Stable Diffusion and Paper Reading

DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation

Junzhe Yi

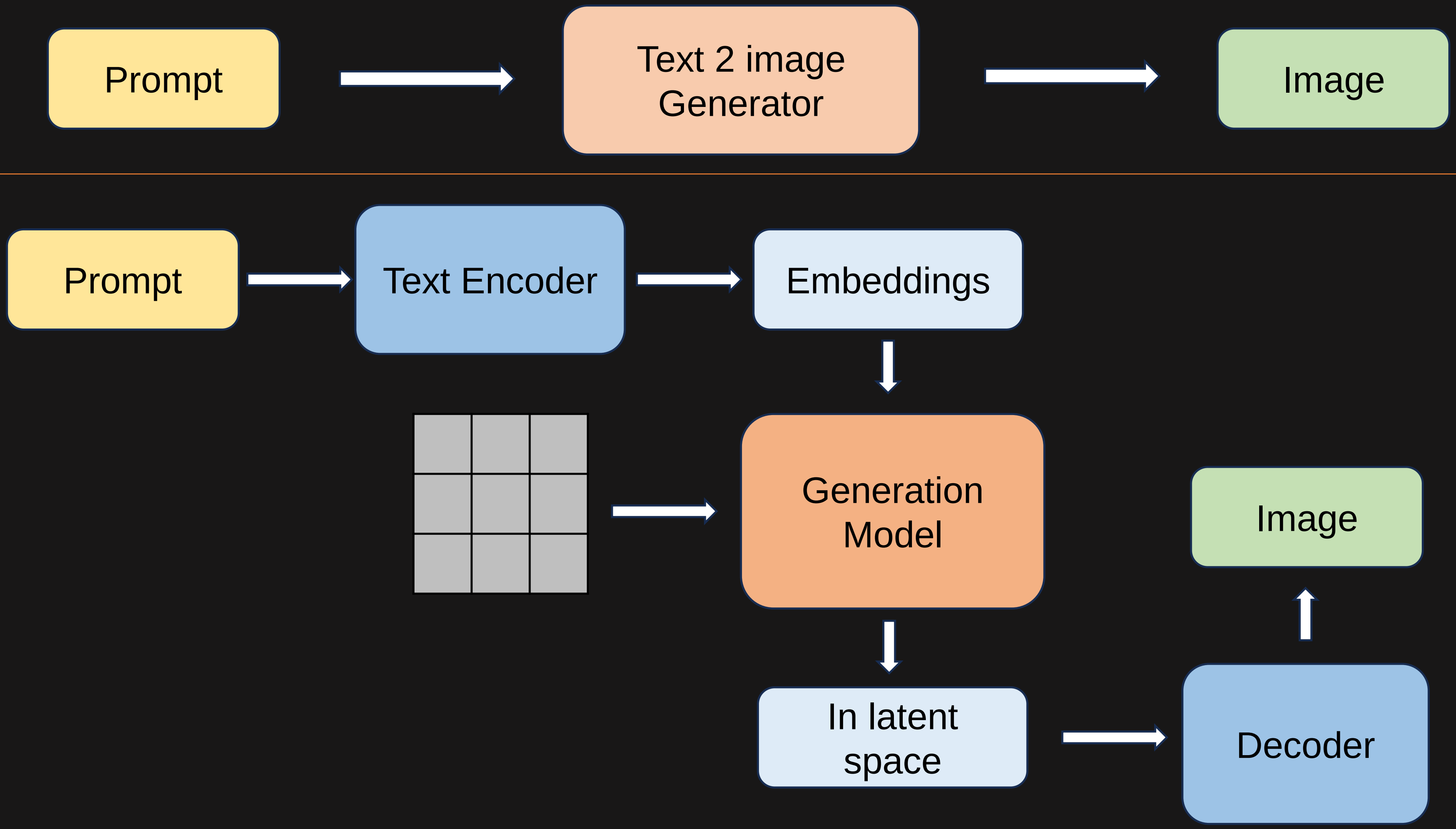
HuazhongUST

April 27, 2024

Summary

1. Framework of stable diffusion
2. Understanding how diffusion model works
3. Fine tuning: Dreambooth
4. Methodology Overview
5. Reference

Framework of Diffusion Models



Framework of Stable Diffusion

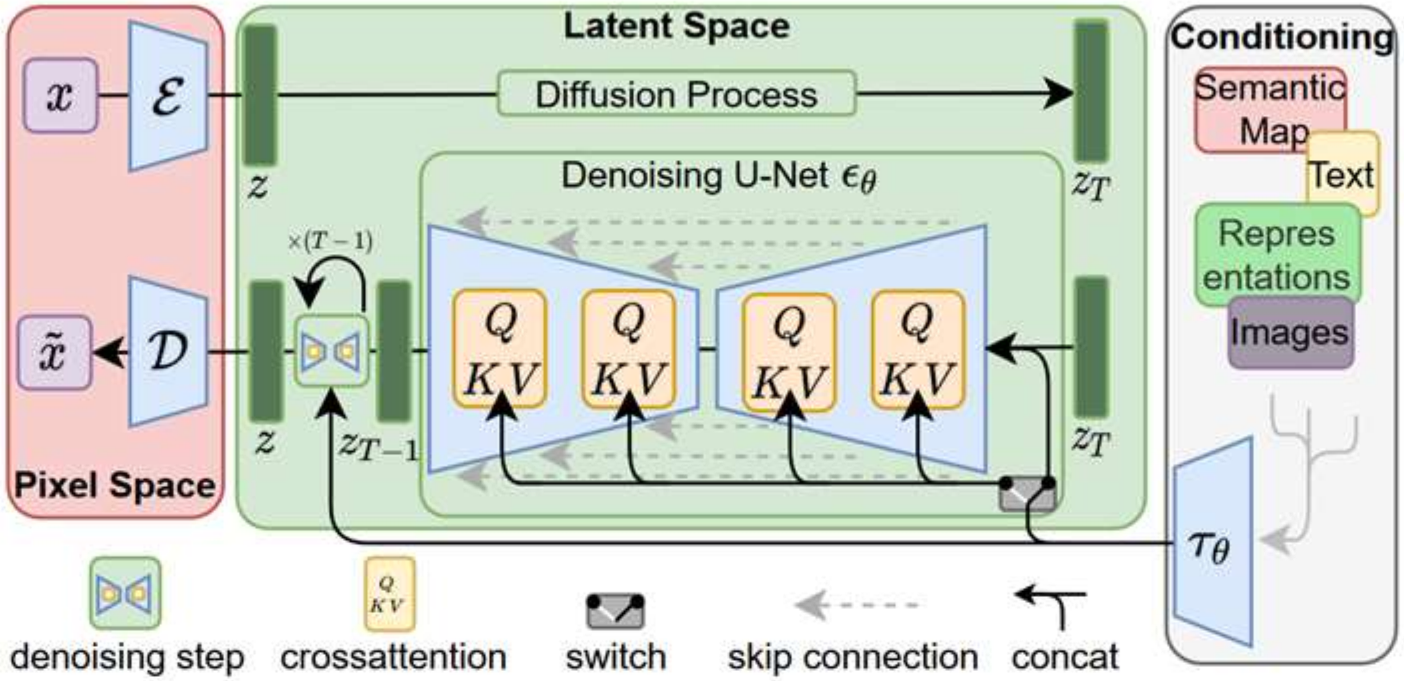
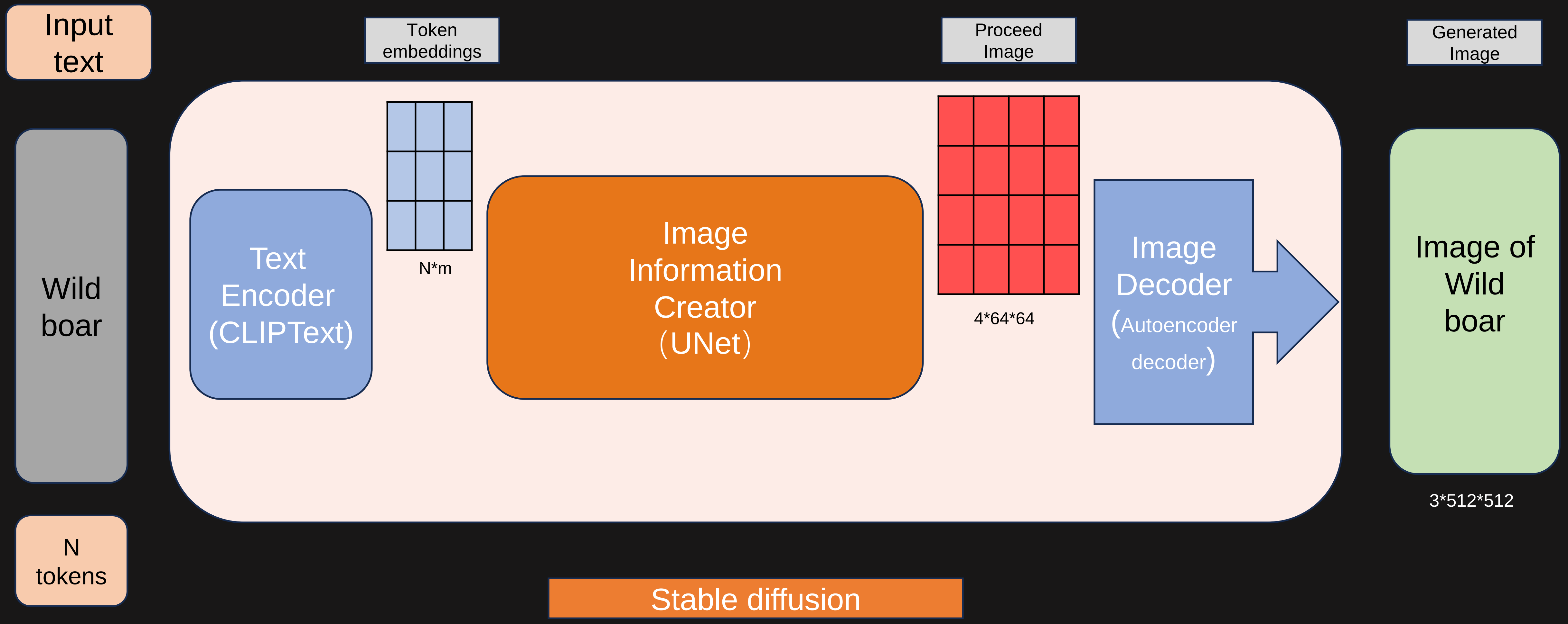


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

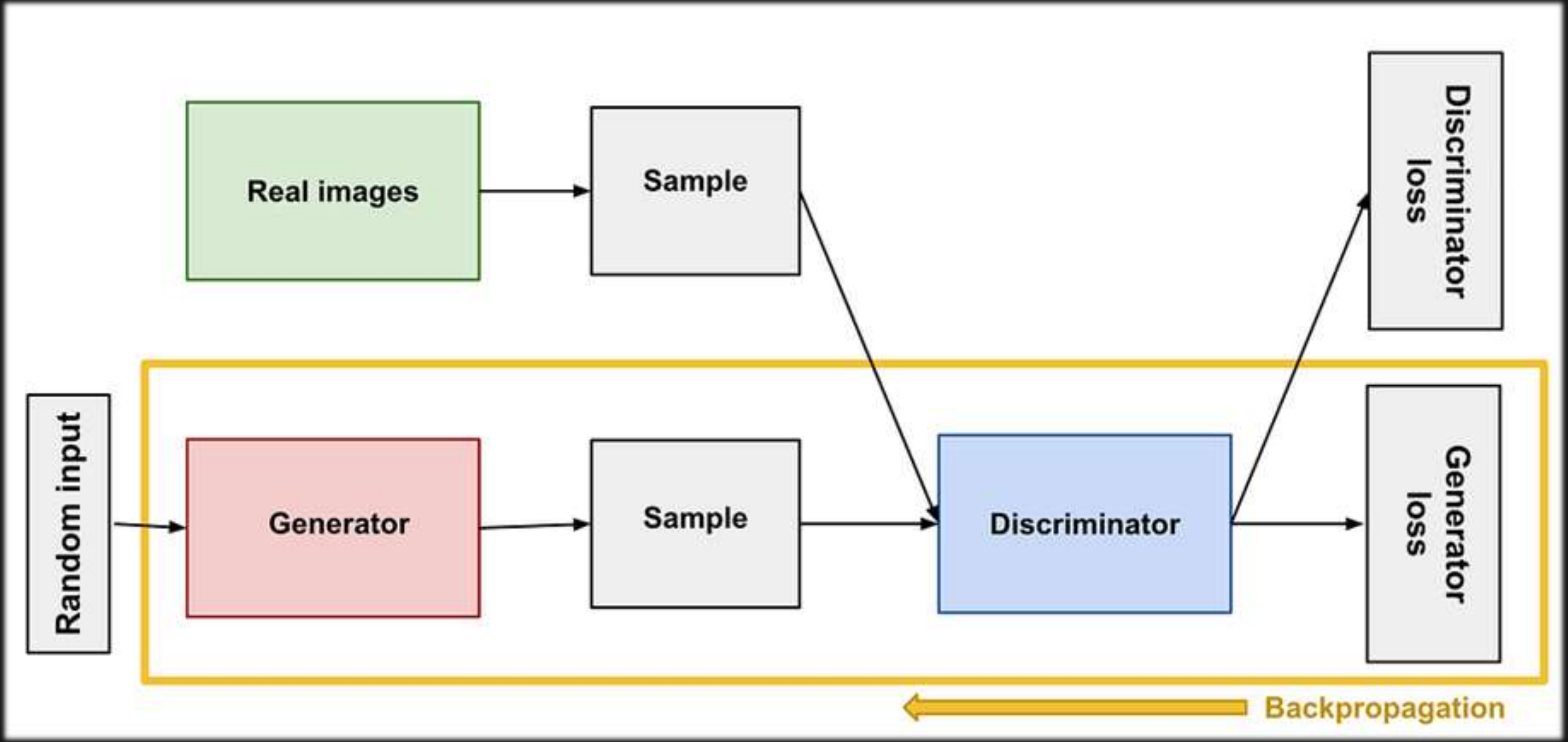
High-Resolution Image Synthesis With Latent Diffusion Models (2022 CVPR)
https://arxiv.org/abs/2112.10752

Framework of Stable Diffusion



How Image Creator Work?

Generative adversarial network (GAN)



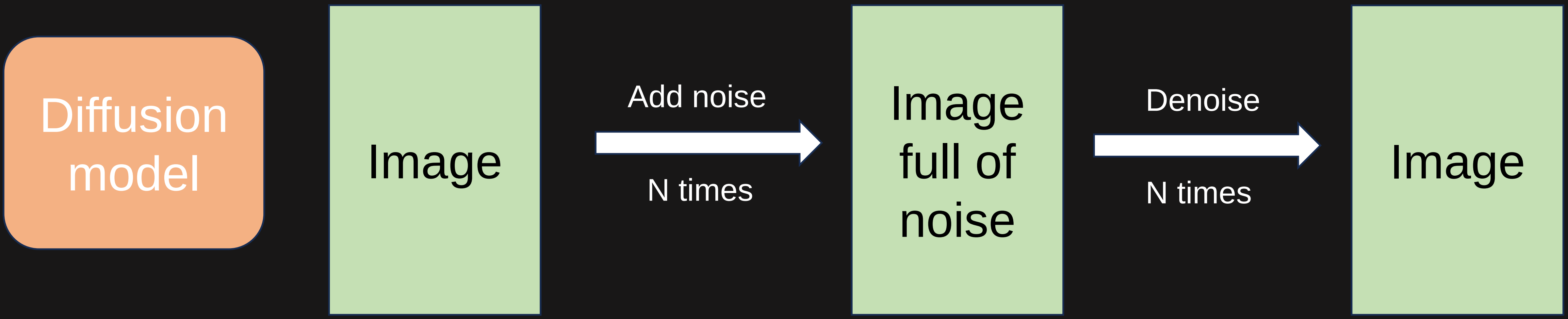
Generative Adversarial Nets
https://arxiv.org/abs/1406.2661

DDPM

Diffusion model

DDPM（NeurIPS2020）

They work by gradually adding Gaussian noise to the original data in the forward diffusion process and then learning to remove the noise in the reverse diffusion process.



Denoising Diffusion Probabilistic Models
https://arxiv.org/abs/2006.11239

Two Step Process

A denoising diffusion modeling is a two step process:

- Forward diffusion process — The forward diffusion process is the Markov chain of diffusion steps in which we slowly and randomly add noise to the original data.
- Reverse diffusion process — The reverse diffusion process tries to reverse the diffusion process to generate original data from the noise.

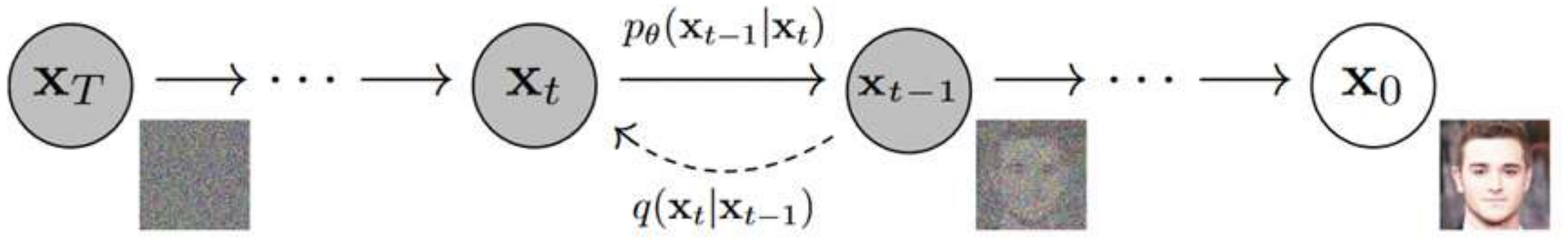


Figure 2: The directed graphical model considered in this work.

Algorithm

Algorithm 1 Training

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

Algorithm 2 Sampling

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

Understanding Diffusion in Mathematical Detail

Markov chain: $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$ $\quad x_{t-1} = \sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}$

We can arrive $x_t = \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1}$

That is $x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}}\bar{\epsilon}_{t-2}$

Conclusion: $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$

Forward diffusion process finished.

Using Bayes' rule, we have: $q(x_{t-1}|x_t, x_0) = \dfrac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$

Given the condition of knowing X_t, find X_{t-1}, This probability follows a normal distribution

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0$$

Understanding Diffusion in Mathematical detail

The mean of this normal distribution is correlated with X_t and X_0. This is going to be difficult now, because there is a unknown X_0. However If we go back to the previous page, we will find that x_0 has already been calculated in the forward process.

so we can estimate X_0 as: $$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t)$$

The mean: $$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right)$$

In this equation, there is only one unknown, which is: ϵ_t

the next step: to estimate the noise!

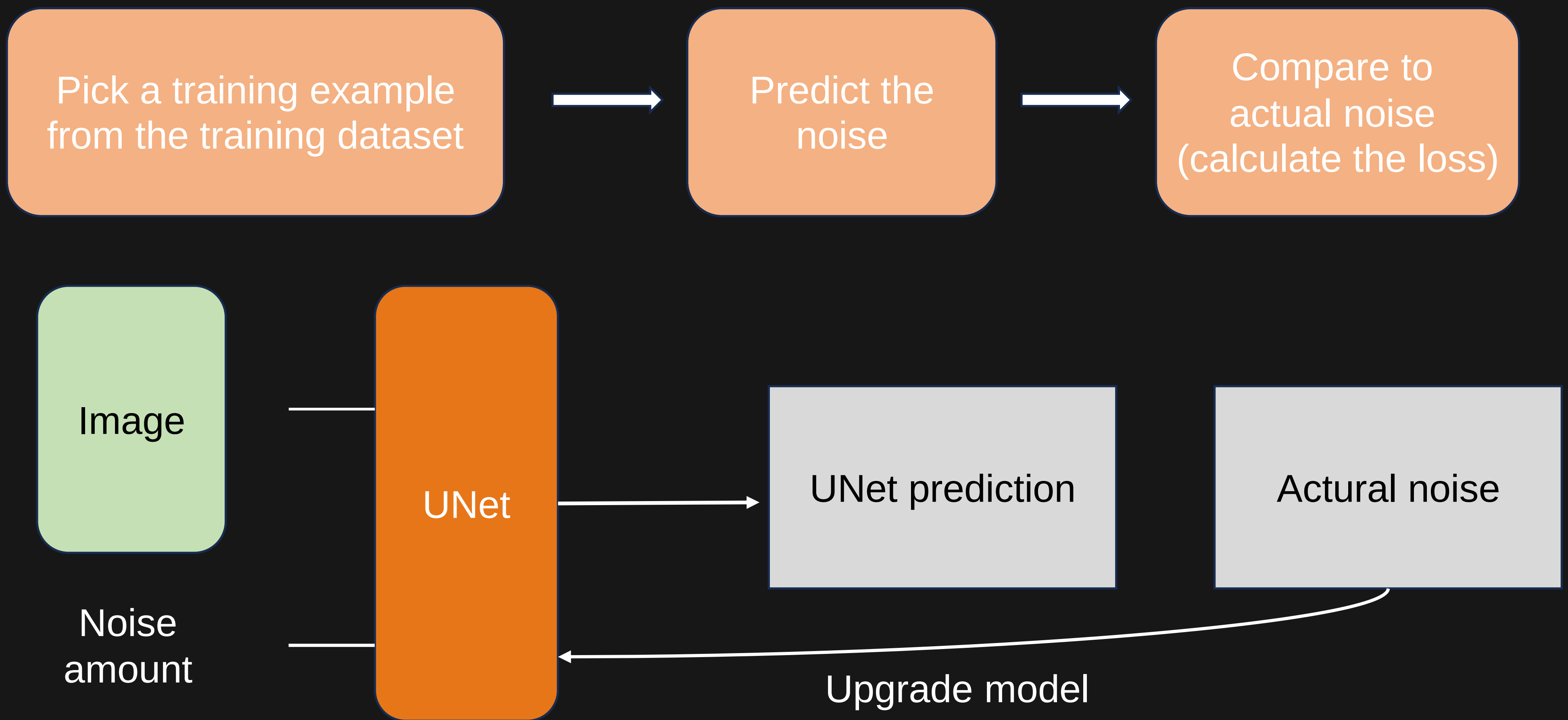
Usually we use ‘U-Net + transformer’. Trained on such a massive dataset, the powerful noise predictor U-Net has the "capability" to iteratively denoise a noisy image during the reverse process of diffusion, transforming it into a perfect image.

Why not have a look into U-Net?

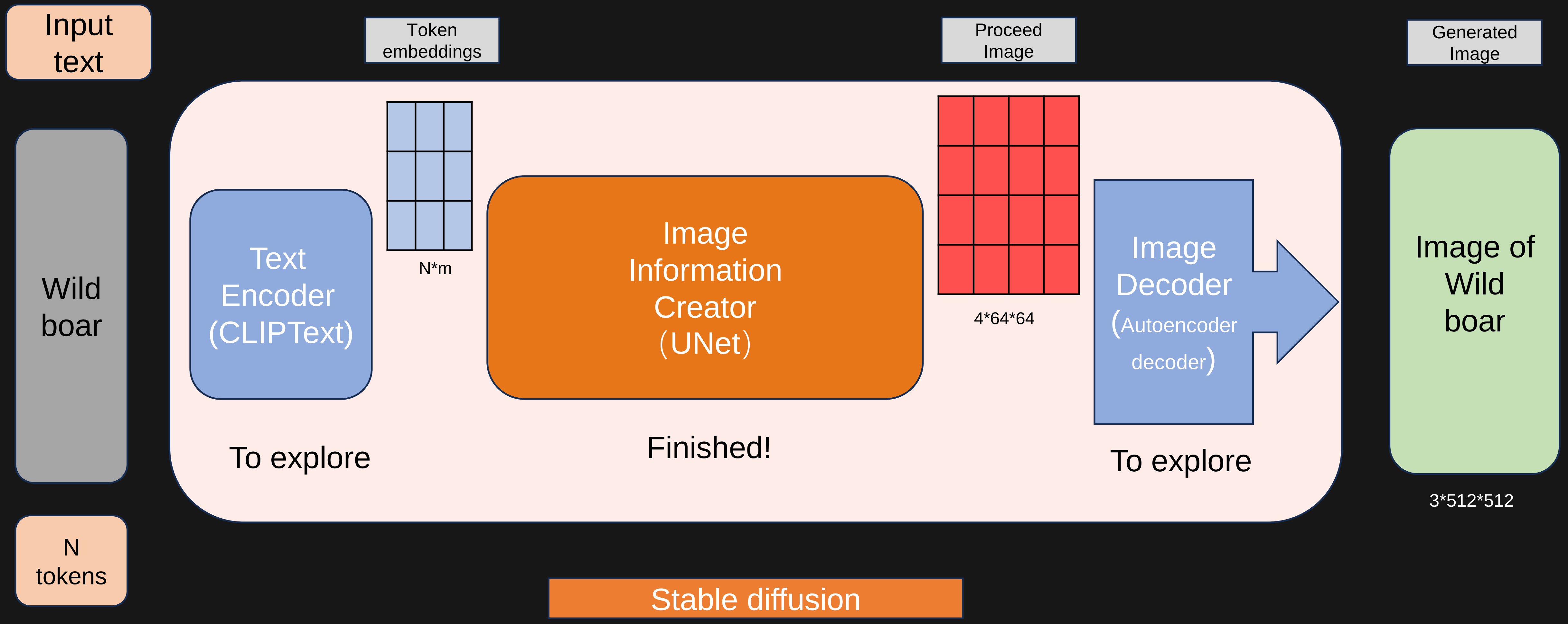
UNet Train Steps

U-Net train step:

supervised learning!



Take a Look Back



AutoEncoder Decoder

To understand AutoEncoder Decoder, you need to know “latent space” as prior knowledge. if you don’t, please refer to the LDM, the paper’s link is given in Page 5 of this Powerpoint.

In order to discover the latent connections and patterns between images and reduce computational complexity, the operation of Stable Diffusion is not conducted directly on the pixel dimensions of the images themselves but rather in the compressed version of the images, known as the "latent space." This compression and decompression process is achieved through an Autoencoder Decoder.

AutoEncoder Decoder

Departure to Latent Space

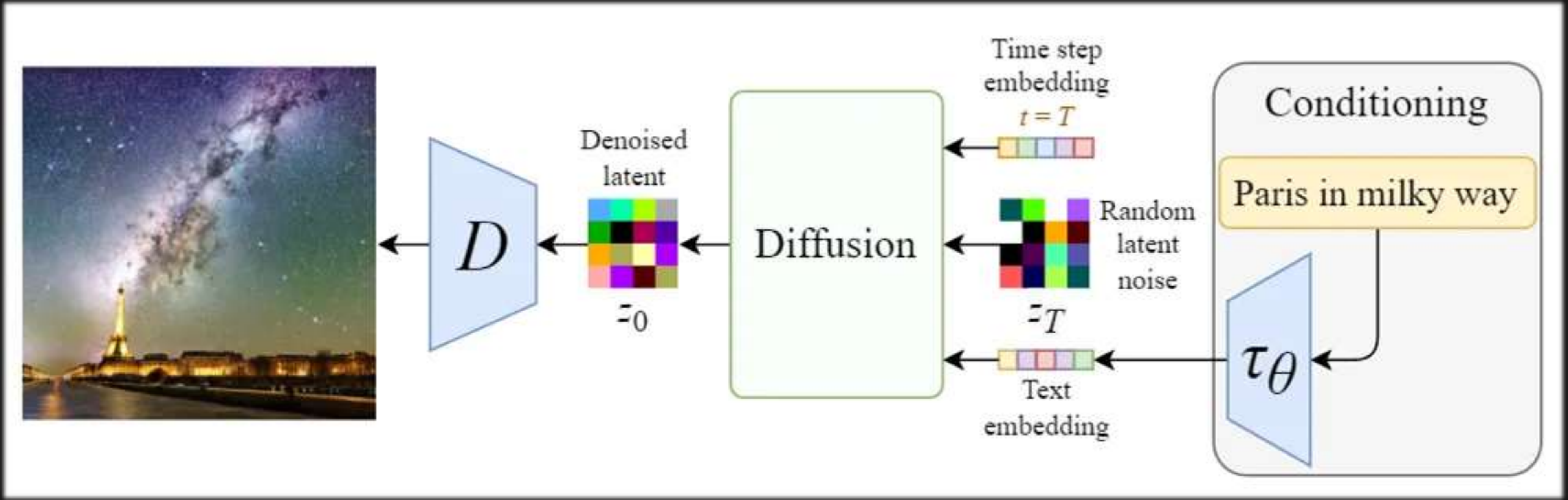
- By using the trained encoder E, we can encode the full-sized image into lower dimensional latent data (compressed data).
- By using the trained decoder D, we can decode the latent data back into an image.

Latent Diffusion

- After encoding the images into latent data, the forward and reverse diffusion processes will be done in the latent space.

Example

An example



CLIPtext (Contrastive Language-Image Pre-Training)

At this point, let us introduce the most powerful module of Stable Diffusion.

If you don't have CLIPtext, you won't be able to use textual prompts to control the semantic content of the output image. In that case, Diffusion will generate images randomly.

CLIPtext is a Text Encoder, represented by the deep blue module in the diagram. It is a special type of Transformer-based natural language model. It takes the input text prompts and generates the Token embeddings matrix.

You can understand it as a conditioning mechanism.

Conditioning of Stable Diffusion

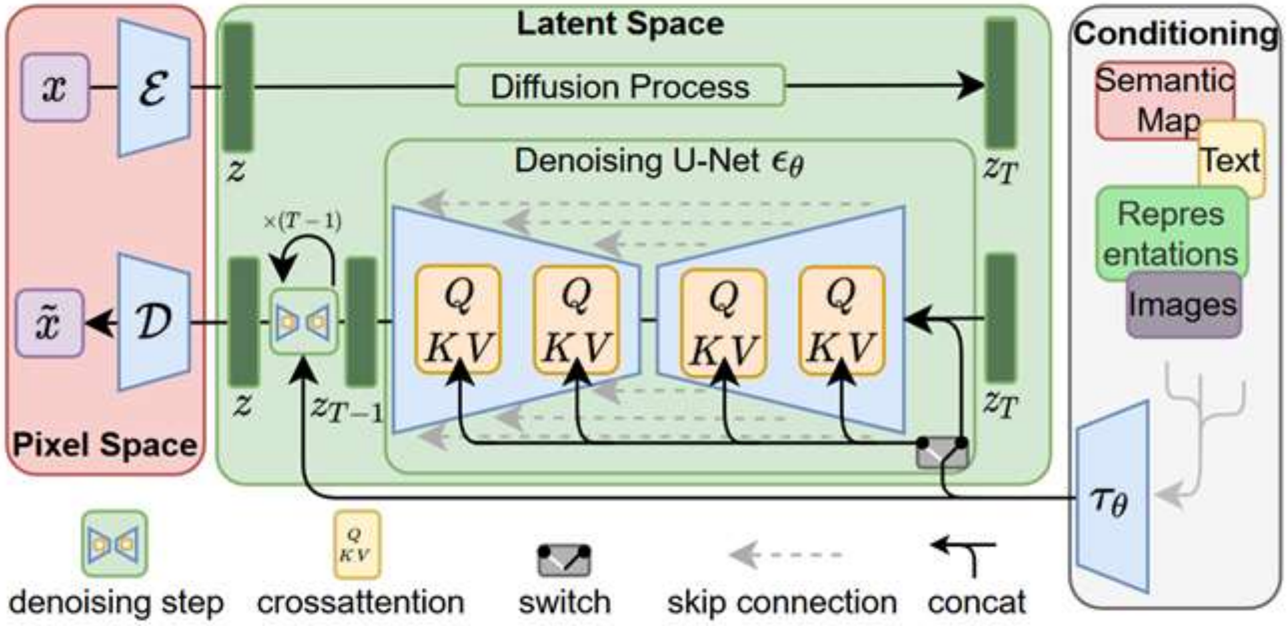


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

This is done by modifying the inner diffusion model to accept conditioning inputs.

Conditioning of Stable Diffusion

The inner diffusion model is turned into a conditional image generator by augmenting its denoising U-Net with the cross-attention mechanism.

The switch in the above diagram is used to control between different types of inputs:

- For text inputs, they are first converted into embeddings (vectors) using a language model $\tau\theta$ (e.g. BERT, CLIP), and then they are mapped into the U-Net via the (multi-head) Attention(Q, K, V) layer.
- For other spatially aligned inputs (e.g. semantic maps, images, inpainting), the conditioning can be done using concatenation.

DreamBooth

The current text-to-image generation models are capable of generating high-quality images based on given prompts. However, these models are unable to mimic the appearance of objects in a given reference image and generate new images in different scenes.

Until DreamBooth was introduced.

DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation

Nataniel Ruiz[*,1,2] Yuanzhen Li[1] Varun Jampani[1]
Yael Pritch[1] Michael Rubinstein[1] Kfir Aberman[1]
[1] Google Research [2] Boston University

Identified Gaps

- Most text-to-image generation models are unable to perform few-shot learning
  - There is no efficient method to “inject” subjects into the model training
    - Overfitting
    - Language Drift
    - Best ex.: GAN reproducing the same face given ~100 images[1]

[1] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019

Contributions

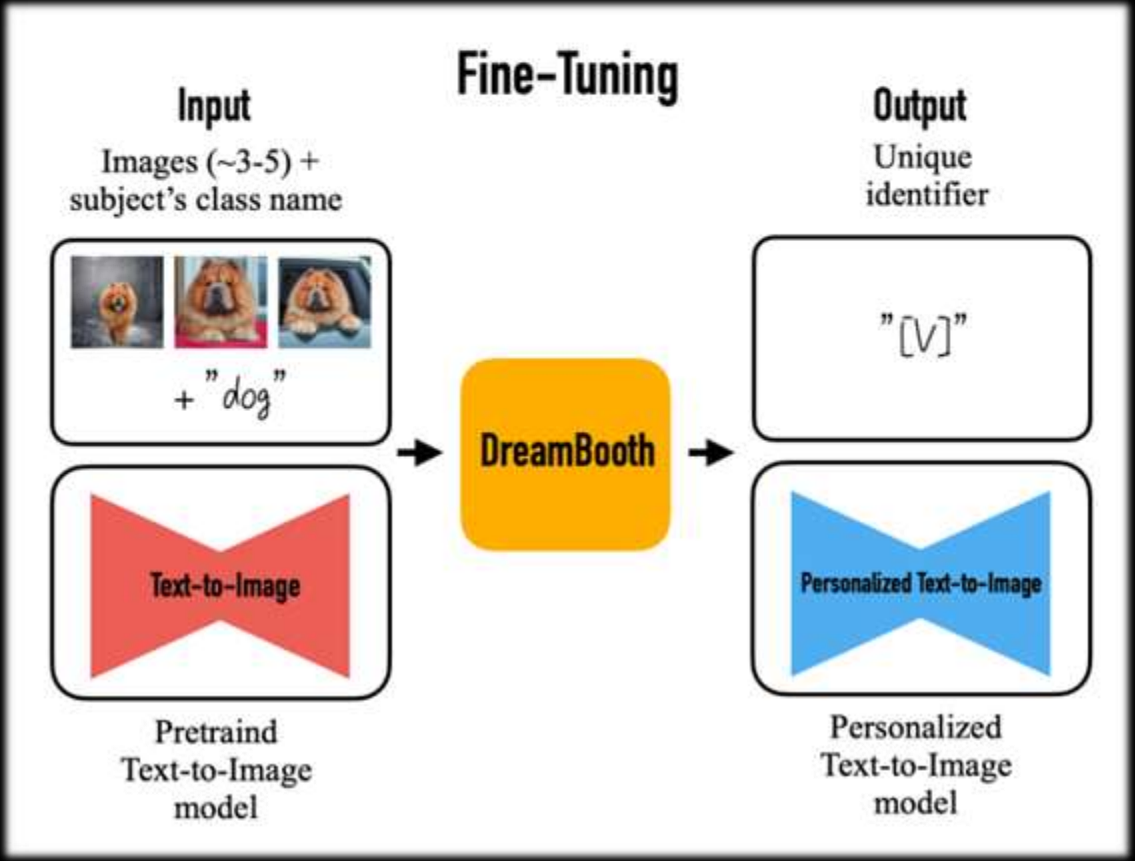
- New efficient few-shot fine-tuning technique that preserves semantic class knowledge
    - Tuning requires only 3~5 examples
- Created new problem to explore diffusion model capabilities-subject-driven generation
    - Goal: maintain fidelity in new contexts

Improvements over Existing Work

Subject-Driven Generation (Personalization)

Methodology Overview

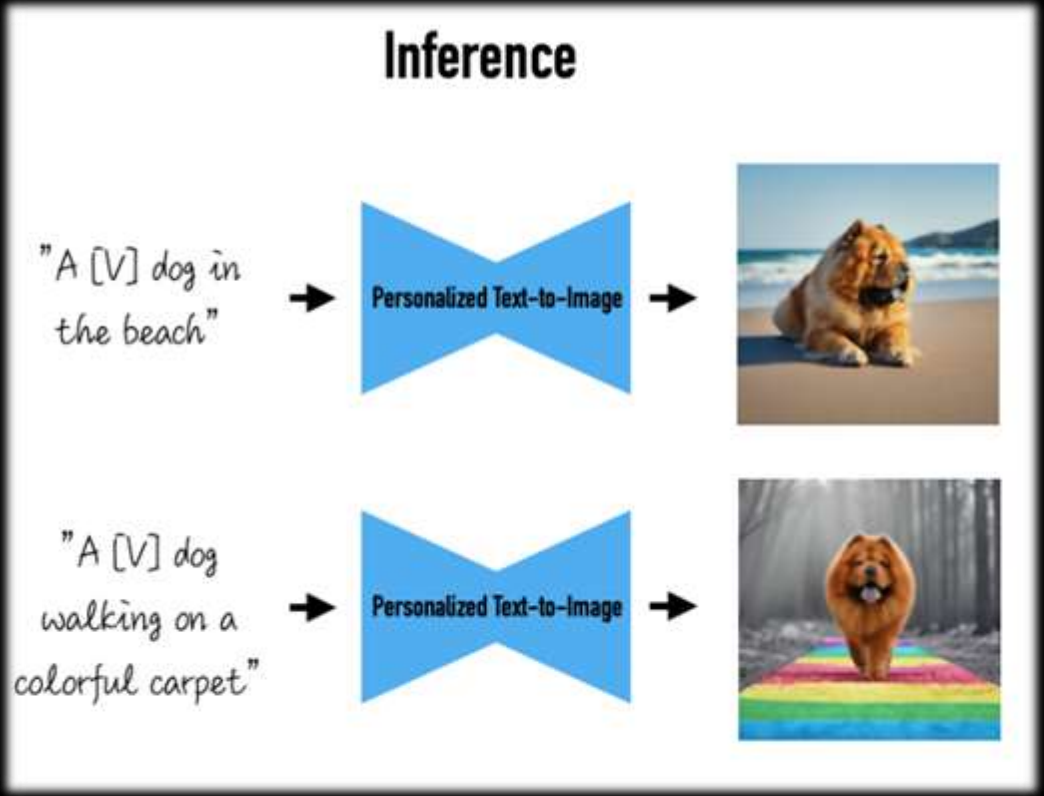
- Few-shot text-guided diffusion
  - ➢ Fine-tune existing diffusion models



Rare-Token Identifier

? There is no efficient method to “inject” subjects into the model training

- How to refer to a new subject?
    - ➢ Image is straightforward
- Diffusion model already has a ‘vocabulary’
    - ➢ Goal: Implant the subject into this vocabulary



Rare-Token Identifier: Native Approaches

- Use English words to describe the new class
  - English lexicon has strong priors

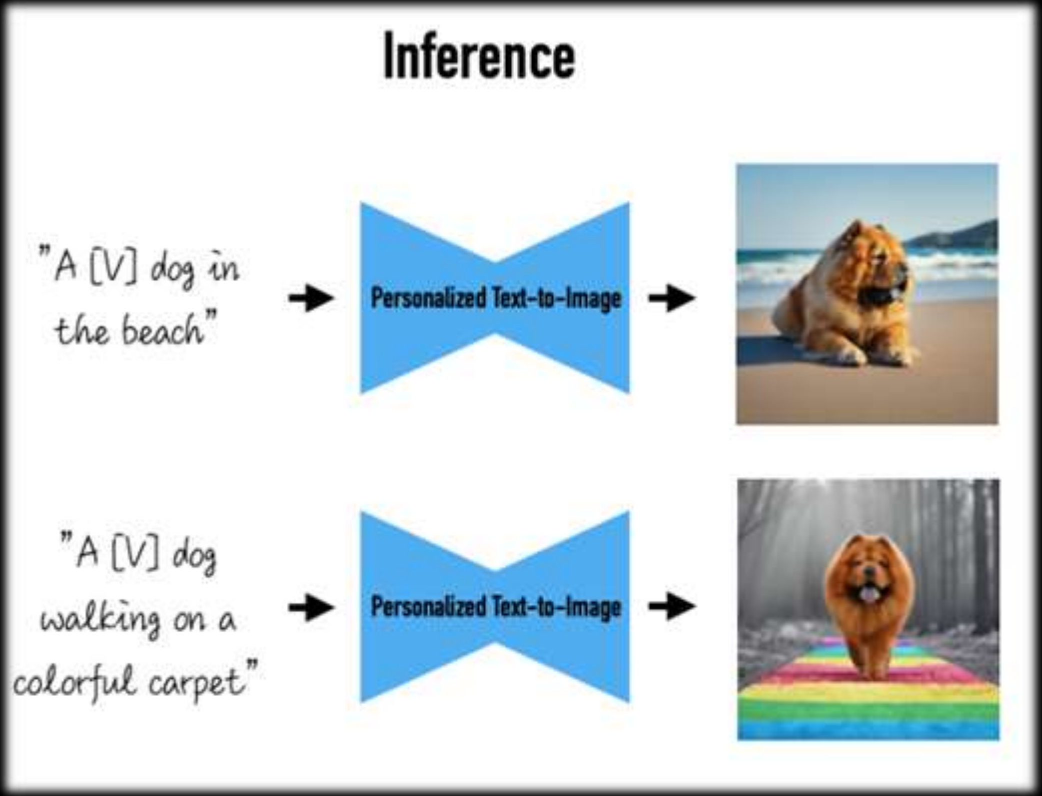
‘unique’ or ‘special’

- Generate a string of random characters
  - Nonsense identifier leads to literal artifacts

e.g. “xxy5syt00”

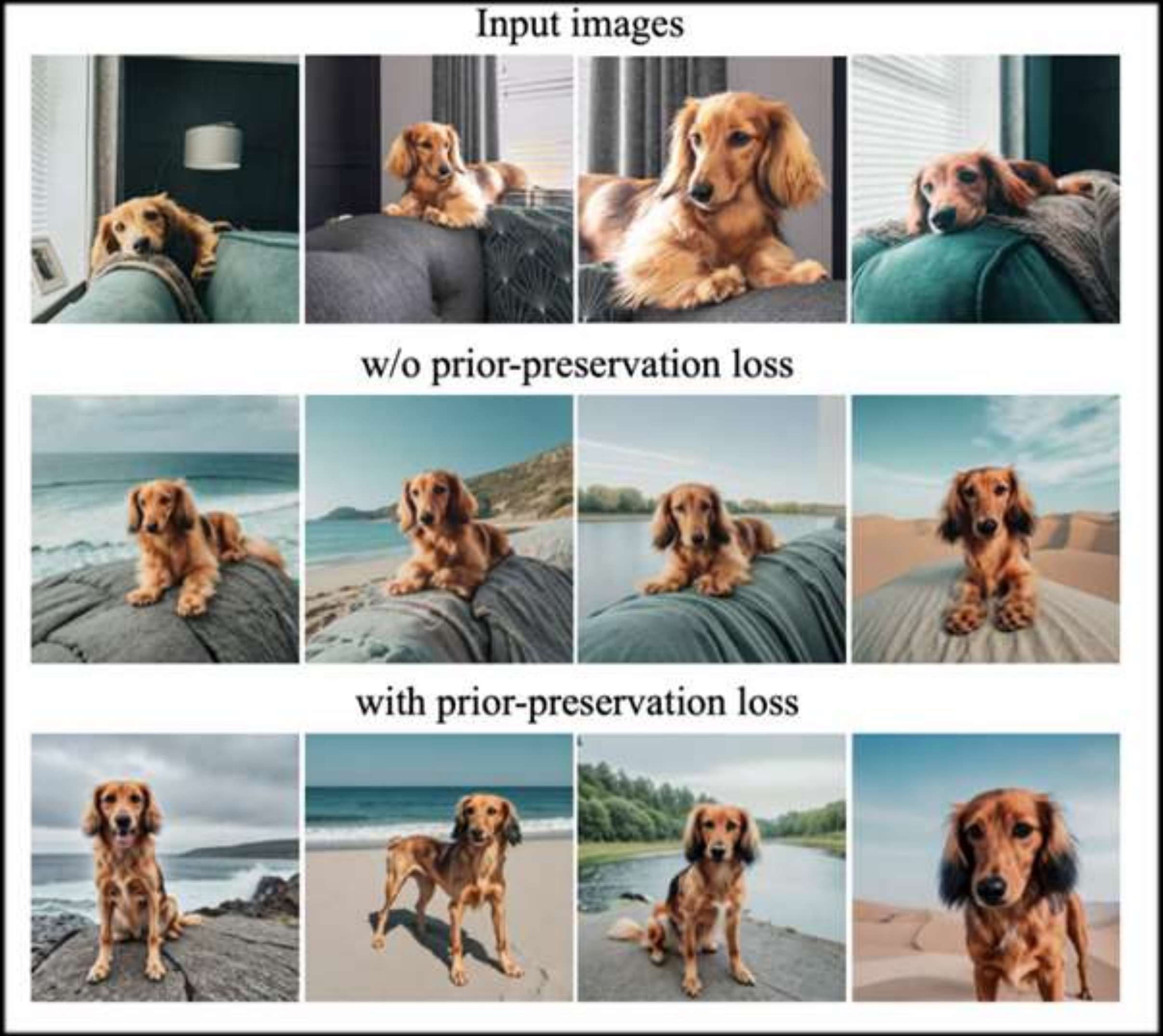
Rare-Token Identifier: Solution

1. Perform a rare-token lookup in the vocabulary
2. Invert the rare token, resulting in the plain text. (1~3 characters)
3. Use the plain text as the unique identifier



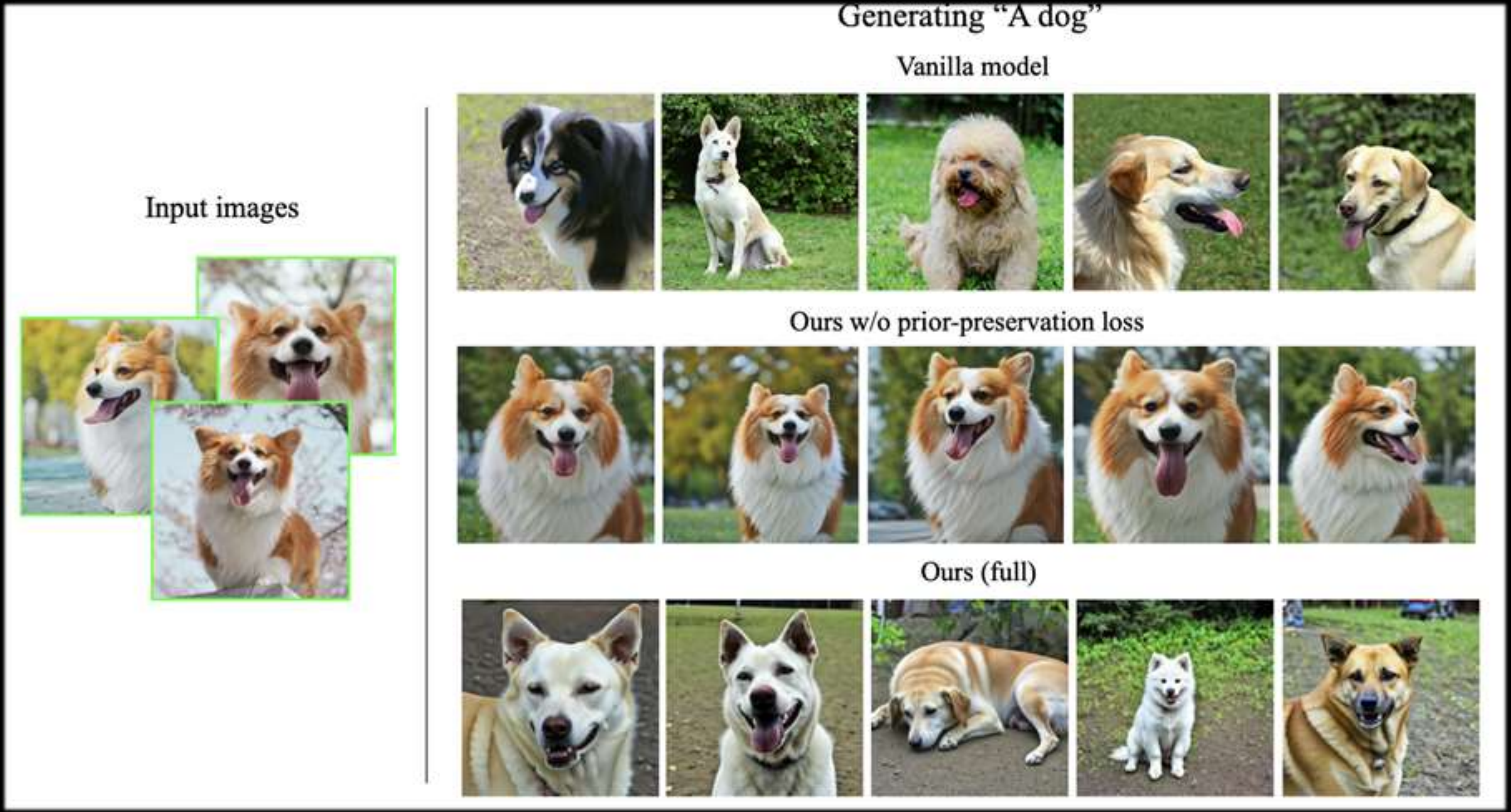
Autogenous Class-Specific Prior Preservation Loss

1. Overfitting

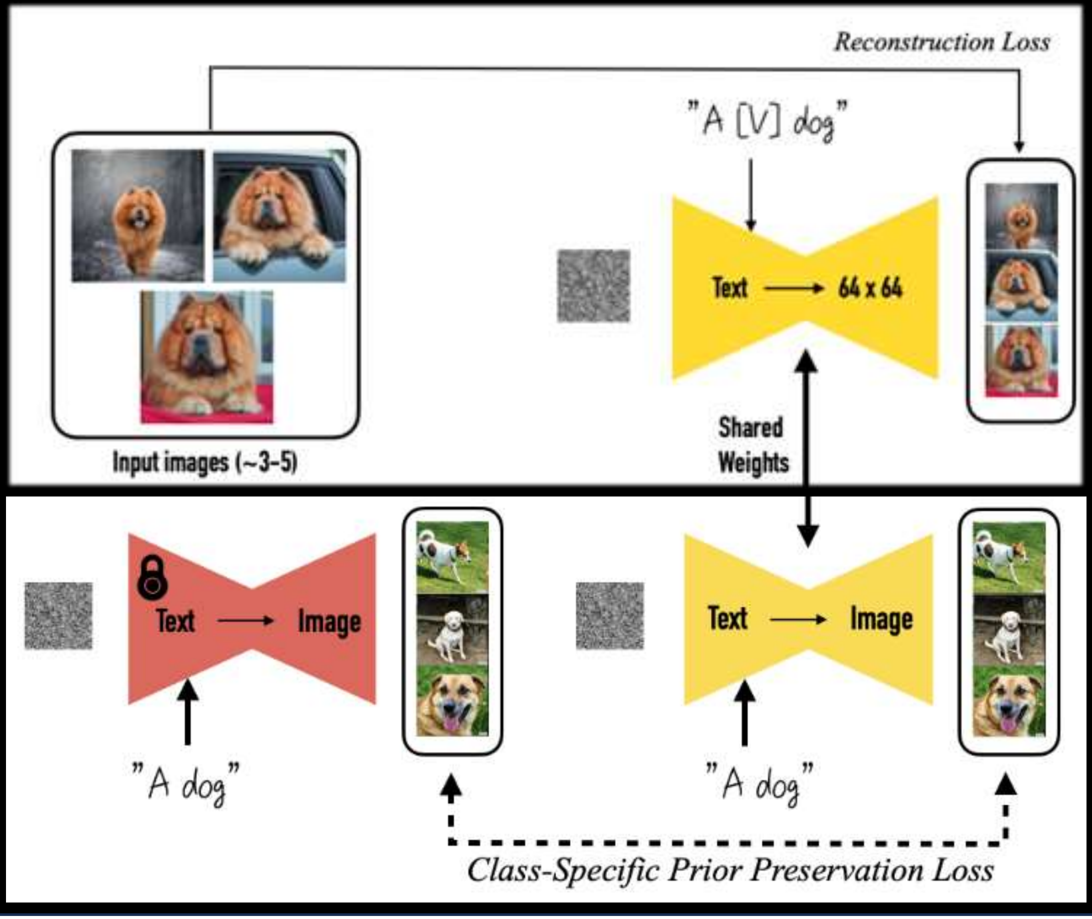


Autogenous Class-Specific Prior Preservation Loss

2. Language Drift



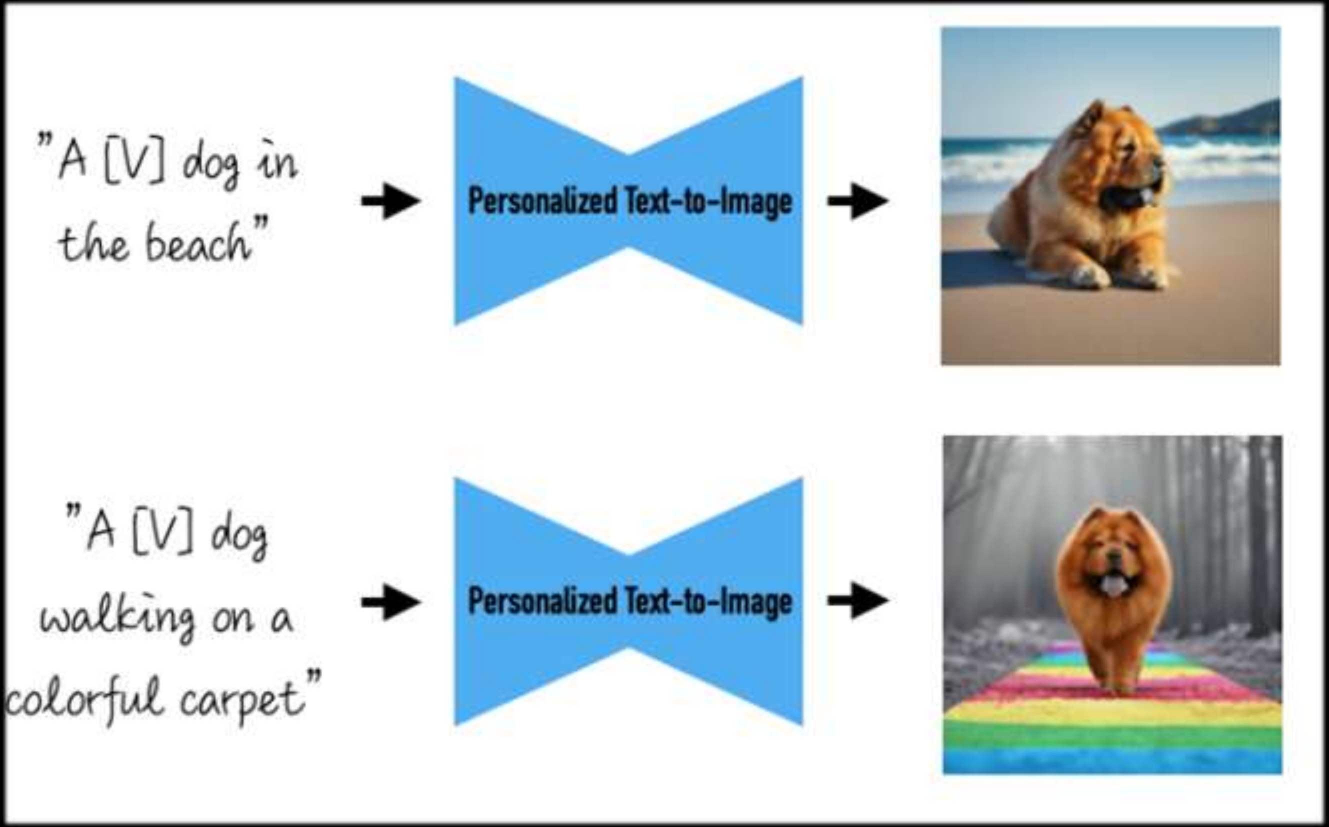
Autogenous Class-Specific Prior Preservation Loss



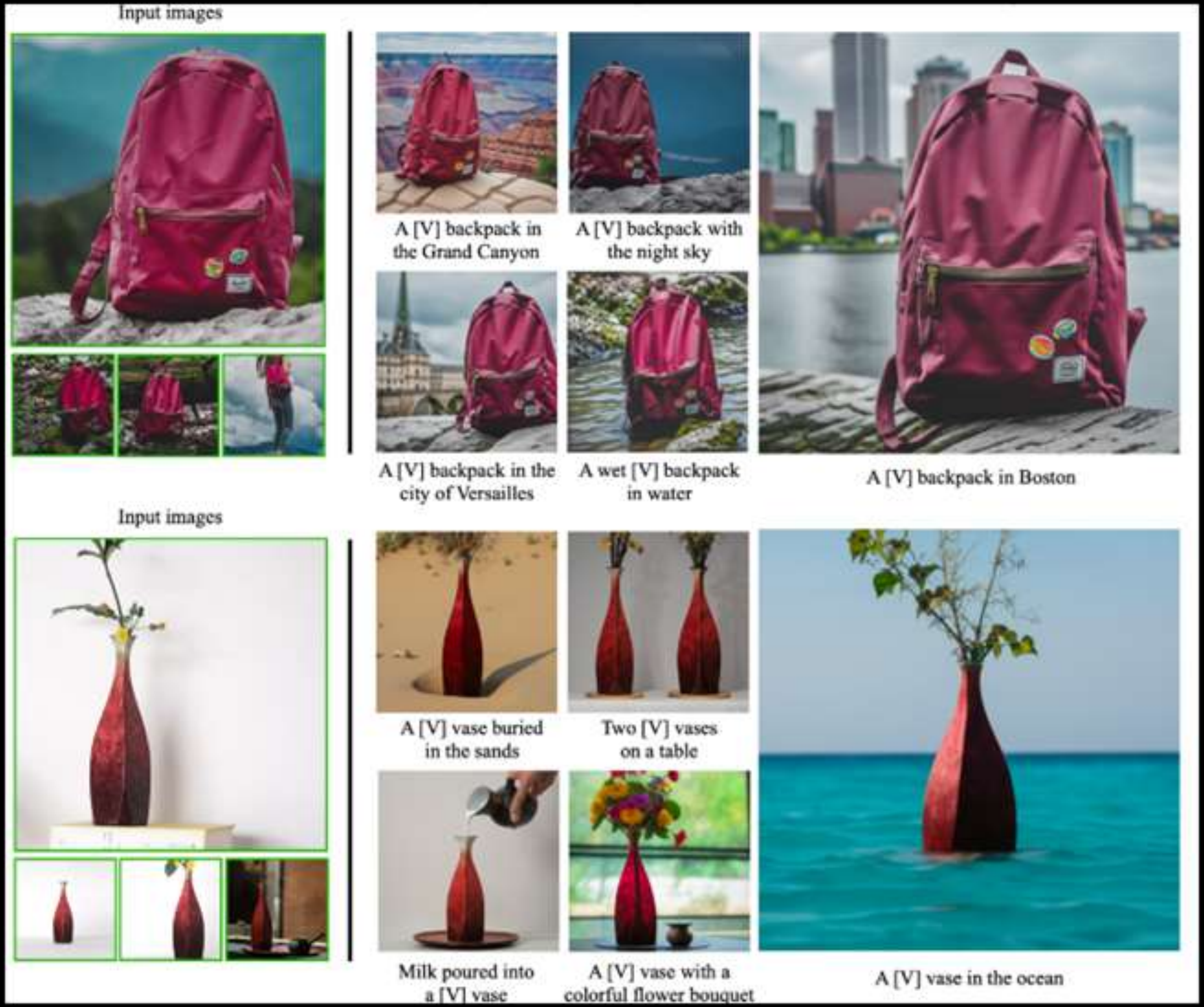
Standard Diffusion Loss — Autogenous Class-Specific Prior Preservation Loss

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[w_t\|\hat{\mathbf{x}}_\theta(\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\epsilon},\mathbf{c})-\mathbf{x}\|_2^2\right] \quad +\lambda w_{t'}\|\hat{\mathbf{x}}_\theta(\alpha_{t'}\mathbf{x}_{\text{pr}}+\sigma_{t'}\boldsymbol{\epsilon}',\mathbf{c}_{\text{pr}})-\mathbf{x}_{\text{pr}}\|_2^2]$$

Results

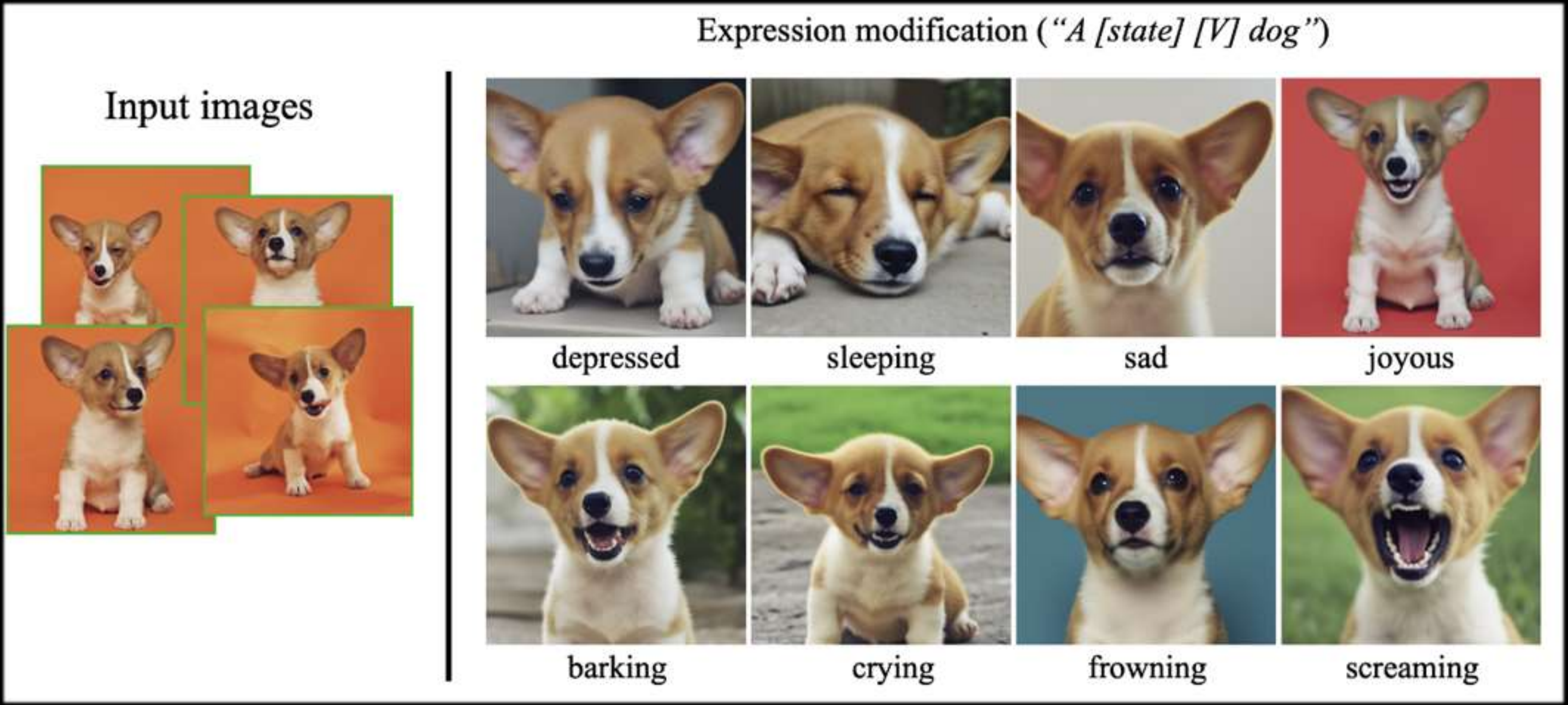

"A [V] dog in the beach"
Personalized Text-to-Image
"A [V] dog walking on a colorful carpet"
Personalized Text-to-Image

Results: Recontextualization

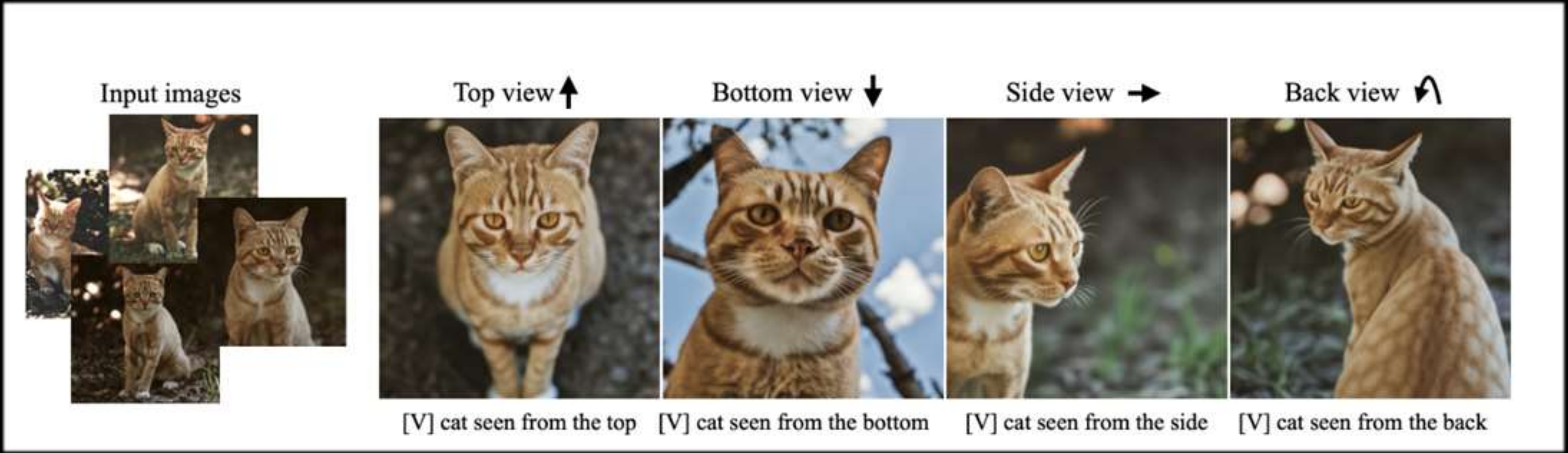


Results: Art Renditions

Results: Expression Manipulation

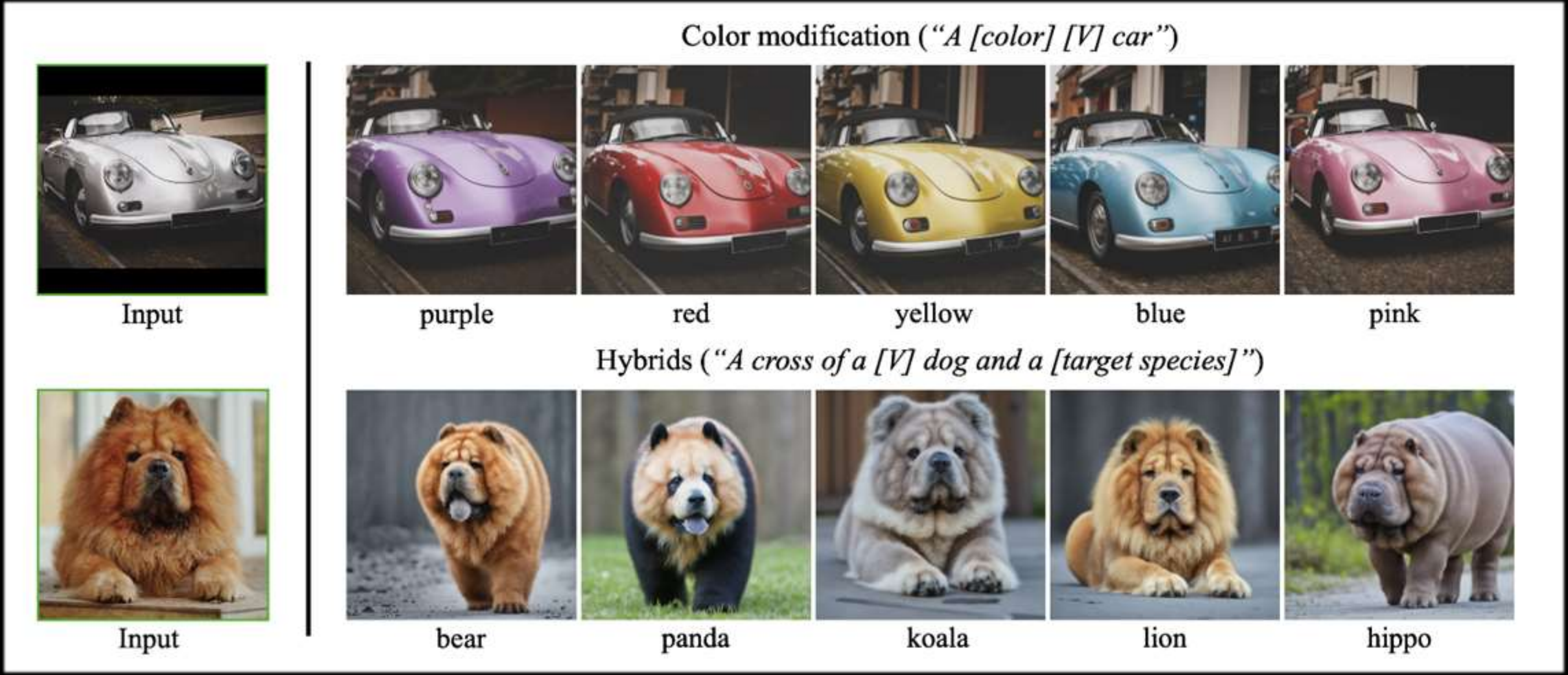


Results: View Synthesis



Results: Accessorization

Results:Property Modification



Main Failures

Reference

- *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation*
- *Generative Adversarial Nets*
- *Denoising Diffusion Probabilistic Models*
- *High-Resolution Image Synthesis With Latent Diffusion Models*
- *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion*
- https://dreambooth.github.io/
- https://jalammar.github.io/illustrated-stable-diffusion/
- https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e
- https://theaisummer.com/diffusion-models/
- https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c
- https://kailashahirwar.medium.com/a-very-short-introduction-to-diffusion-models-a84235e4e9ae
- https://erdem.pl/2023/11/step-by-step-visual-introduction-to-diffusion-models#forward-diffusion-diagram

Acknowledgement

Thank you for your time and listening!