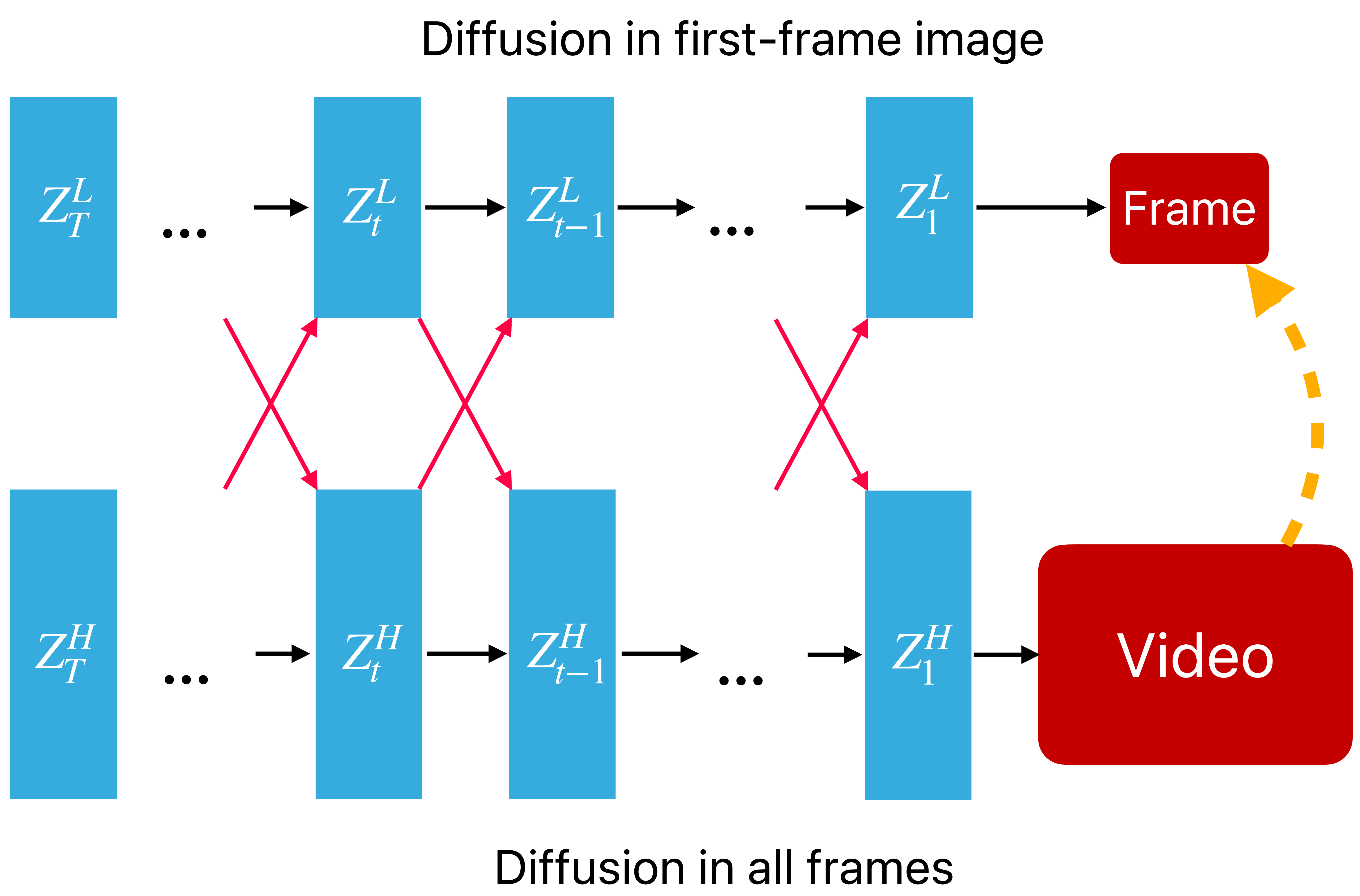
Cinematic photo of a fluffy koala with knitted hat holding a large cup of latte, close up, studio lighting, 4k



A green sign that says “MDM” and is at the edge of the Grand Canyon

a colorful artwork of Batman wearing sunglasses I romantic wall graffiti, close-up I dark pink and yellow I street murals

Also works for Video Generation



Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., "Matryoshka Diffusion Models," ICLR 2024

The Diversity Problem

Diversity of generation is variable and controlling the content can be difficult

Standard diffusion model

A cat sat on the mat → Diffusion Model →

Diffusion models, while adept at generating high-quality images from text, often produce limited visual diversity

Gu, J., Zhai, S., Zhang, Y., Susskind, J. & Jaitly, N., “Matryoshka Diffusion Models,” ICLR 2024

Why standard diffusion models fail?

Diffusion models use Classifier-free Guidance (CFG) to improve the generation:

$$\tilde{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, \mathbf{c}) = \gamma \cdot (\boldsymbol{x}_\theta(\boldsymbol{x}_t, \mathbf{c}) - \boldsymbol{x}_\theta(\boldsymbol{x}_t)) + \boldsymbol{x}_\theta(\boldsymbol{x}_t)$$

$$\nabla_x \log \tilde{p}_\theta(x \mid c) = \gamma \left[\nabla_x \left(\log \boxed{p_\theta(x \mid c)} - \log p_\theta(x) \right) \right] + \nabla_x \log p_\theta(x)$$

γ

Kaleido Diffusion

Explicitly model "mode selection" before applying diffusion steps

$$z \sim p_\theta(z \mid c)$$

Latent Modeling

$$x \sim \tilde{p}_\theta(x \mid z, c)$$

Latent-augmented Diffusion Models

- Diffusion with CFG:

$$\nabla_x \log \tilde{p}_\theta(x \mid c, z) = \gamma \left[\nabla_x \left(\log p_\theta(x \mid c) + \boxed{\log p_\theta(z \mid x, c)} - \log p_\theta(x) \right) \right] + \nabla_x \log p_\theta(x)$$

Kaleido-Diffusion Models

Adding autoregressive latent variables to improve controllability and diversity

Kaleido diffusion model

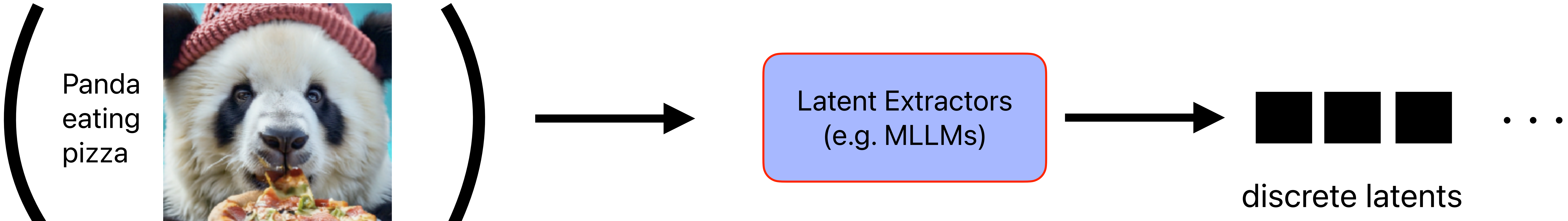
A cat sat on the mat → Diffusion Model →

A cat sat on the mat ↓ Autoregressive Model (Prior) → Latent tokens (e.g., captions, box, seg, visual tokens) ↑ Diffusion Model

Gu, J., Zhai, S., Zhang, Y., Jaitly, N., & Susskind, J.,
"*Kaleido Diffusion: Improving Conditional Diffusion Models with Autoregressive Latent Modeling*," Arxiv 2024

Generating the Posteriors of Latents

Use other models / data to generate discrete latents from the images



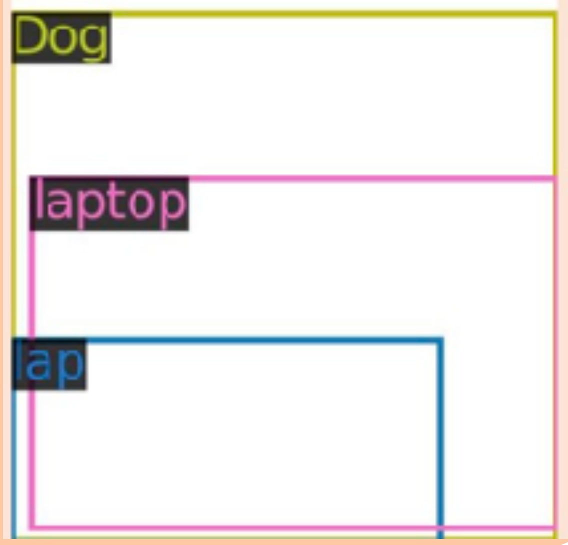
Gu, J., Zhai, S., Zhang, Y., Jaitly, N., & Susskind, J.,
"*Kaleido Diffusion: Improving Conditional Diffusion Models with Autoregressive Latent Modeling*," Arxiv 2024

Generating the Posteriors of Latents

Can use a set of Pretrained models to generate a variety of descriptors

Object blobs

Detection bounding boxes



Caption: Dog Lying on a human’s lap

Textual Descriptions

A person in a blue sweater and jeans is sitting on the floor on top of a gray couch with their laptop in their lap. They have a yellow Labrador Retriever in their lap, who is looking at the camera. The dog has its tongue out and is lying down on the person’s lap…

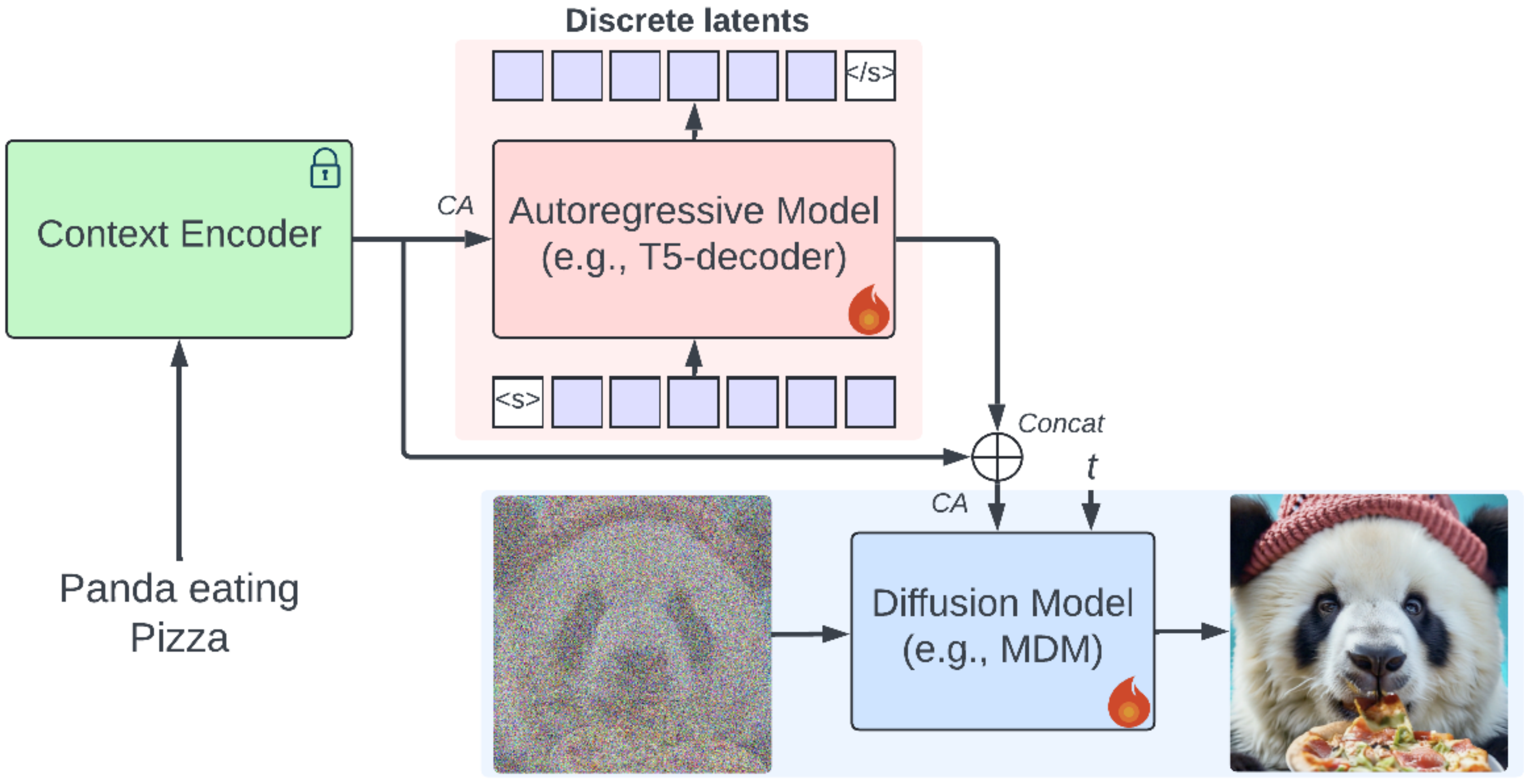
Visual Tokens

1 13 4 7 · · · 9

Gu, J., Zhai, S., Zhang, Y., Jaitly, N., & Susskind, J.,
“*Kaleido Diffusion: Improving Conditional Diffusion Models with Autoregressive Latent Modeling*,” Arxiv 2024

Autoregressive and Diffusion Joint Training

Can use a set of Pretrained models to generate a variety of descriptors



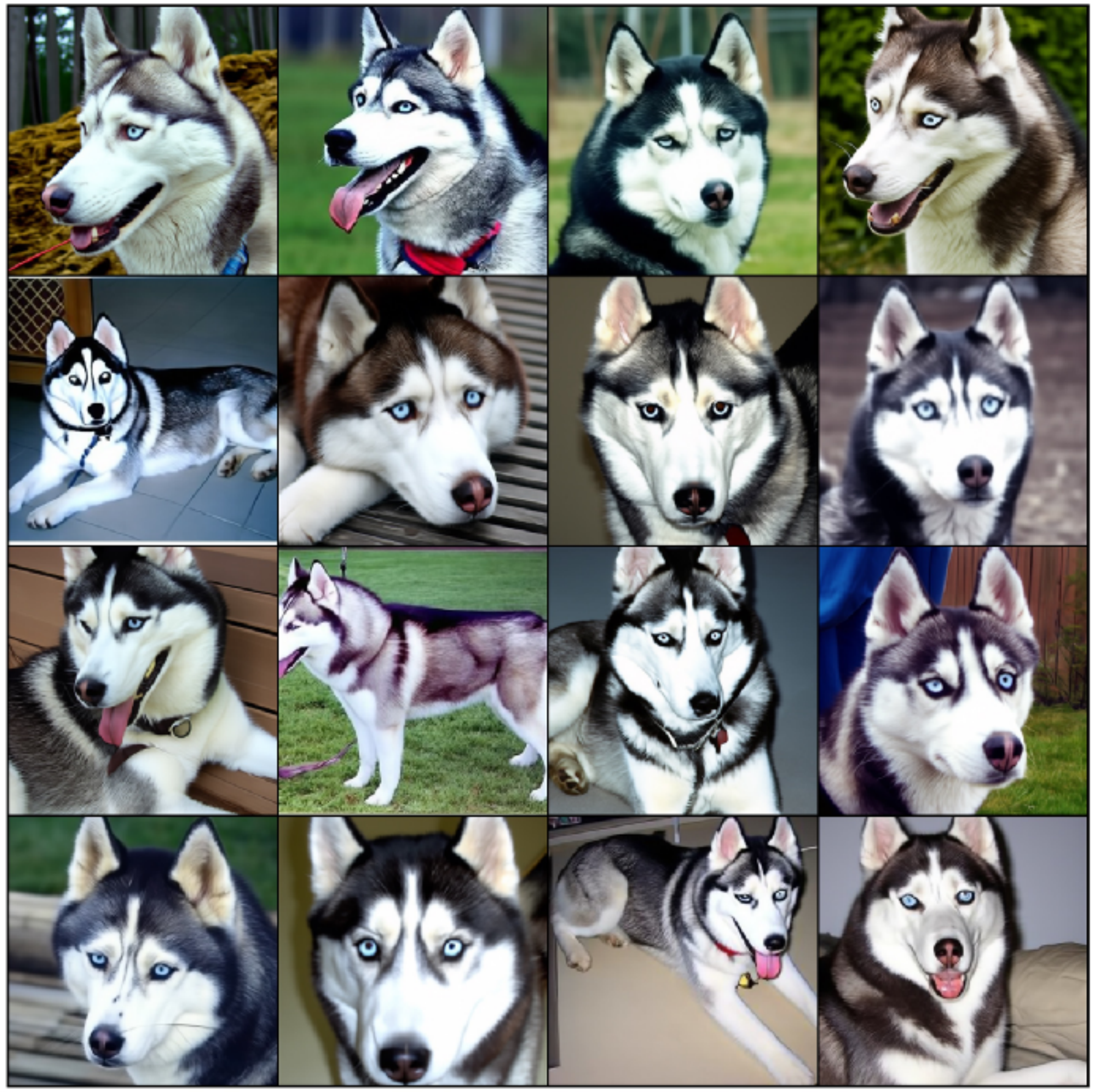
Stage II: Autoregressive and Diffusion Joint Training

$$L = L^{DM} + \eta \cdot L^{AR}$$

Gu, J., Zhai, S., Zhang, Y., Jaitly, N., & Susskind, J.,
"*Kaleido Diffusion: Improving Conditional Diffusion Models with Autoregressive Latent Modeling*," Arxiv 2024

Much More Diverse Generations

"Siberian husky" (Class to Image Generation)



baseline

Kaleido-diffusion

Much More Diverse Generations

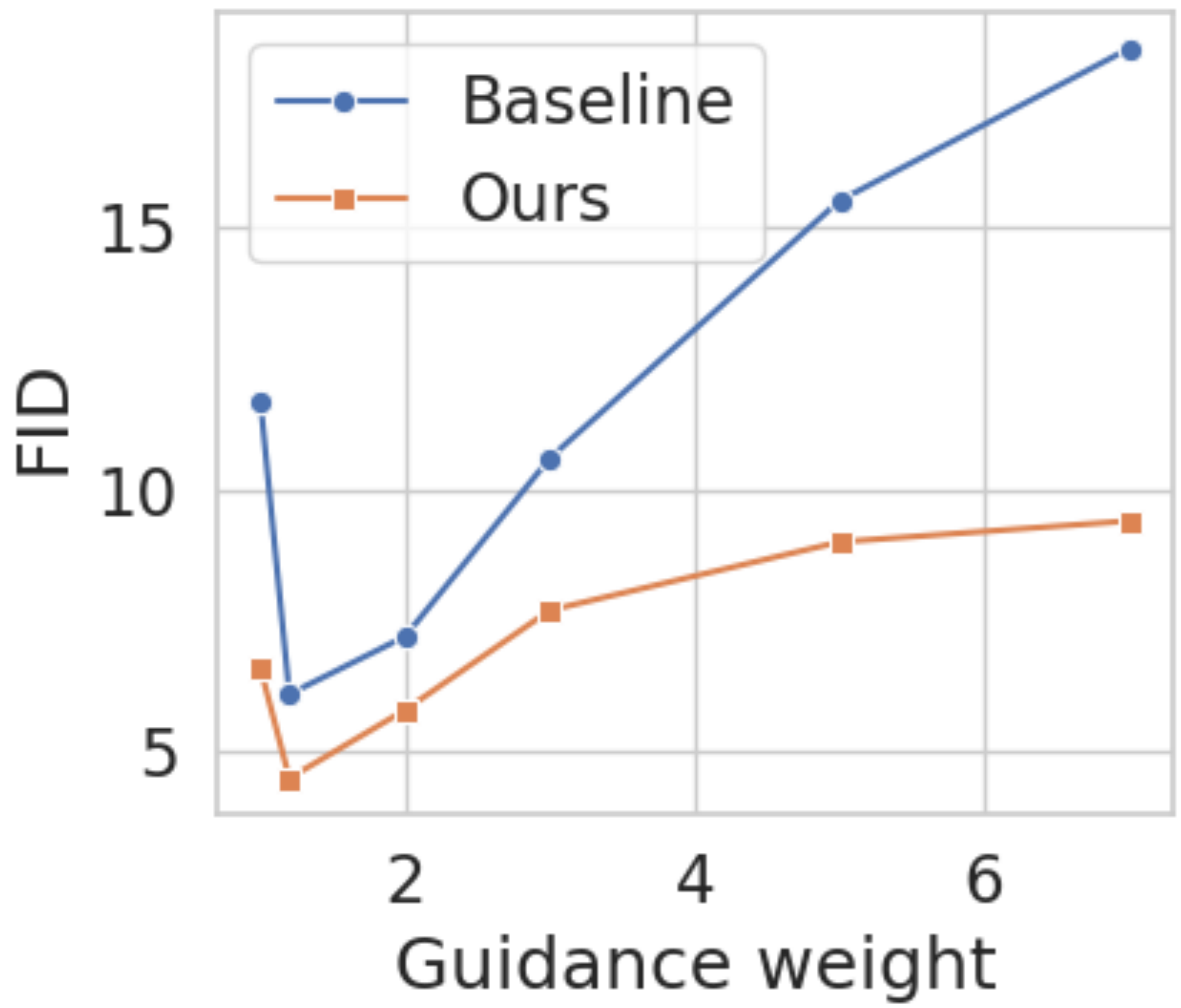
"A bald eagle made of chocolate powder, mango, and whipped cream" (Text to image generation)

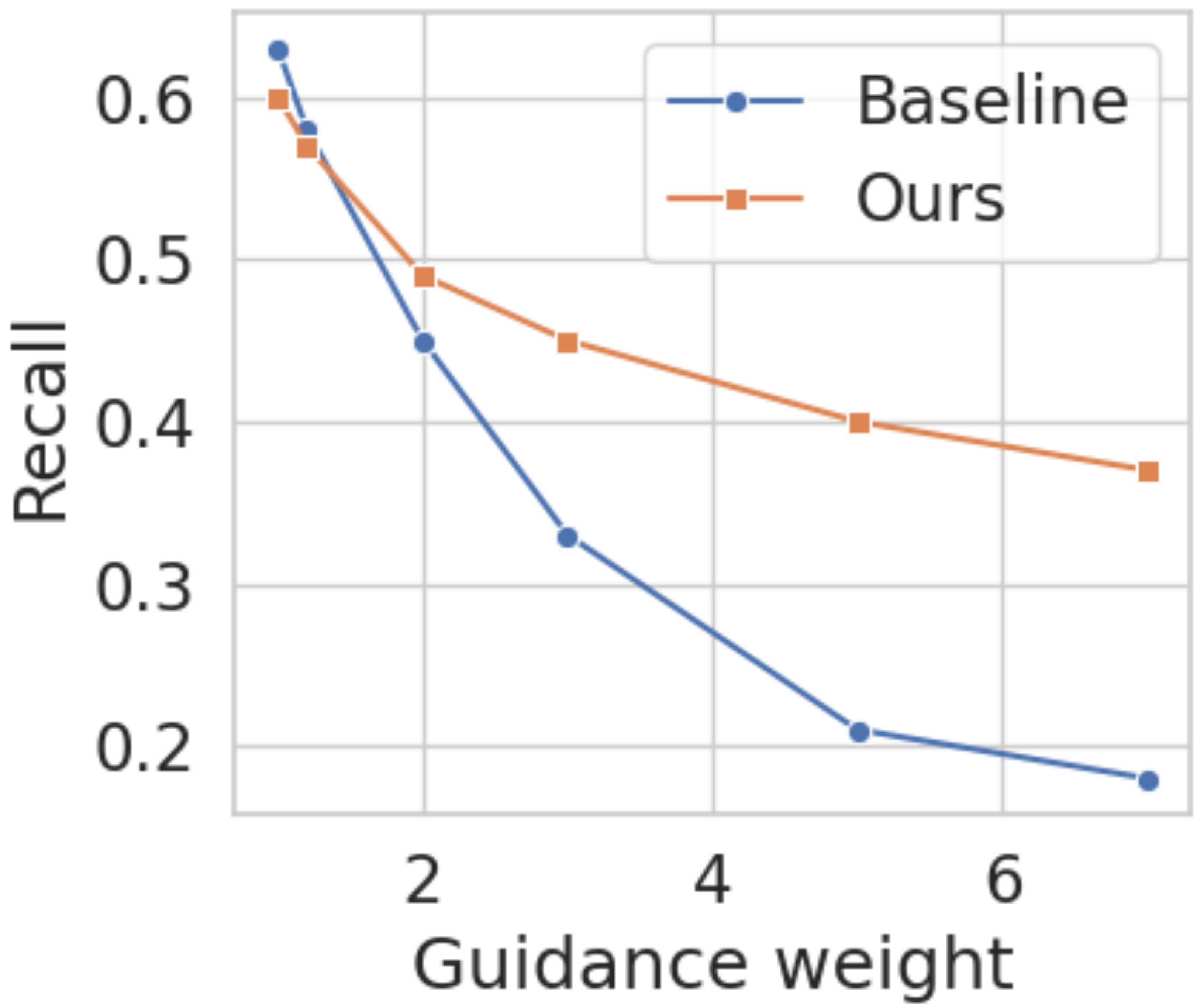
baseline

Kaleido-diffusion

Quantitative Results

- Kaleido consistently enhances the diversity of samples without compromising their quality across different CFG



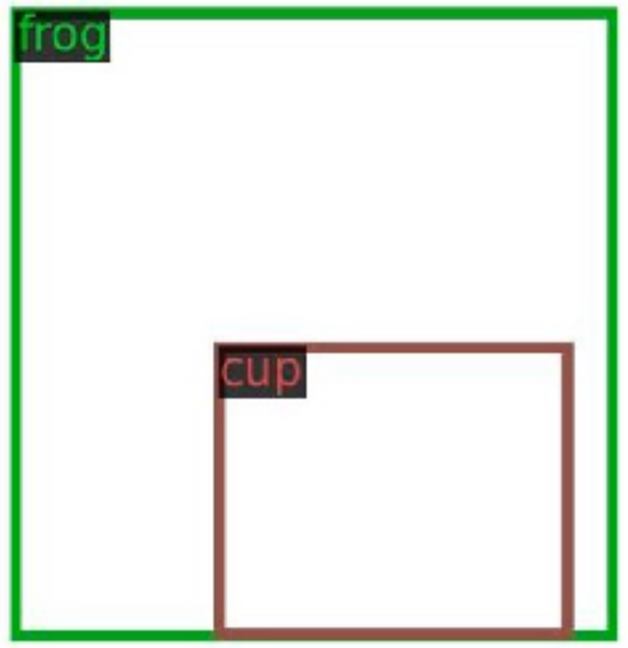


Latent Editing

Input: "A photo of a frog drinking coffee"

In the image a frog is seen sipping on a cup of coffee, seemingly enjoying a relaxing break. The frog is positioned on a log with its eyes closed and a small smile on its face, as if it's savoring the flavor of the coffee. The cup of coffee is placed on a rock next to the frog, and the background features a body of water. The frog's green and yellow coloration stands out against the natural setting, making for a charming and whimsical scene.

+



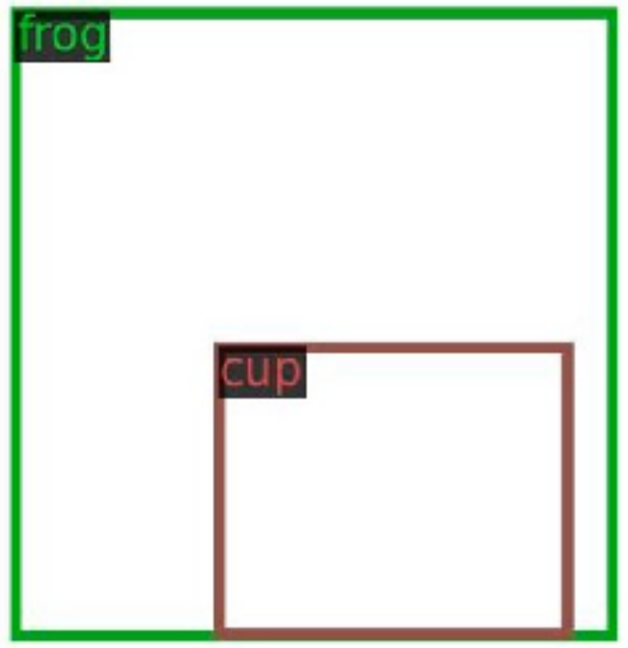
Latents generated

Image generated by diffusion

Latent Editing

Input: "A photo of a frog drinking coffee"

In the image a frog is seen sipping on a cup of coffee, seemingly enjoying a relaxing break. The frog is positioned on a log with its eyes closed and a small smile on its face, as if it's savoring the flavor of the coffee. The cup of coffee is placed on a rock next to the frog, and the background features a body of water. The frog's green and yellow coloration stands out against the natural setting, making for a charming and whimsical scene.



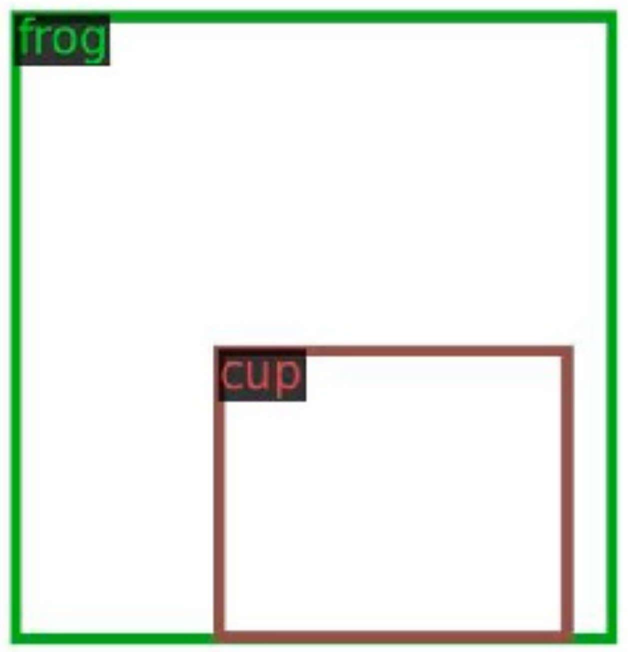
Latents generated

Image generated by diffusion

Latent Editing

Input: “A photo of a frog drinking coffee”

In the image a frog is seen sipping on a cup of coffee, seemingly enjoying a relaxing break. The frog is positioned on cobblestones with its eyes closed and a small smile on its face, as if it’s savoring the flavor of the coffee. The cup of coffee is placed on a rock next to the frog, and the background features forest. The frog’s green and yellow coloration stands out against the natural setting, making for a charming and whimsical scene.



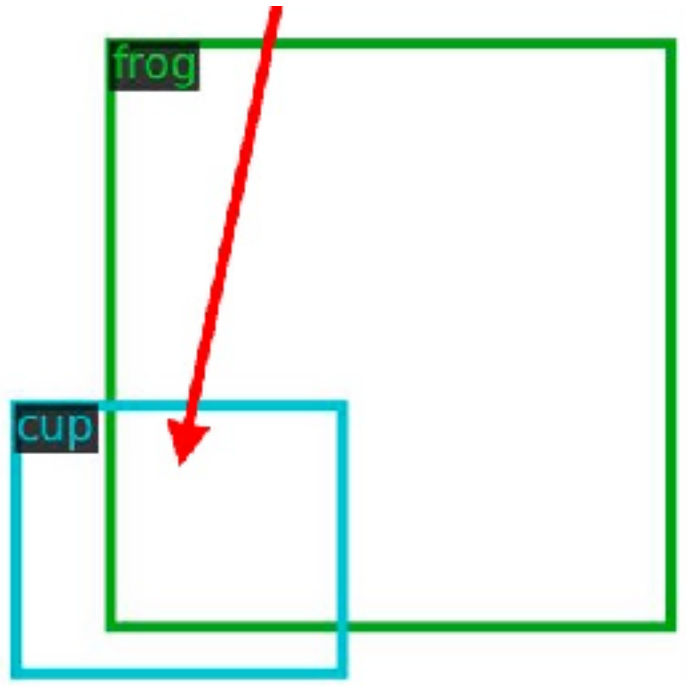
Edited Latents

Image regenerated by diffusion

Latent Editing

Input: "A photo of a frog drinking coffee"

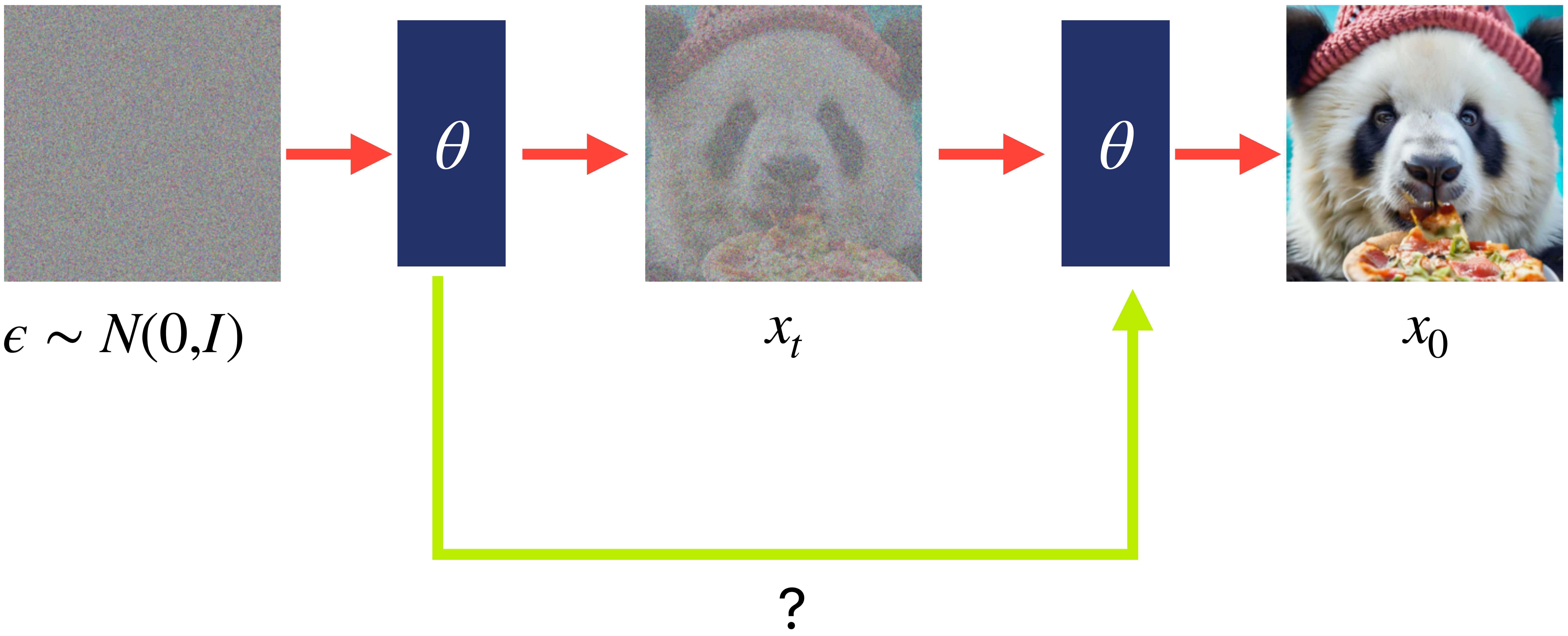
In the image a frog is seen sipping on a cup of coffee, seemingly enjoying a relaxing break. The frog is positioned on cobblestones with its eyes closed and a small smile on its face, as if it's savoring the flavor of the coffee. The cup of coffee is placed on a rock next to the frog, and the background features forest. The frog's green and yellow coloration stands out against the natural setting, making for a charming and whimsical scene.



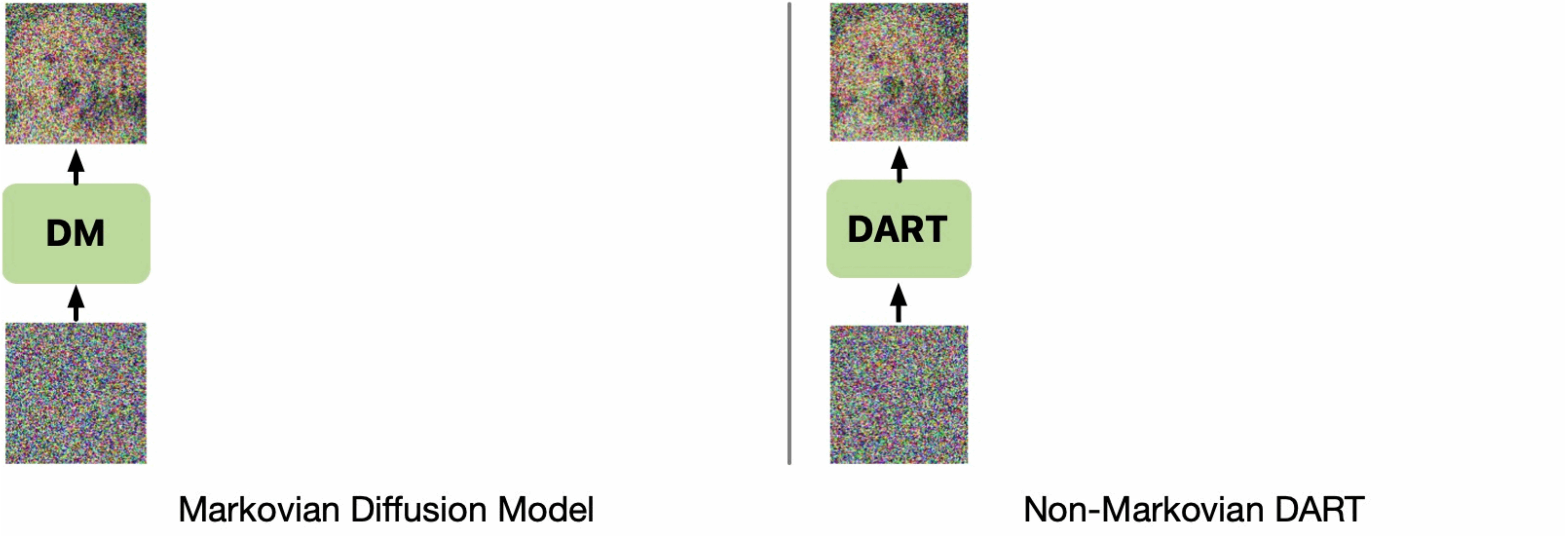
Edited Latents

Image regenerated by diffusion

Is Diffusion the best answer?

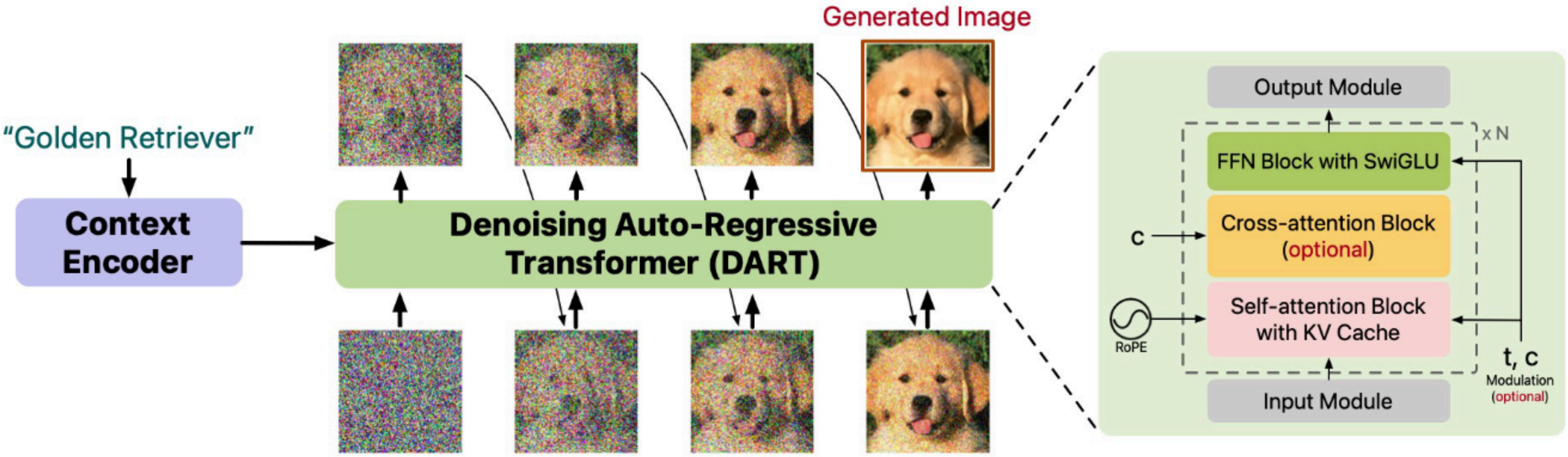


Denoising Autoregressive Transformer



Gu, J., Wang, Y., Zhang, Y., Zhang, Q., Zhang, D., Jaitly, N., Susskind, J., Zhai, S. “DART: Denoising Autoregressive Transformer for Scalable Text-to-Image Generation”, Arxiv 2024

Denoising Autoregressive Transformer

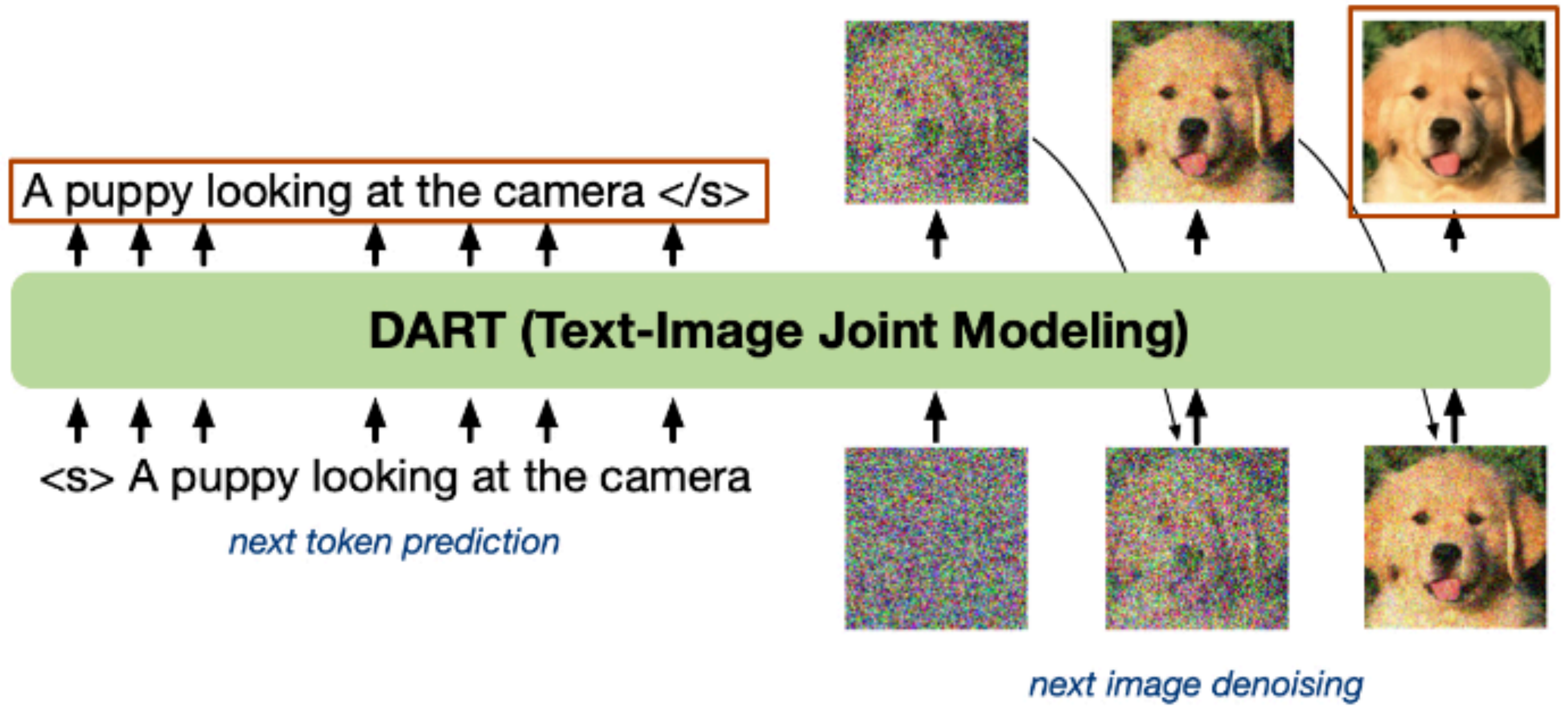


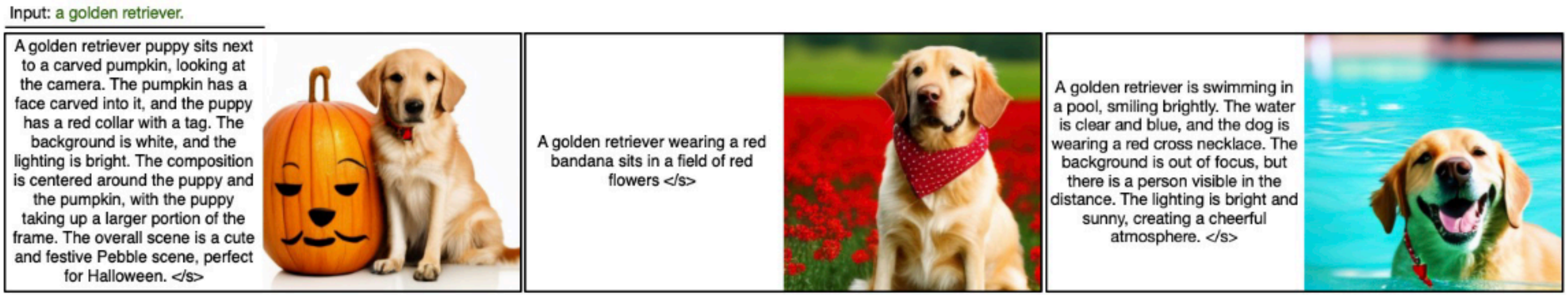
Gu, J., Wang, Y., Zhang, Y., Zhang, Q., Zhang, D., Jaitly, N., Susskind, J., Zhai, S. "DART: Denoising Autoregressive Transformer for Scalable Text-to-Image Generation", Arxiv 2024

Denoising Autoregressive Transformer

Gu, J., Wang, Y., Zhang, Y., Zhang, Q., Zhang, D., Jaitly, N., Susskind, J., Zhai, S. “DART: Denoising Autoregressive Transformer for Scalable Text-to-Image Generation”, Arxiv 2024

Denoising Autoregressive Transformer

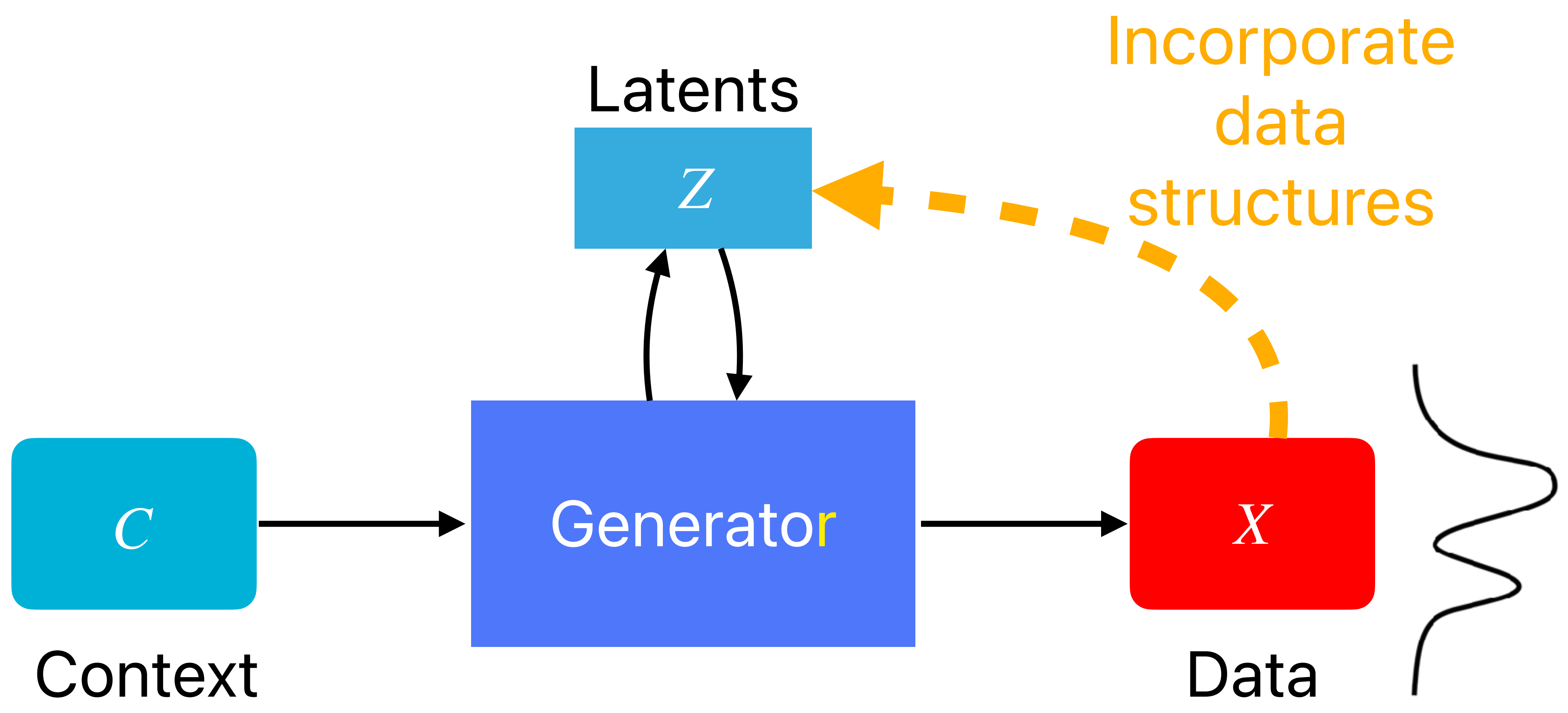




Gu, J., Wang, Y., Zhang, Y., Zhang, Q., Zhang, D., Jaitly, N., Susskind, J., Zhai, S. “DART: Denoising Autoregressive Transformer for Scalable Text-to-Image Generation”, Arxiv 2024

Takeaway

- We can enhance learning scalability from high-dimensional data by using hierarchical and discrete structures to model the latents.



Scalable

Knowledgeable

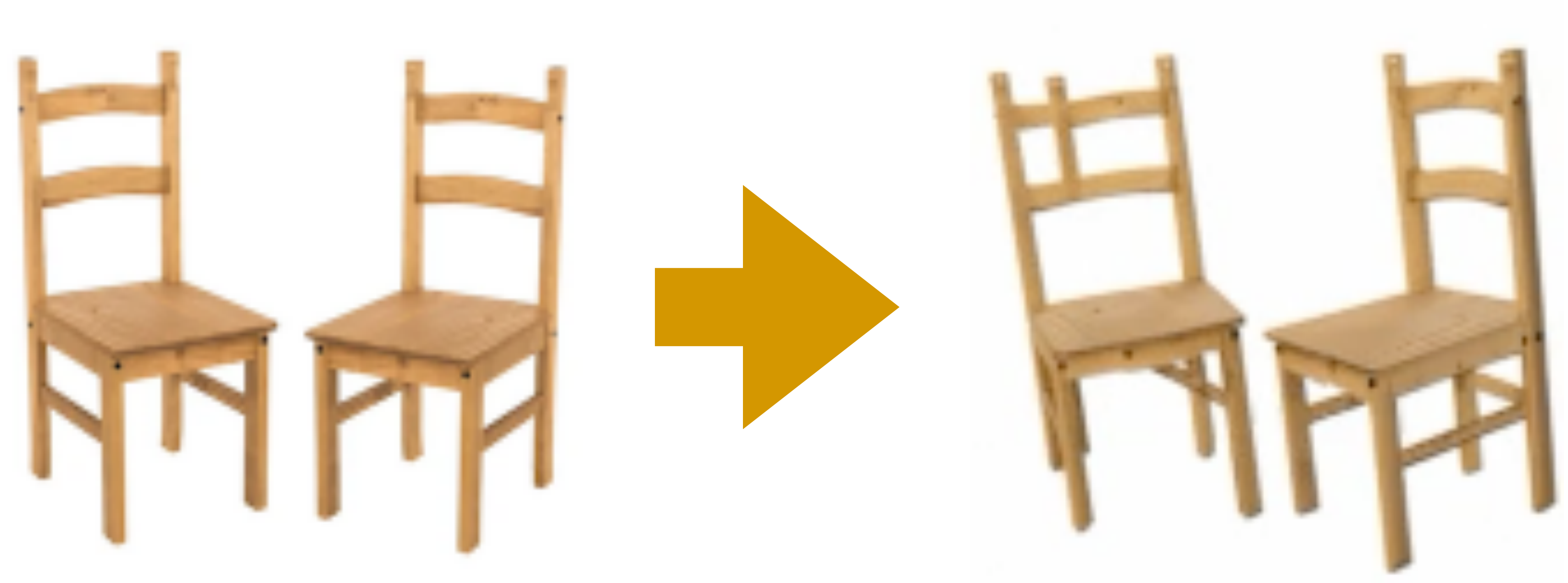
Why Need World Knowledge?

Can SOTA Generative Models learn 3D?

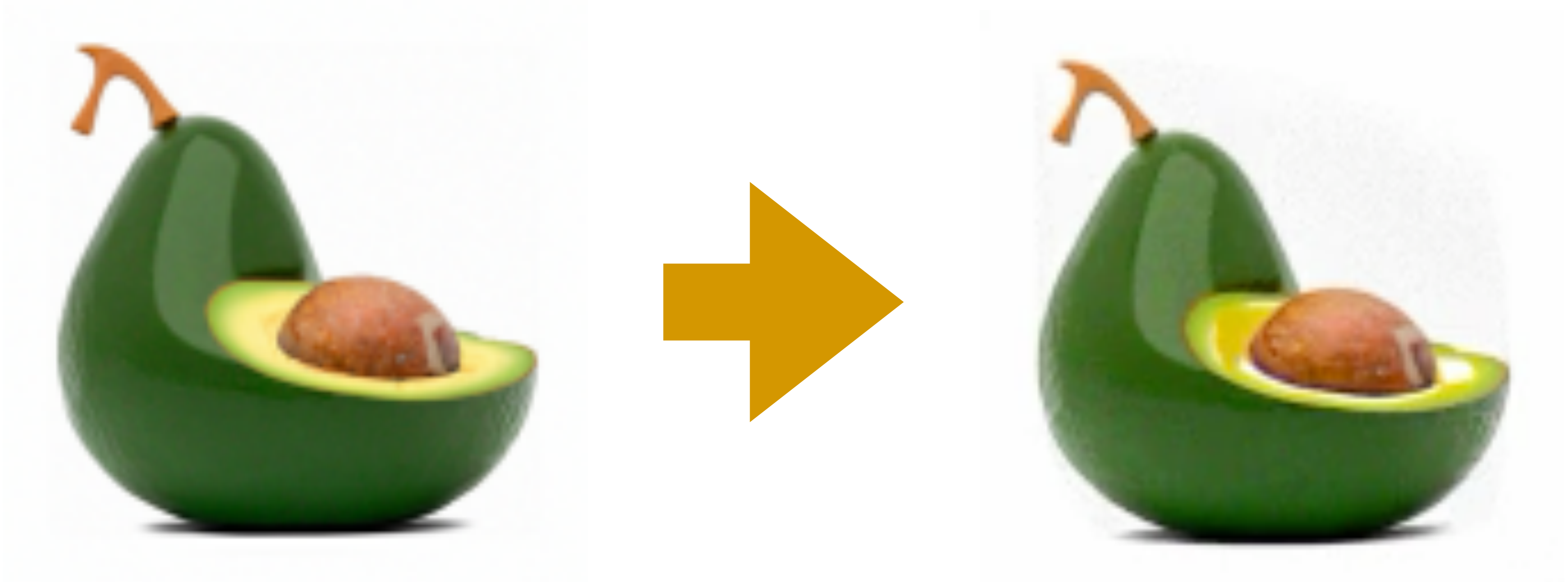
viewpoint condition

Issues with Pure 2D Models

Results of 2D diffusion models:



Context Output

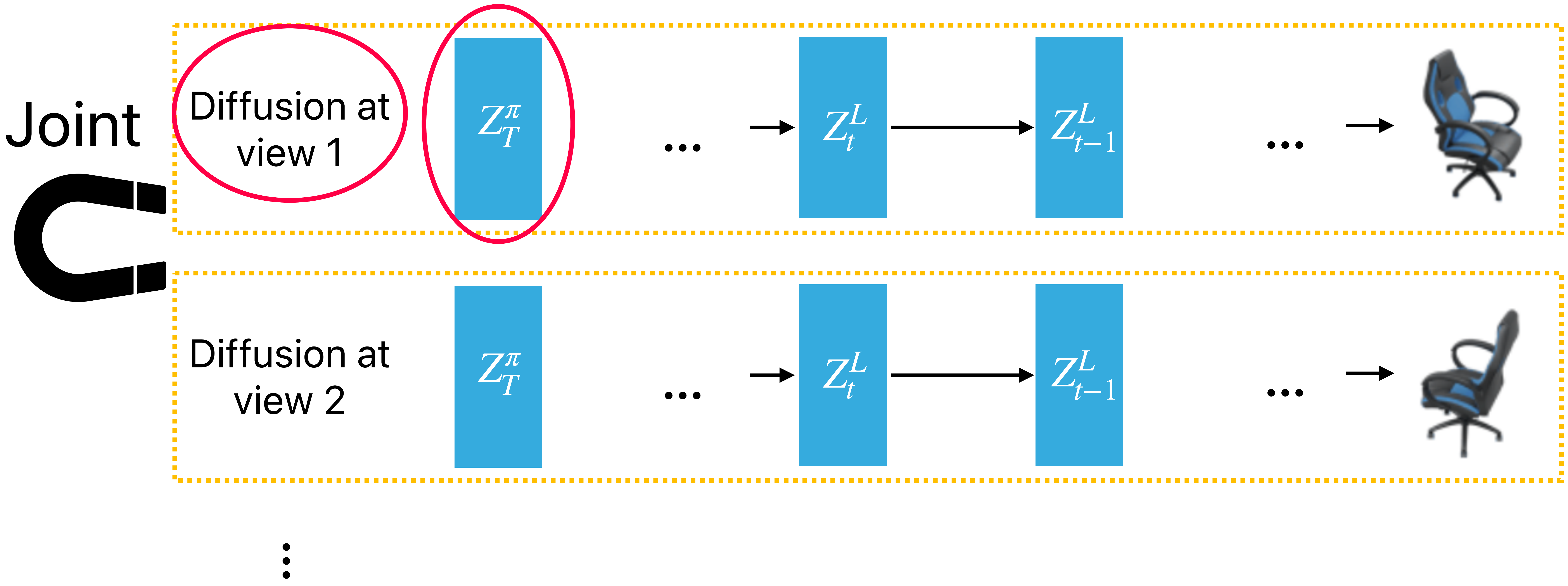


Context Output

Watson, Daniel, et al. "*Novel view synthesis with diffusion models.*" ICLR 2023.

Issues with Pure 2D Models

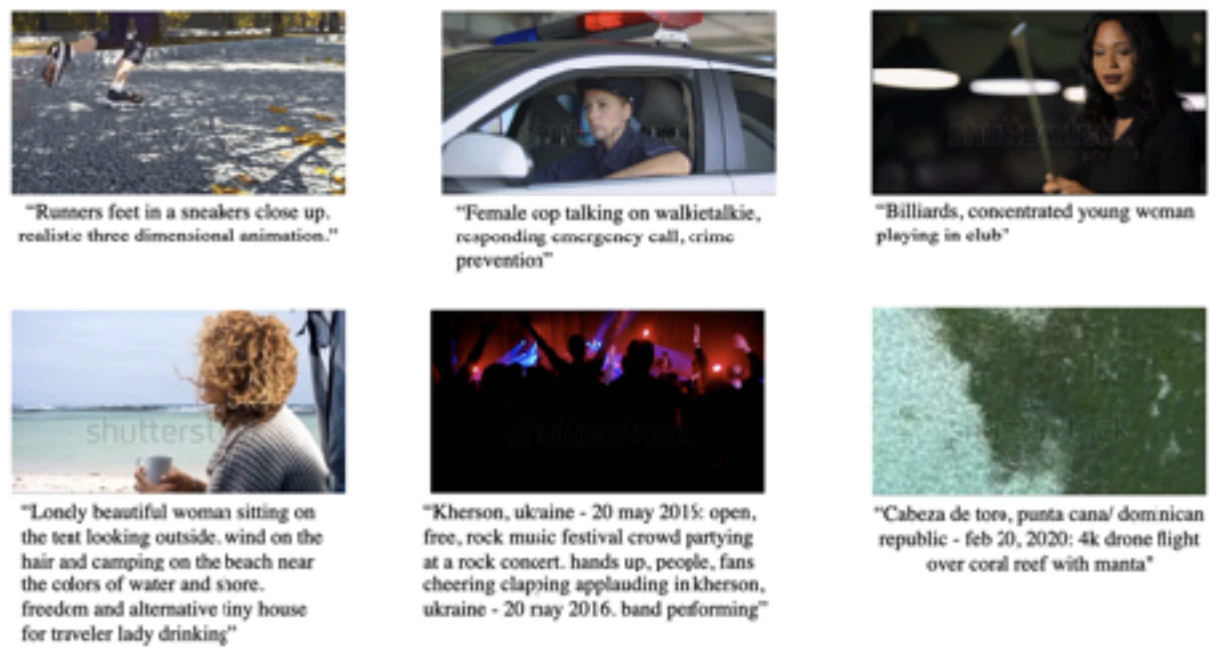
1. Randomness in each view;



Need multi-view datasets;

Not generalize to unseen views

Implicitly learn through large amounts of video data.



Large scale video dataset

→ SORA →

Pure 2D/video network

Drawback:

(a) Data/resource hungry

(b) No 3D guarantee.

Failure cases (again)

Explicit World Knowledge Modeling

World knowledge

Latents

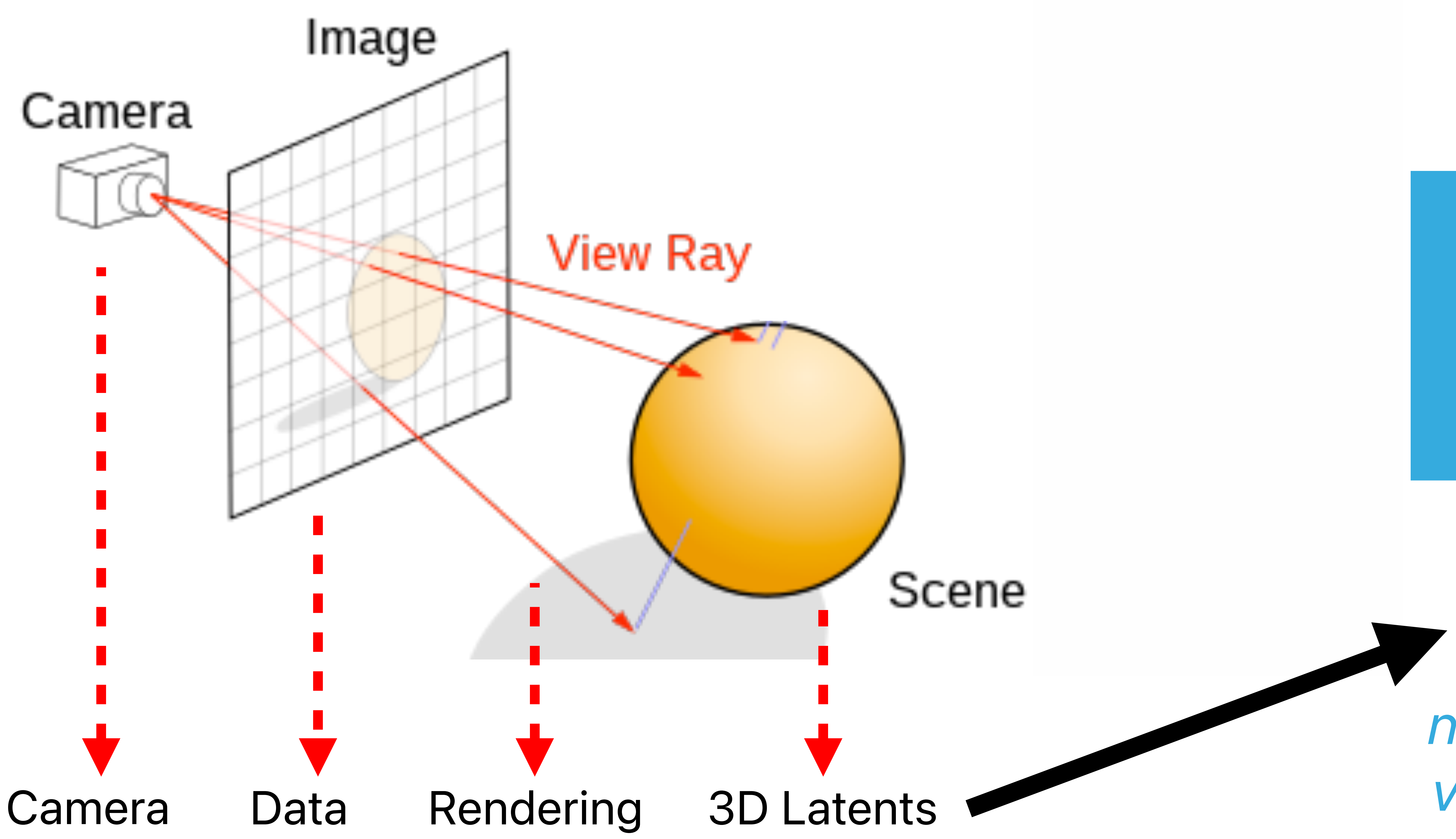
The external world knowledge acts as an additional constraint to regularize the generative process.

Context

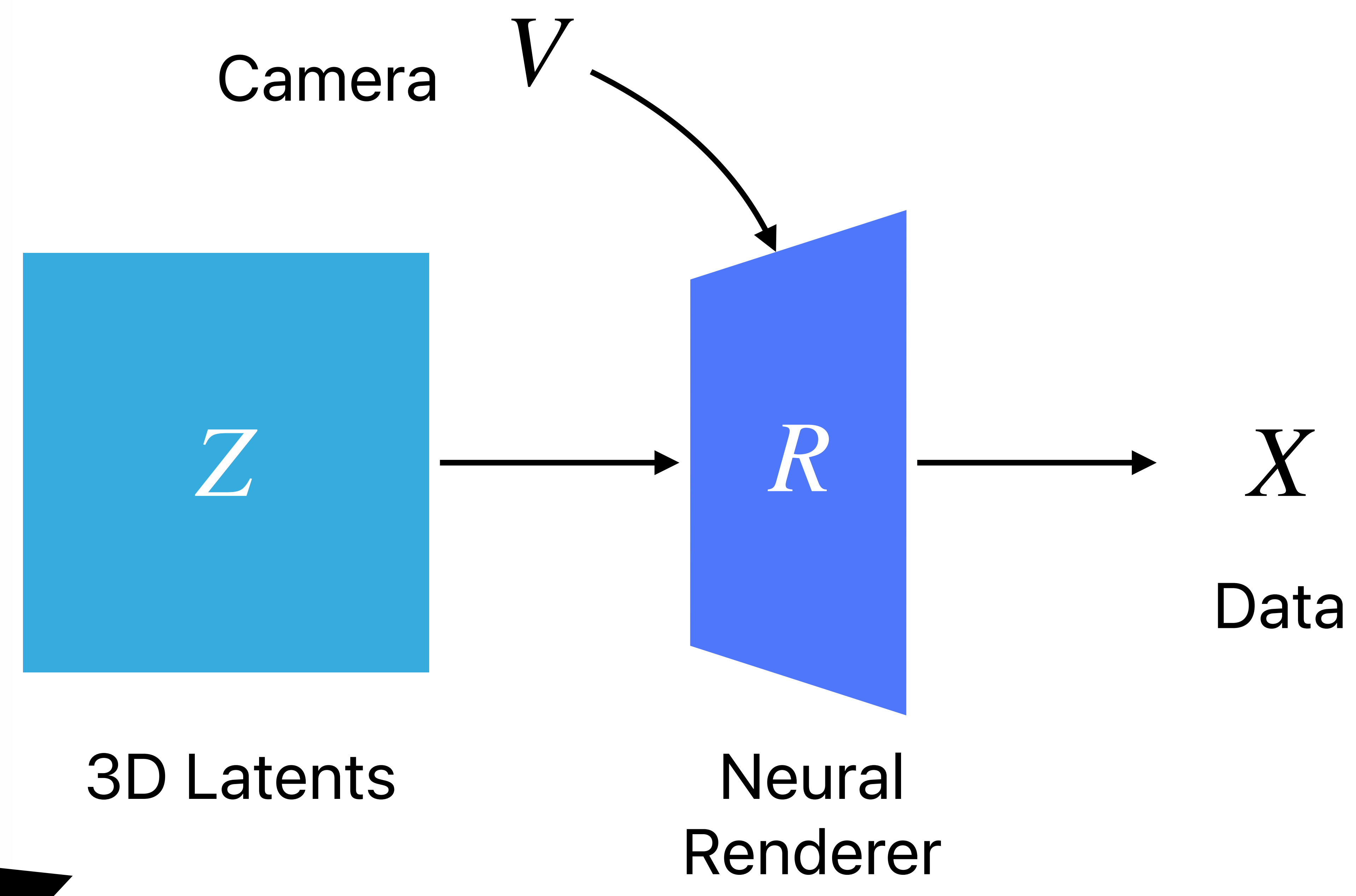
Data

How natural images are created

Computer Graphics



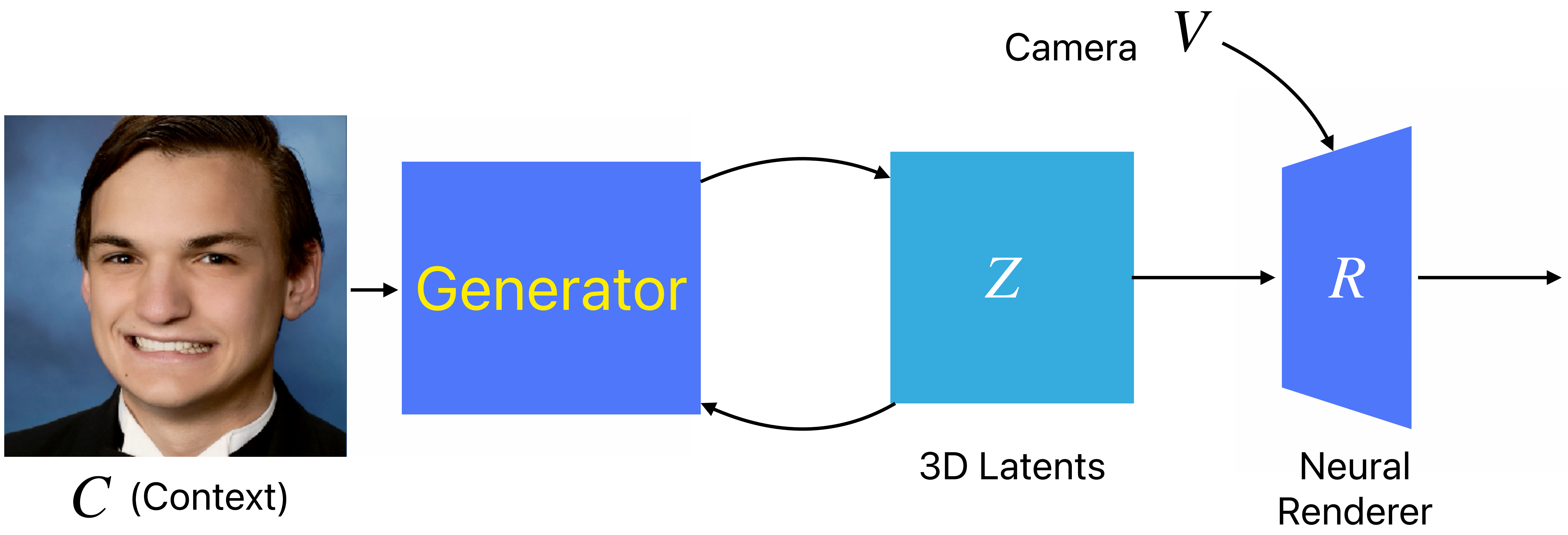
Neural Rendering from 3D Latents



neural fields, voxels, point clouds, meshes, etc

volume rendering, rasterization, etc

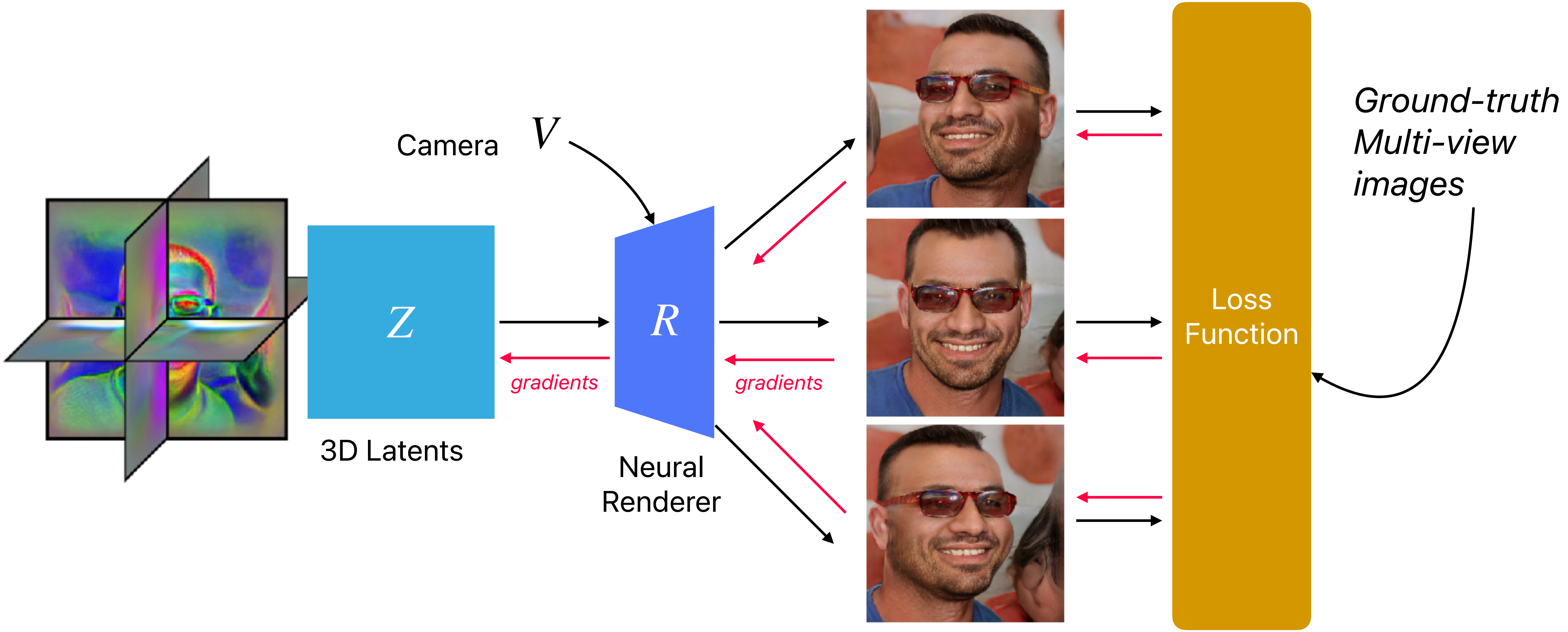
3D-aware Generative Models



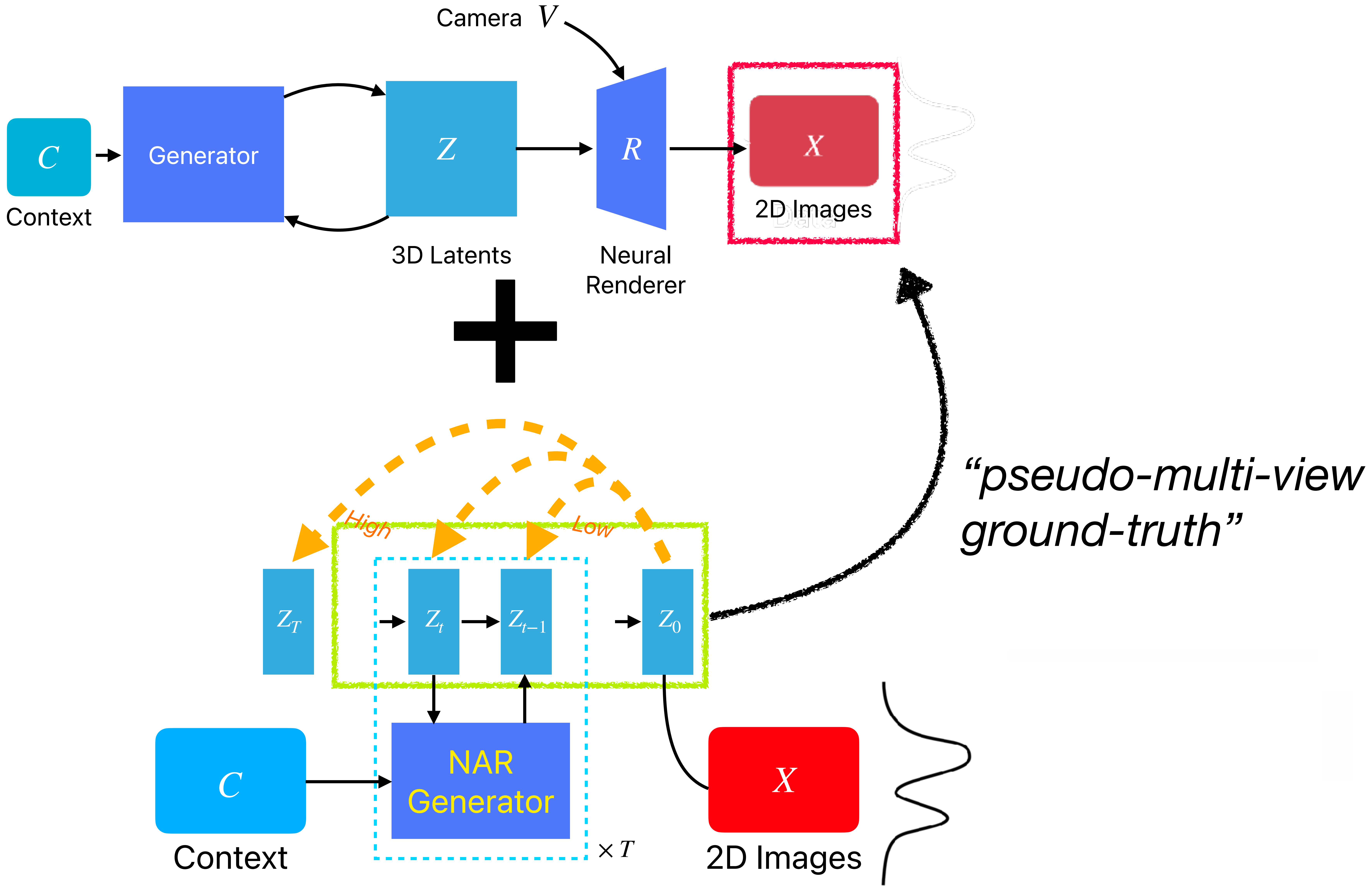
A model grounded in 3D can generalize to new views freely without much training.

Reconstruction from Images

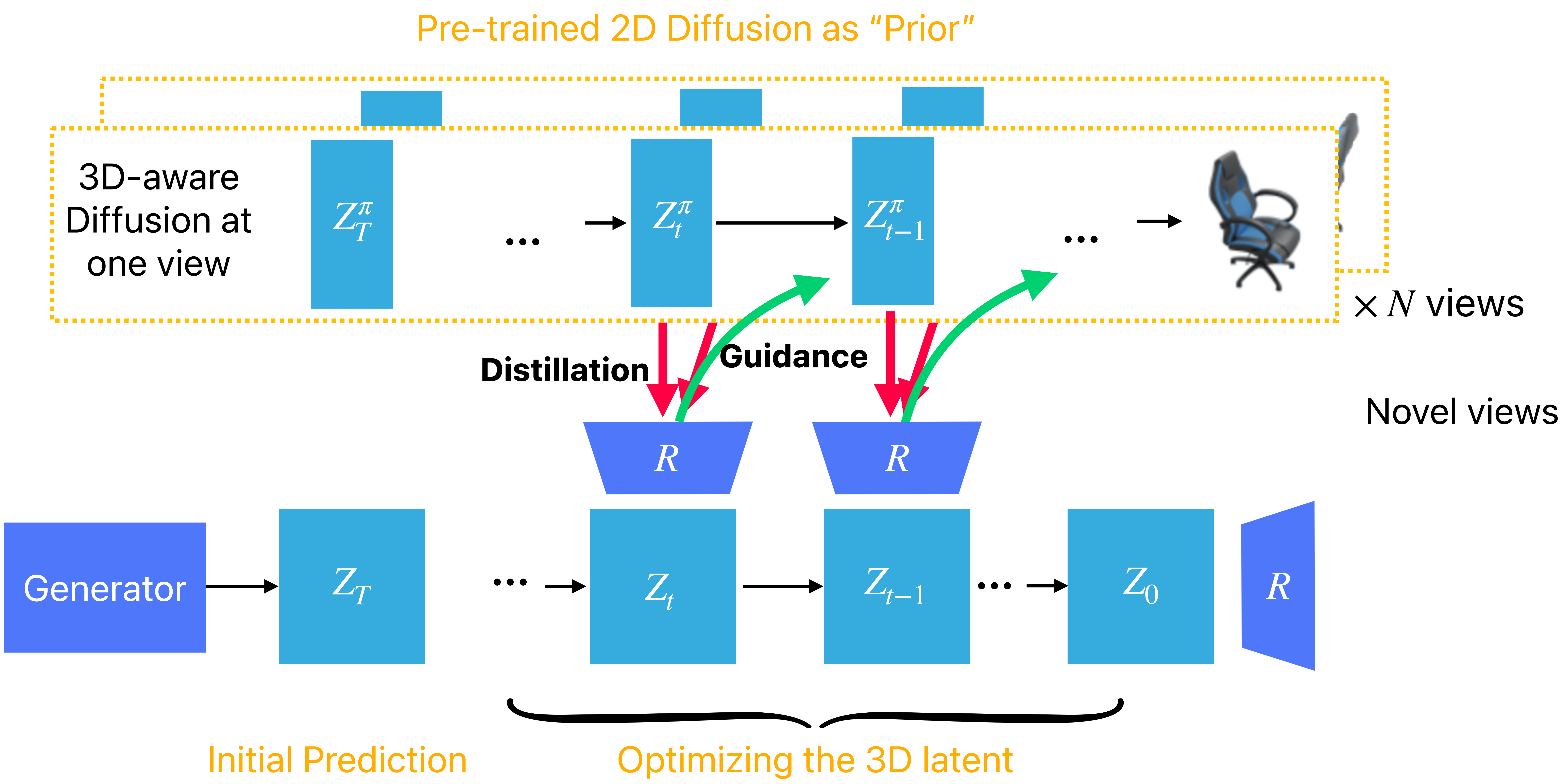
Neural rendering from 3D Latents, gradient back-propagate to update 3D latents



How to Generate 3D Latents?



Distilling Latents from 2D Diffusion!



Distillation

Using denoised views as the rendering target to fine-tune the 3D latents

Guidance

Using the rendered image to guide the multi-view diffusion to move into next step

Gu, J., Trevithick, A., Lin, K. E., Susskind, J. M., Theobalt, C., Liu, L., & Ramamoorthi, R., "NerfDiff: Single-image View Synthesis with NeRF-guided Distillation from 3D-aware Diffusion," ICML 2023

Comparison

2D only | With 3D | 2D only | With 3D

Gu, J., Trevithick, A., Lin, K. E., Susskind, J. M., Theobalt, C., Liu, L., & Ramamoorthi, R.,
“NerfDiff: Single-image View Synthesis with NeRF-guided Distillation from 3D-aware Diffusion,” ICML 2023

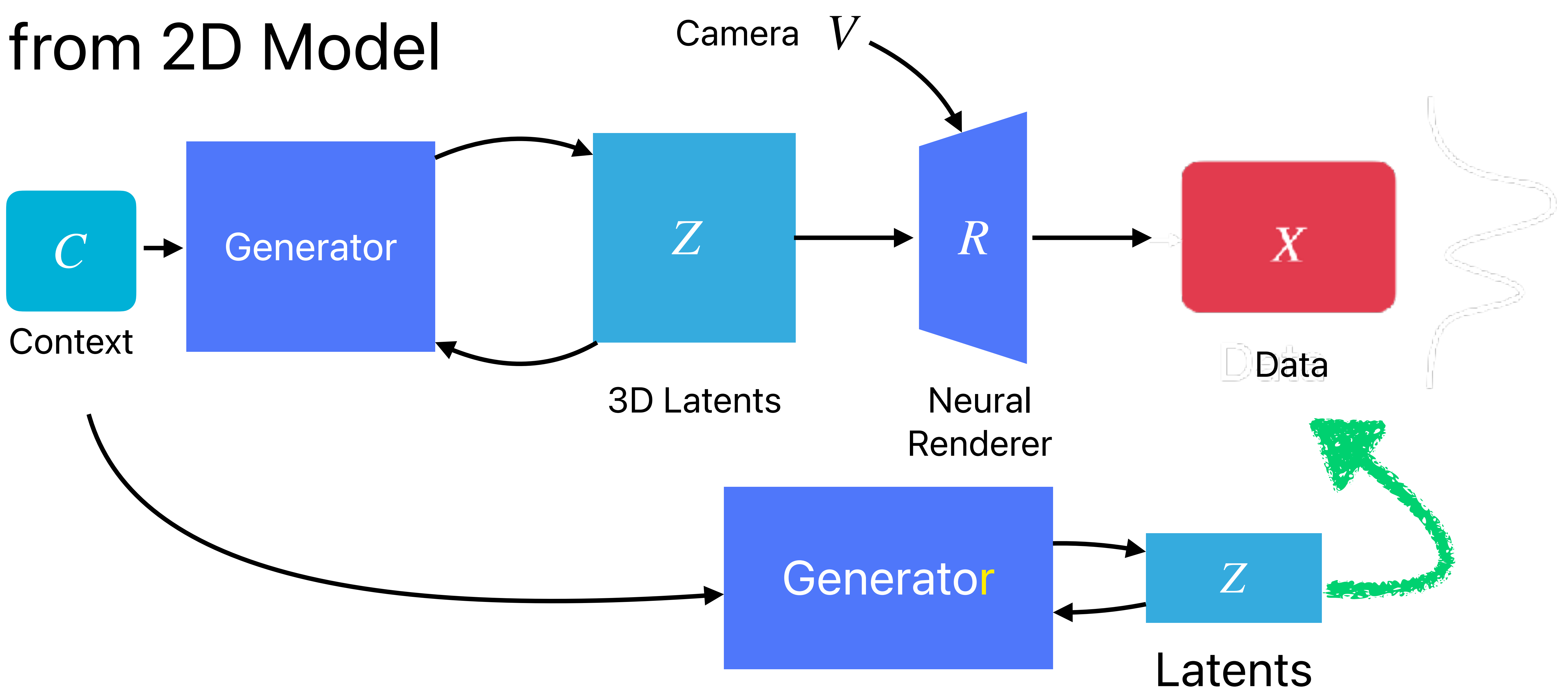
Comparison

	ShapeNet Cars				ShapeNet Chairs				Amazon-Berkeley Objects			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
LFN (Sitzmann et al., 2021)*	22.42	0.89	–	–	22.26	0.90	–	–	–	–	–	–
3DiM (Watson et al., 2022)*	21.01	0.57	–	**8.99**	17.05	0.53	–	6.57	–	–	–	–
SRN (Sitzmann et al., 2019a)	22.25	0.88	0.129	41.21	22.89	0.89	0.104	26.51	–	–	–	–
PixelNeRF (Yu et al., 2021)	23.17	0.89	0.146	59.24	23.72	0.90	0.128	38.49	–	–	–	–
CodeNeRF (Jang & Agapito, 2021)	22.73	0.89	0.128	–	23.39	0.87	0.166	–	–	–	–	–
FE-NVS (Guo et al., 2022)	22.83	0.91	0.099	–	23.21	0.92	0.077	–	–	–	–	–
VisionNeRF (Lin et al., 2023)	22.88	0.90	0.084	21.31	24.48	0.92	0.077	10.05	28.61	0.93	0.095	33.38
NerfDiff-B (Ours)	23.51	**0.92**	0.082	18.09	24.79	0.94	**0.056**	5.65	32.81	0.96	0.057	7.77
w/o NGD	23.81	**0.92**	0.093	42.37	24.77	0.93	0.068	15.72	32.07	0.95	0.063	18.01
NerfDiff-L (Ours)	23.76	**0.92**	**0.076**	15.49	**24.95**	**0.94**	**0.056**	**5.34**	**32.84**	**0.97**	**0.042**	**6.31**
w/o NGD	**23.95**	**0.92**	0.092	43.26	24.80	0.93	0.070	15.50	32.00	0.96	0.061	17.73

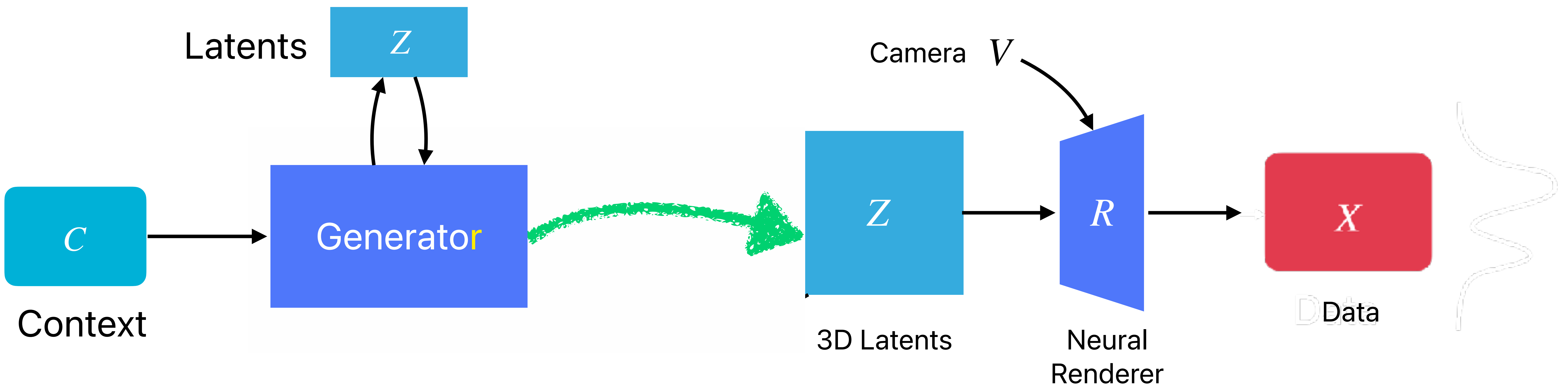
Gu, J., Trevithick, A., Lin, K. E., Susskind, J. M., Theobalt, C., Liu, L., & Ramamoorthi, R.,
"NerfDiff: Single-image View Synthesis with NeRF-guided Distillation from 3D-aware Diffusion," ICML 2023

How to learn?

- Approach I: Distillation from 2D Model

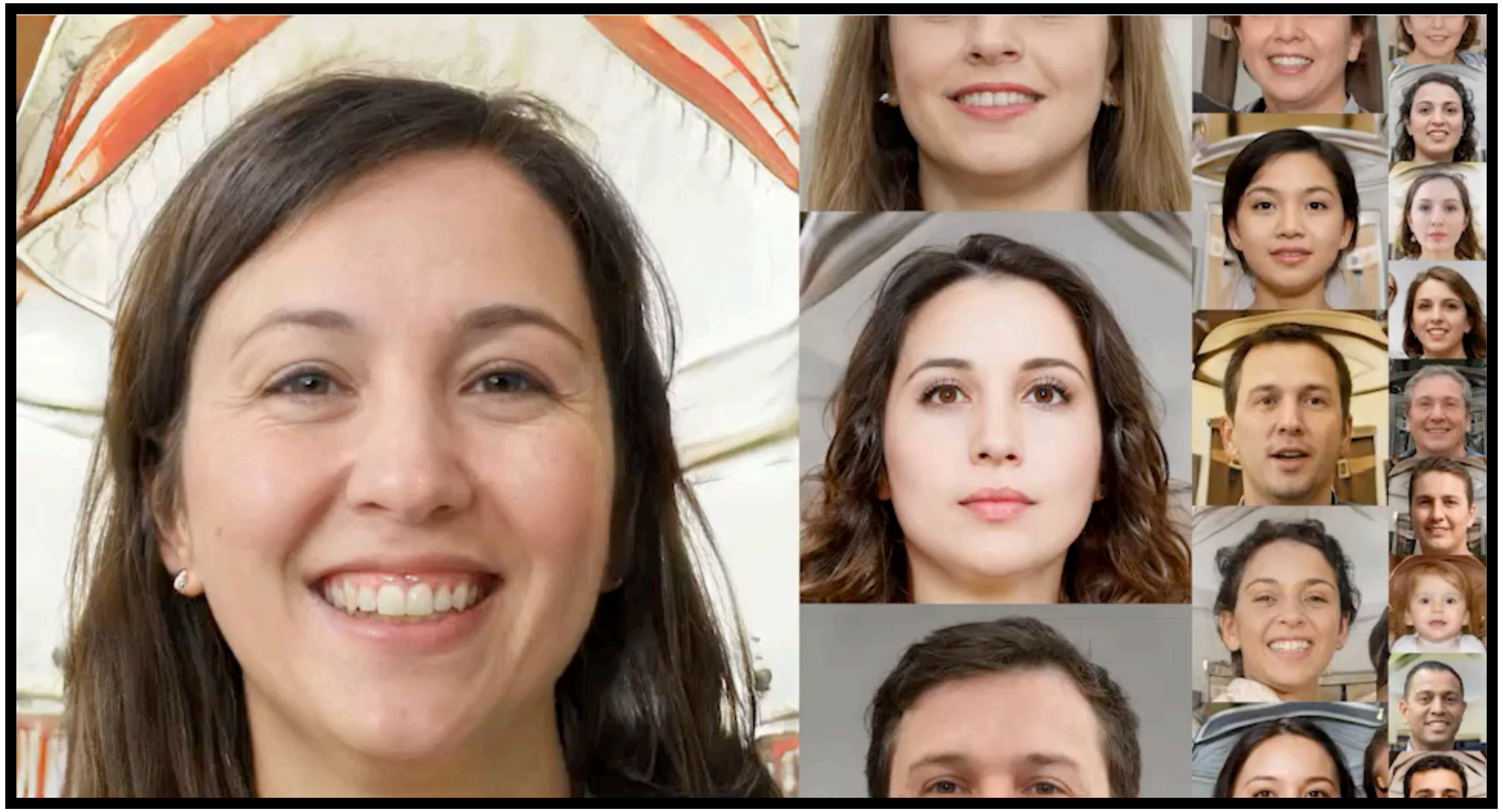


- Approach II: Direct 3D Generation



Direct 3D GANs

This is the first time a generative model can synthesize high-resolution images from novel views while preserving high 3D consistency!

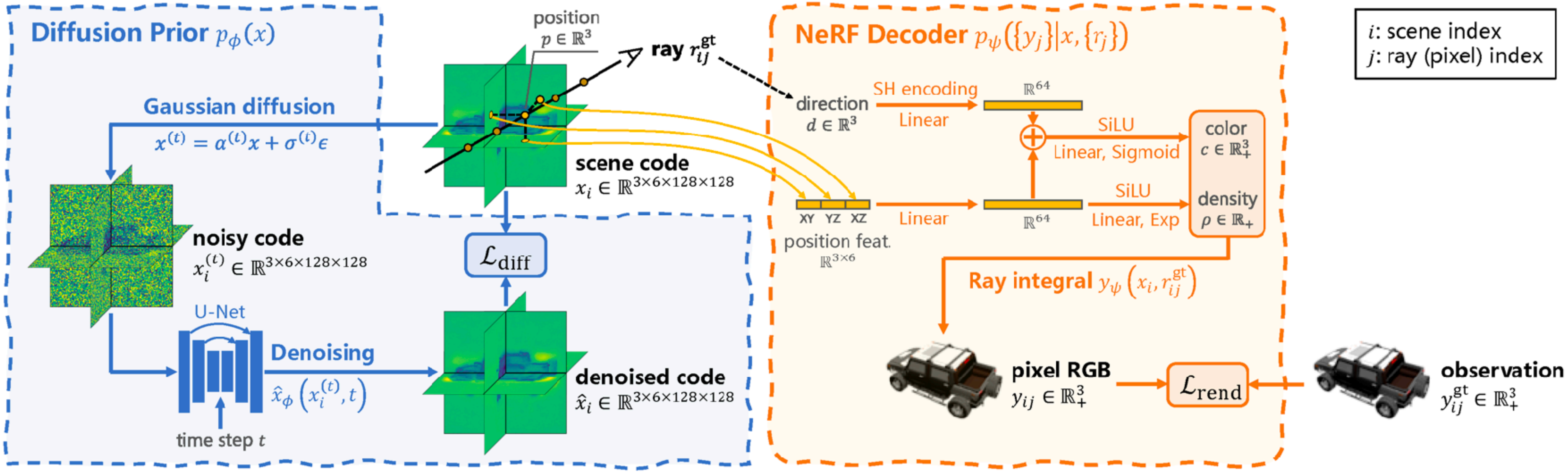


Our synthesized results (512x512)

Gu, J., Liu, L., Wang, P., & Theobalt, C.,
"Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis," ICLR 2022

Direct 3D Diffusion

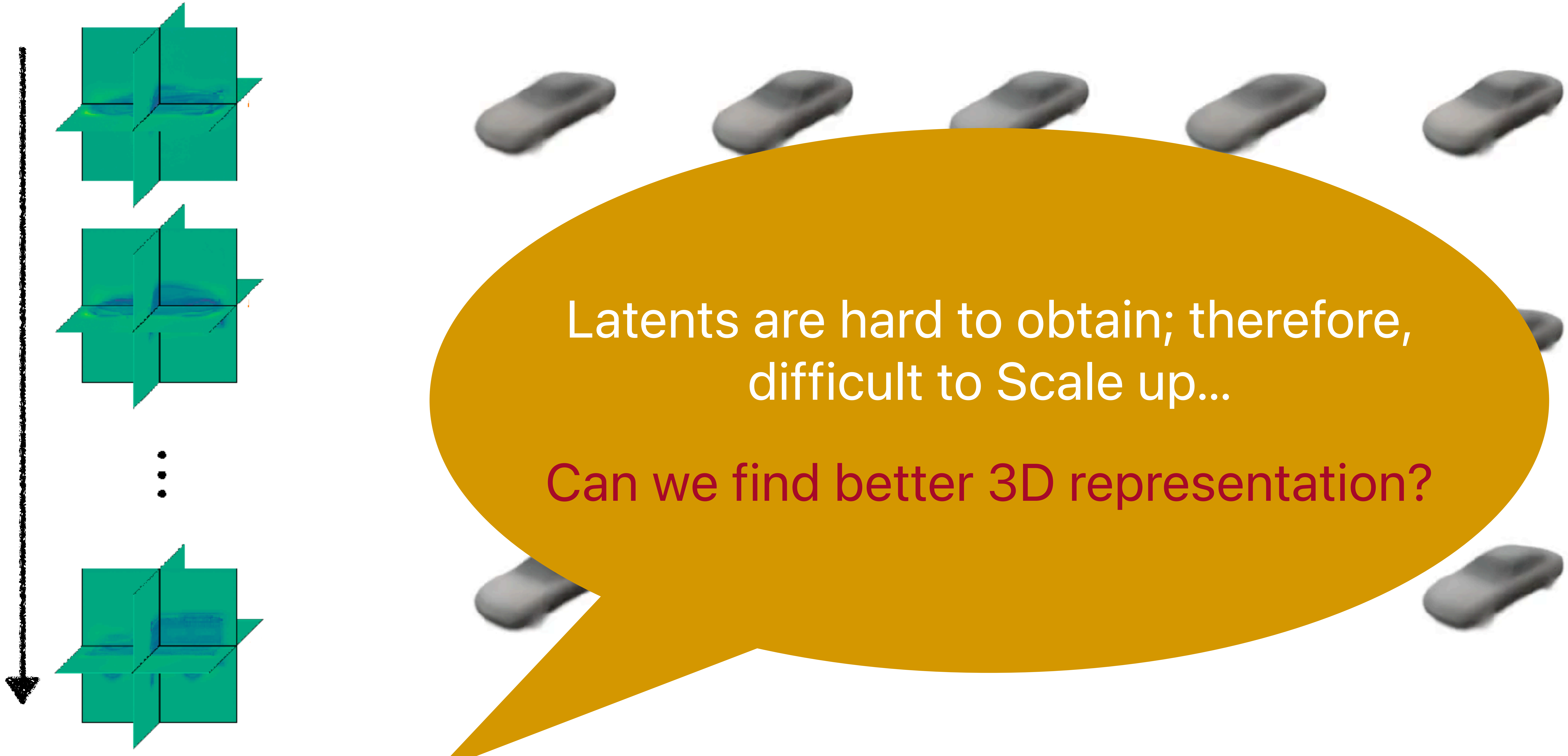
For each scene, we will simultaneously run 3D latents reconstruction and generative model learning on the optimized latents.



i: scene index
j: ray (pixel) index

Chen, H., ***Gu, J.***, Chen, A., Tian, W., Tu, Z., Liu, L., & Su,
"Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction," ICCV 2023

Progress of Generation



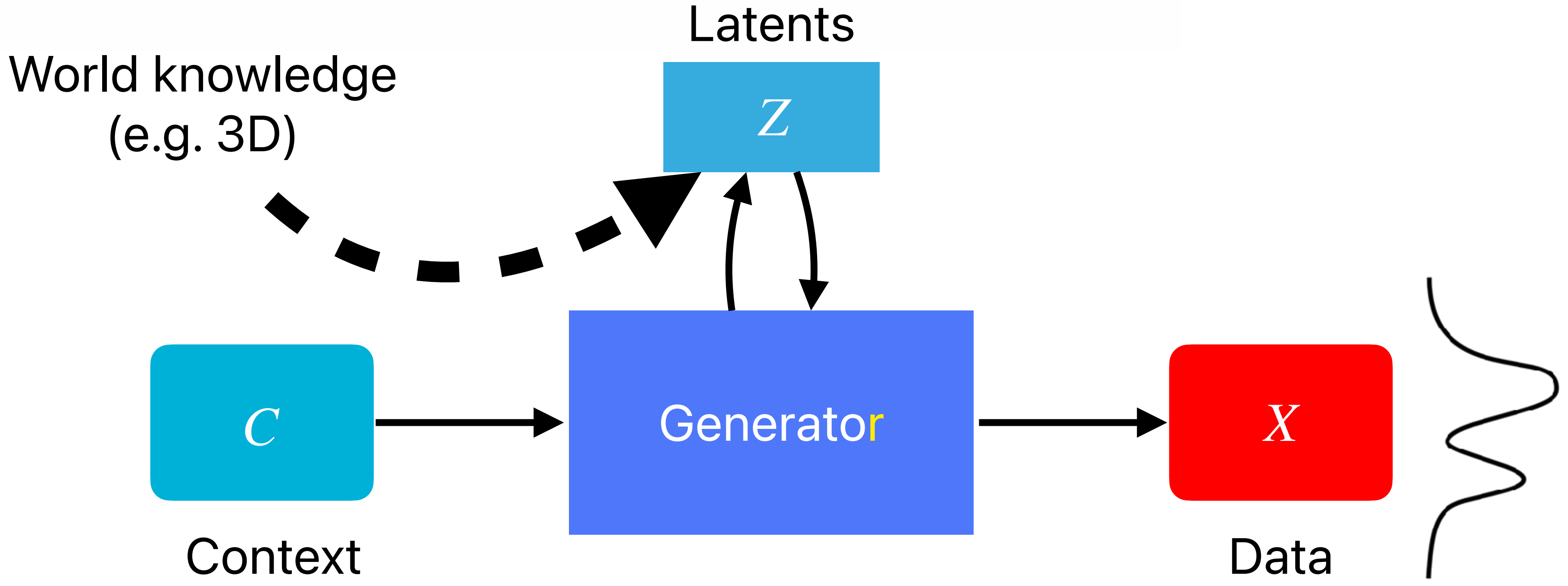
Latents are hard to obtain; therefore, difficult to Scale up…

Can we find better 3D representation?

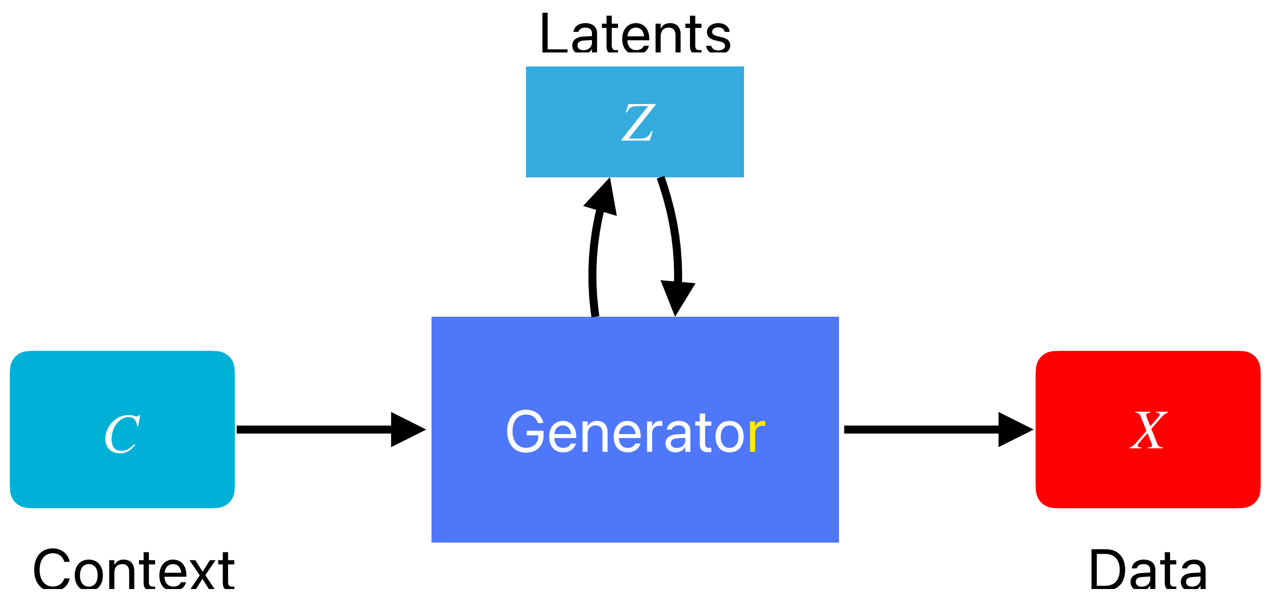
Chen, H., ***Gu, J.***, Chen, A., Tian, W., Tu, Z., Liu, L., & Su,
“Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction,” ICCV 2023

Takeaway

- Learning 3D latents allows for free-view synthesis in generative models.



To Summarize



Flexible

Scalable

Knowledgeable

Combine latents to design non-autoregressive generative models for flexible text generation.

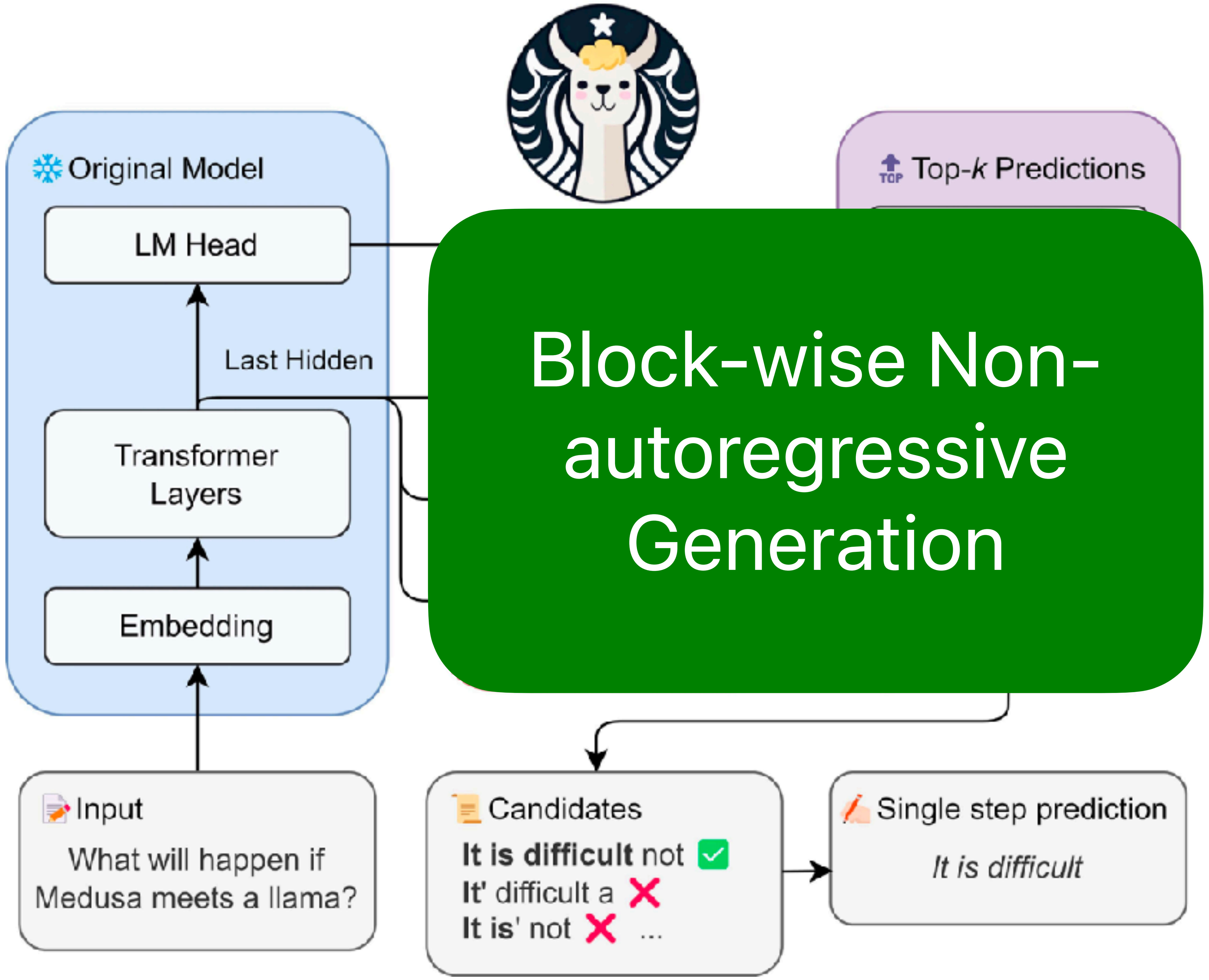
Integrate data structures into latent for high-resolution image and video synthesis.

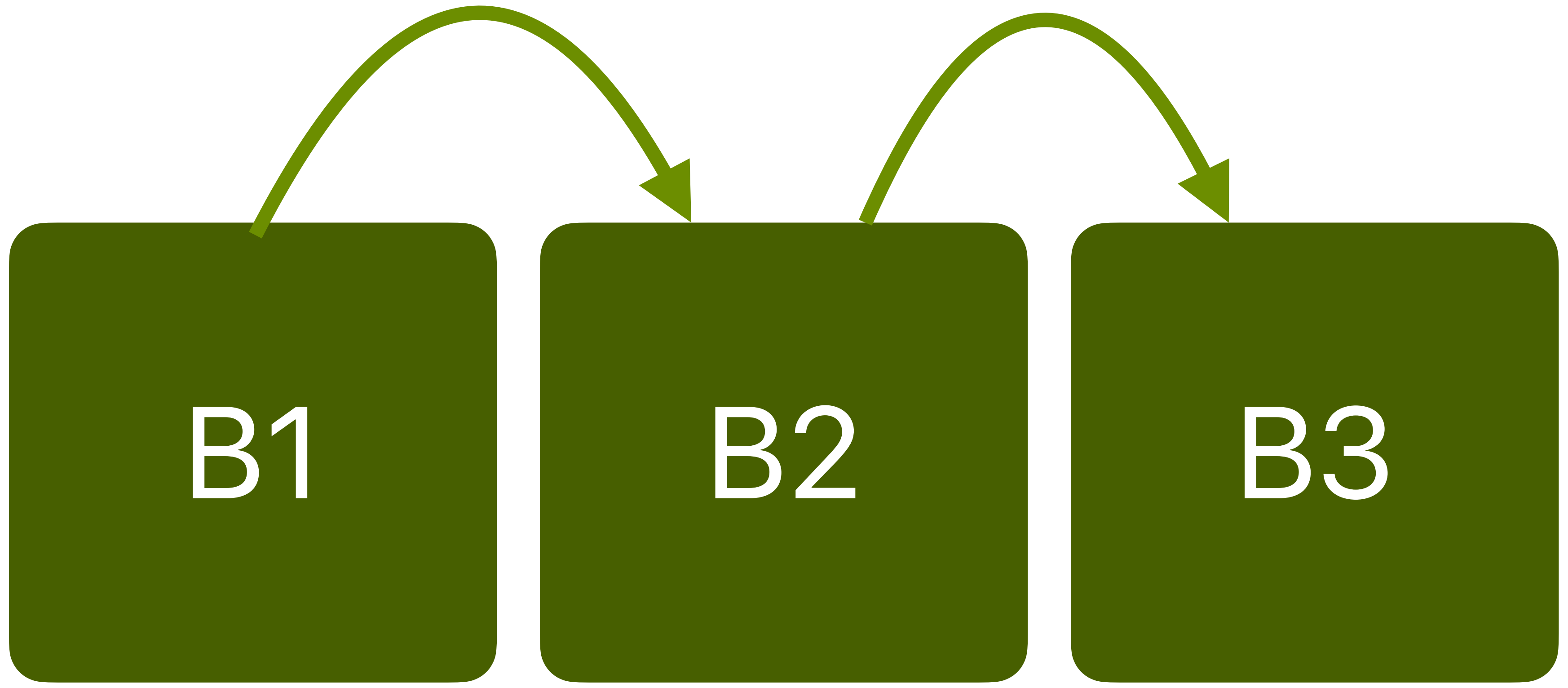
Model 3D knowledge as 3D latents in generative models for free-view synthesis.

Future Work

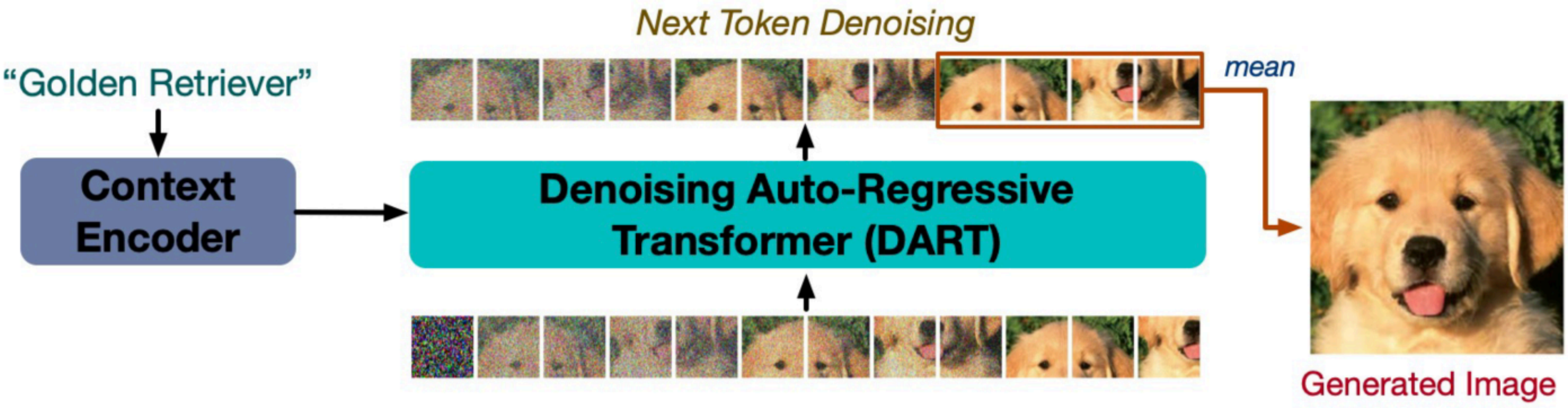
Flexibility of NAR for LLMs

Can we design more flexible large language models? For instance, apply NAR to fast generation and editing.





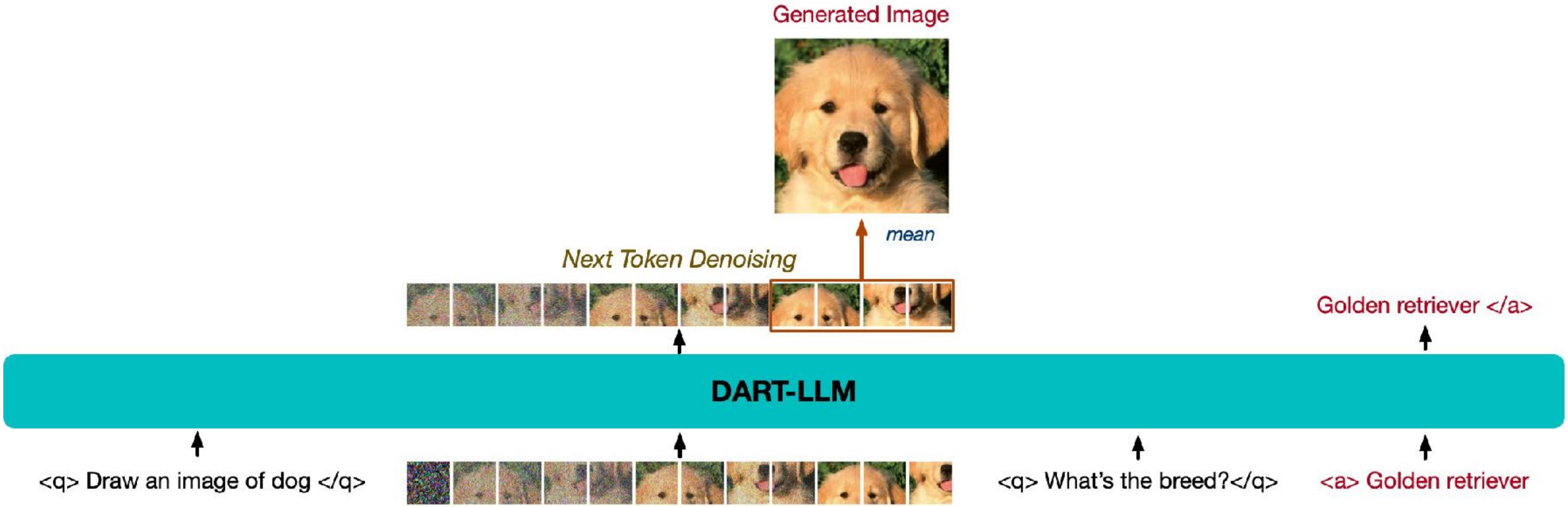
Scalable Learning: Unifying LLMs with Diffusion Models



$$\mathcal{L}^{\text{DART}} = \frac{1}{N}\sum_{n=1}^{N} \omega_n \| f_\theta(\boldsymbol{x}_{1:n-1}) - \bar{\boldsymbol{x}}_n \|_2^2$$

Work in progress (to be submitted to ICLR 2025)

Scalable Learning: Unifying LLMs with Diffusion Models



Work in progress

Physics-informed Generative AI

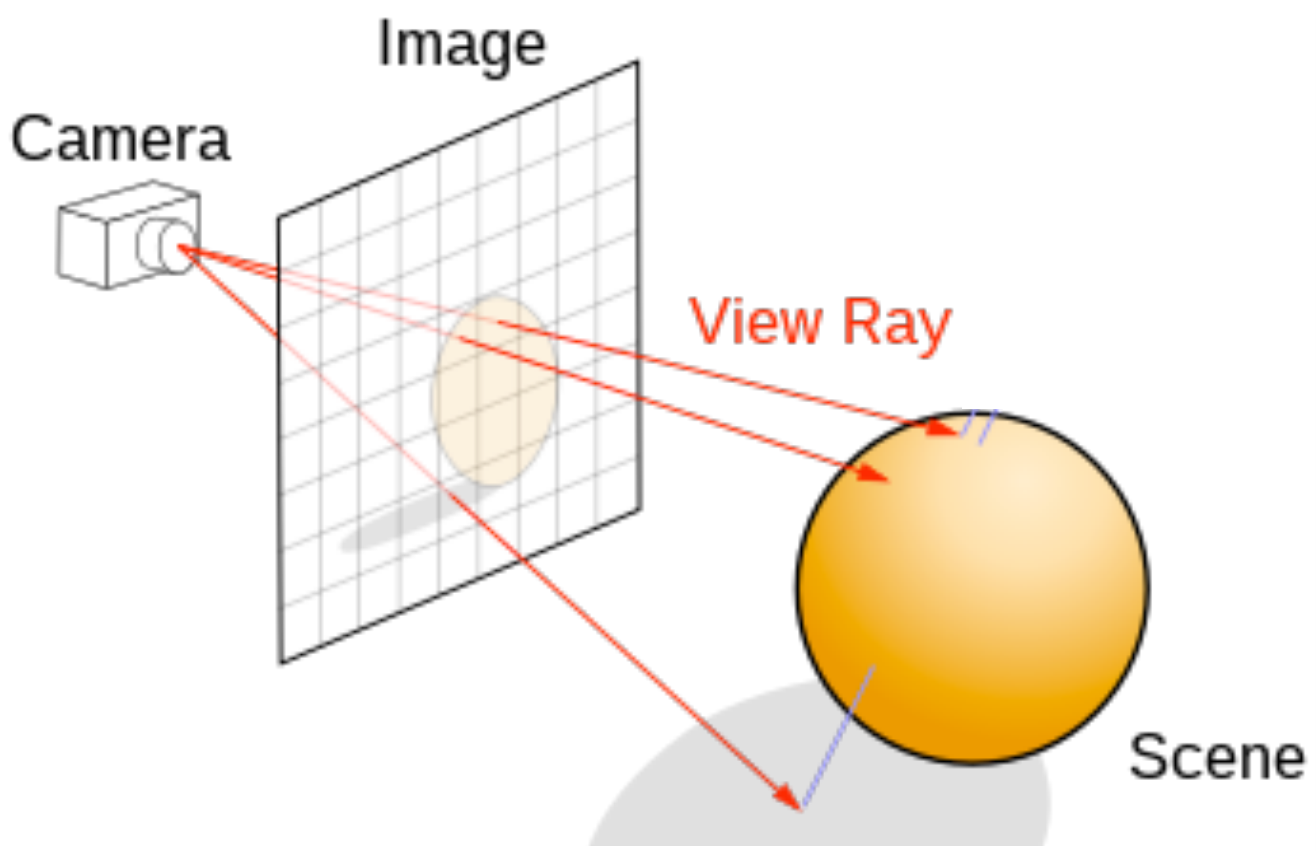
Can we incorporate more physics world knowledge?

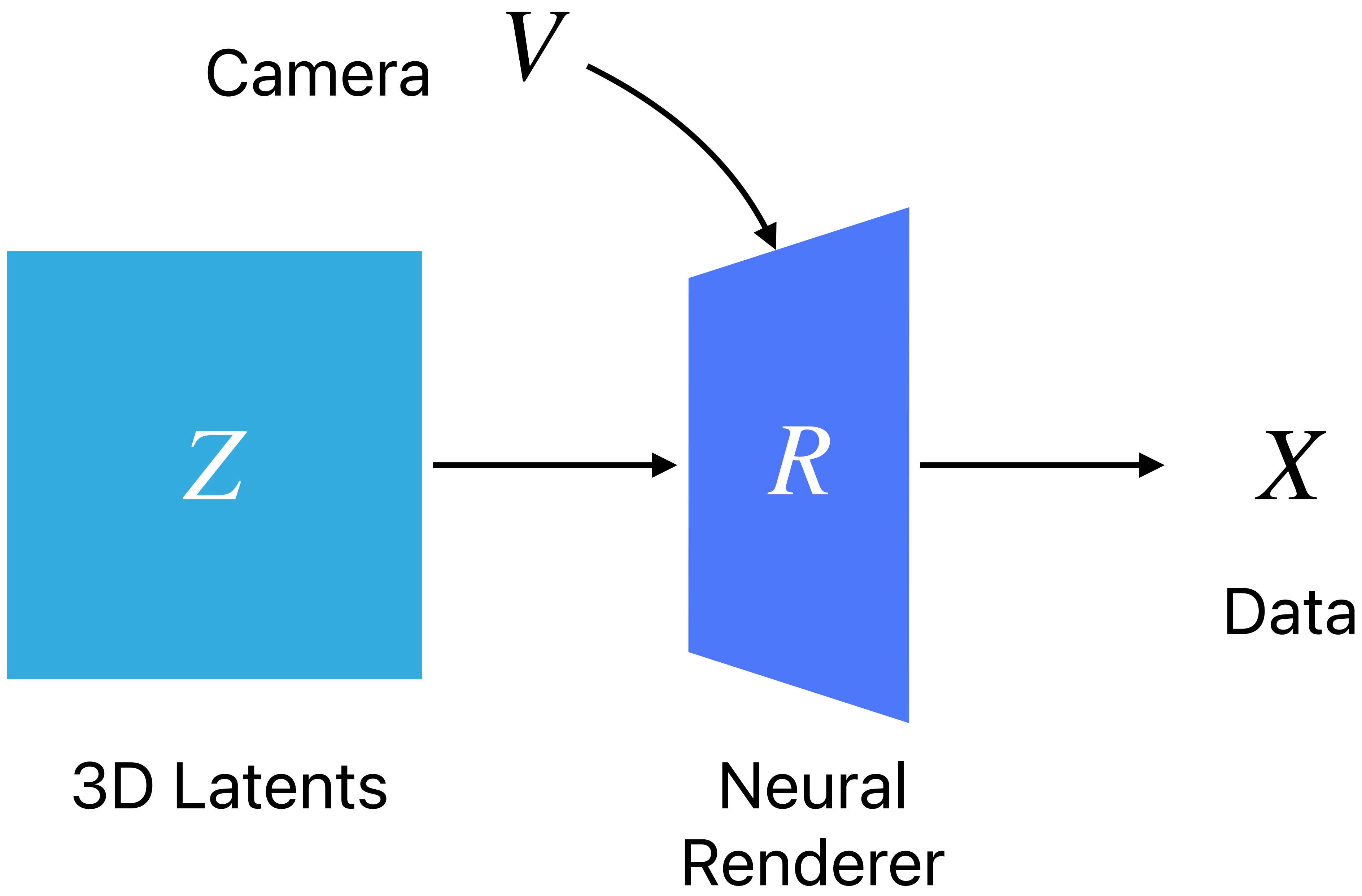
State-of-the-art Video Generation
(OpenAI Sora)

?

Physics-informed Generative AI

Can we take inspiration from 3D latents so far?



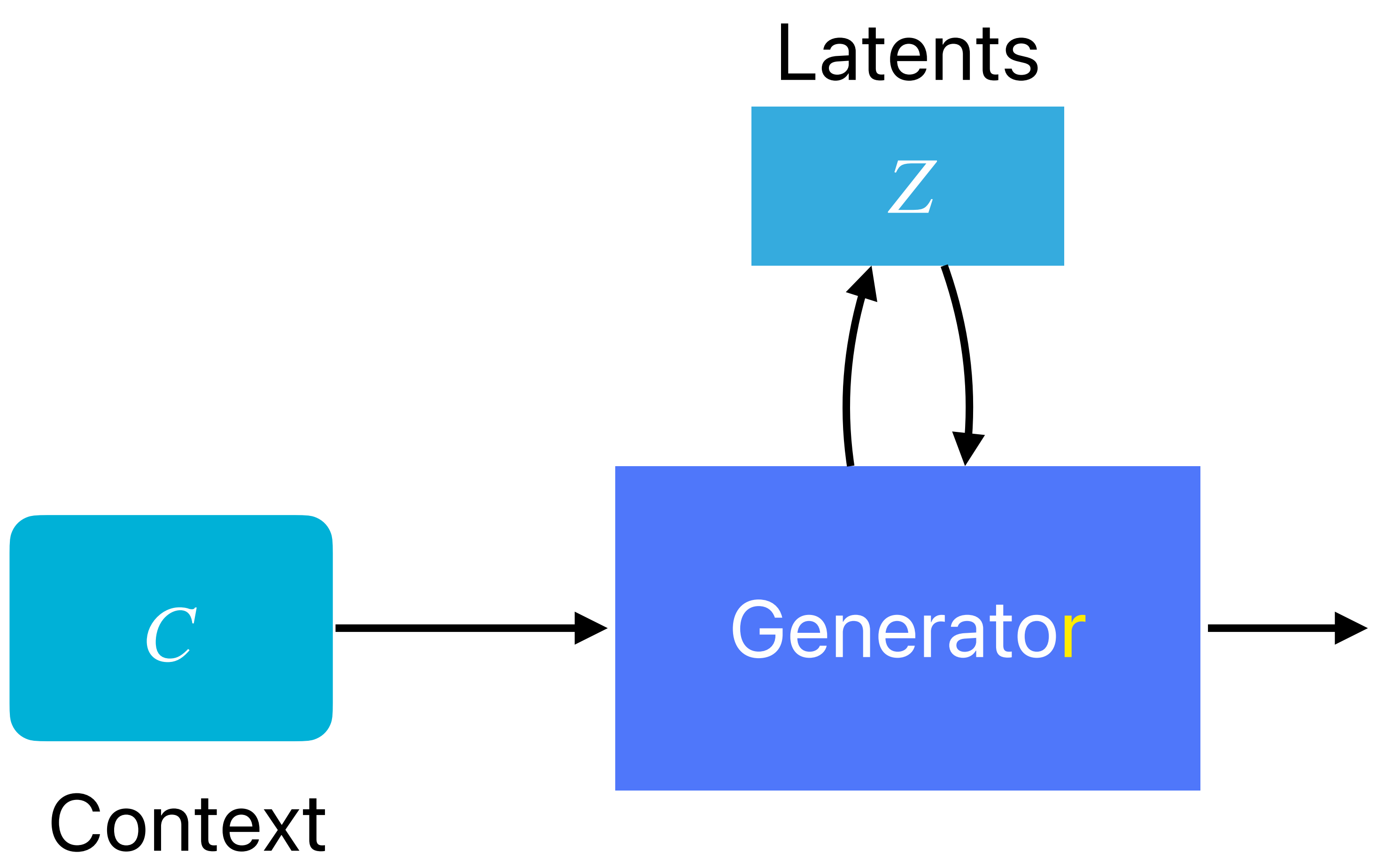


Physics simulator $\rightarrow f(t)$

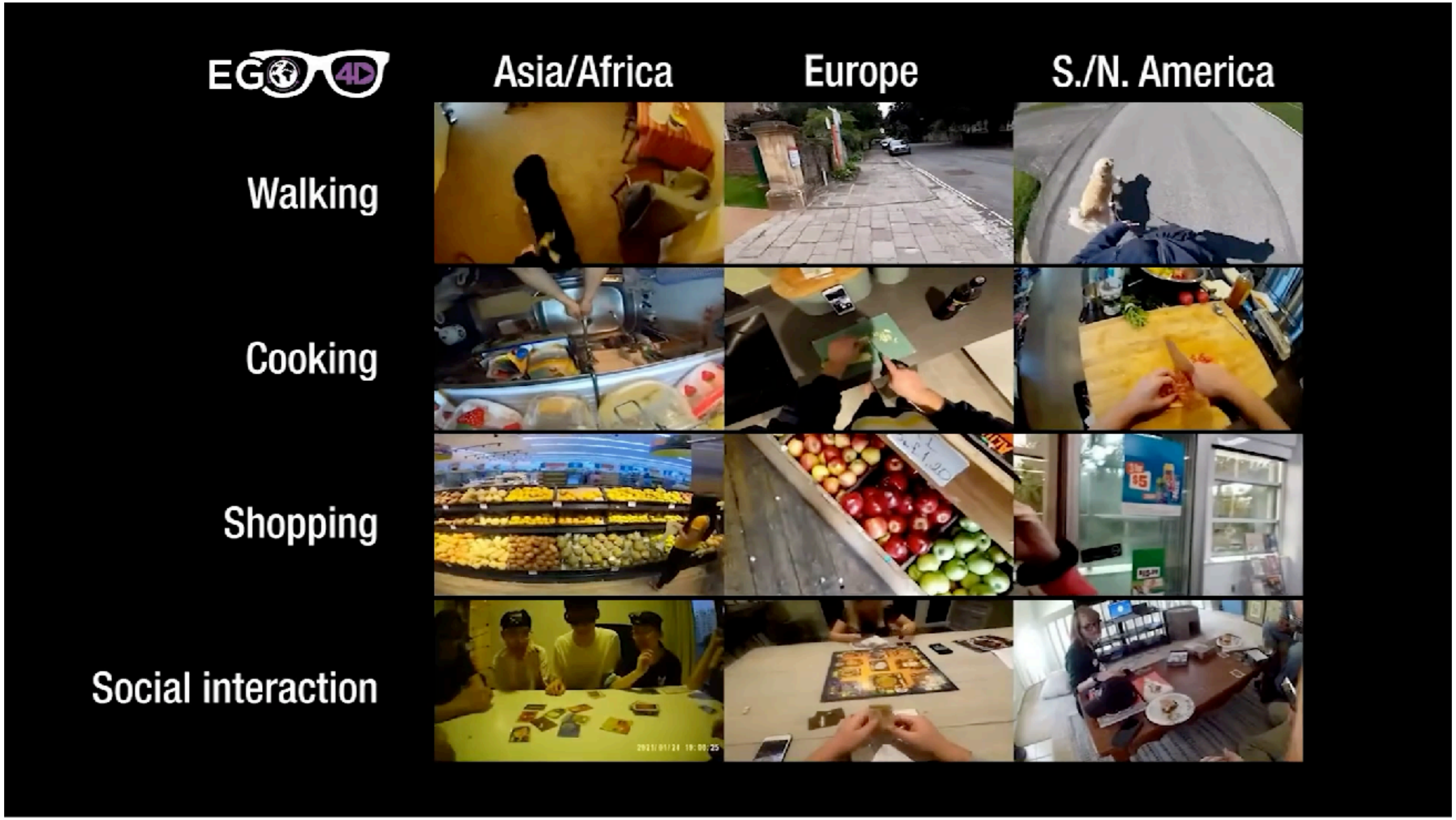
?

Generative AI for Embodied AI

Can we learn flexible, scalable, and knowledgeable generative models directly from large-scale ego-centric video data?



World Model for Embodied Systems



Generative AI for Applications

We can deploy such generative models for wider applications. For instance, creating high-quality and controllable synthetic training datasets.

Self-driving

Robotics

Medical Imaging

A sample of 1024x1024 Generation from “*Matryoshka Diffusion Models*”, ICLR 2024