Stable Diffusion fine tuning approaches

Dominik Lewy

LLMs Landscape

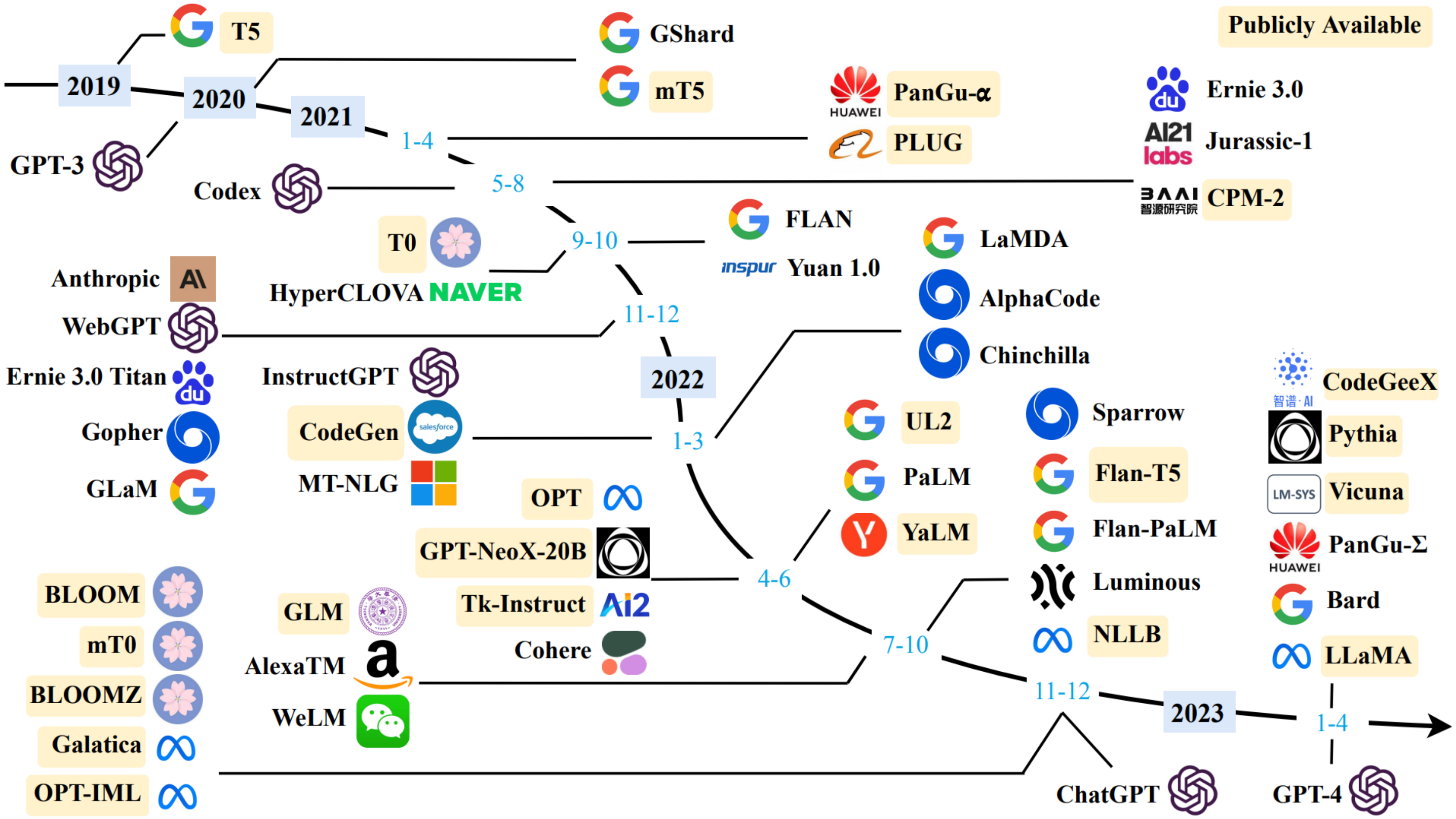


Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years.

Graphics source: https://arxiv.org/pdf/2303.18223.pdf A Survey of Large Language Models

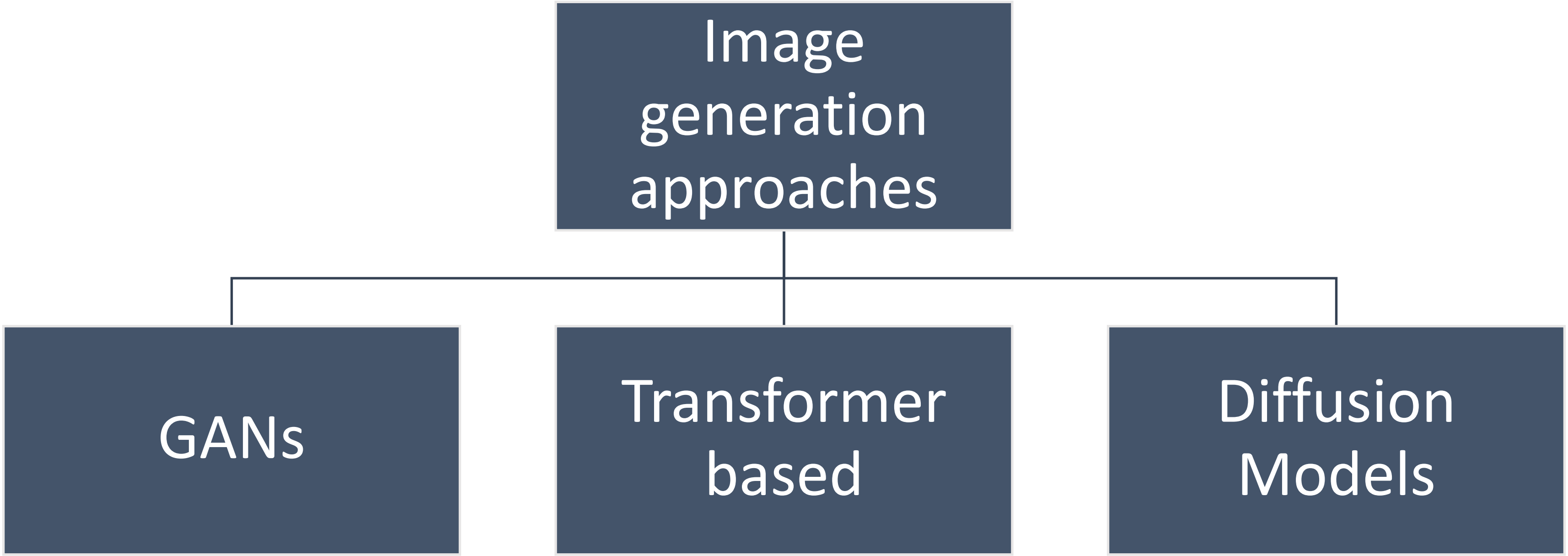
What will be tomorrow?

	PRE-2020	2020	2022	2023?	2025?	2030?
TEXT	Spam detection Translation Basic Q&A	Basic copy writing First drafts	Longer form Second drafts	Vertical fine tuning gets good (scientific papers, etc)	Final drafts better than the human average	Final drafts better than professional writers
CODE	1-line auto-complete	Multi-line generation	Longer form Better accuracy	More languages More verticals	Text to product (draft)	Text to product (final), better than full-time developers
IMAGES			Art Logos Photography	Mock-ups (product design, architecture, etc.)	Final drafts (product design, architecture, etc.)	Final drafts better than professional artists, designers, photographers)
VIDEO / 3D / GAMING			First attempts at 3D/video models	Basic / first draft videos and 3D files	Second drafts	AI Roblox Video games and movies are personalized dreams

Large model availability: First attempts · Almost there · Ready for prime time

Graphics source: https://www.sequoiacap.com/article/generative-ai-a-creative-new-world/

Historical outlook



Transformer based

Encoding for images

Source: https://en.wikipedia.org/wiki/Lists_of_Egyptian_hieroglyphs

DALL-E in practice

Input: Sentence A

Text BPE vocabuolary of size
16384

1	2	3	4	5	6

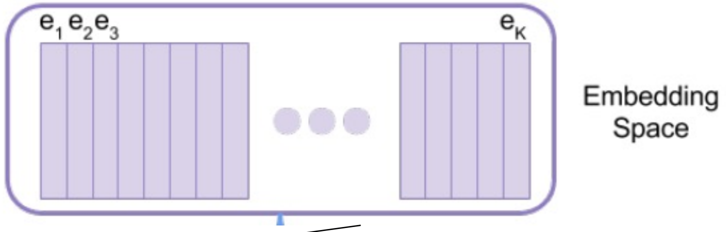
Text token

DALL-E in practice

<u>Input:</u> Sentence A

Text BPE vocabuolary of size
16384

Image learned vocabulary of size
8192



1	2	3	4	5	6	1

Text token

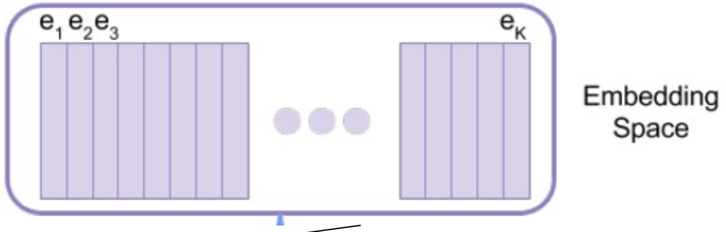
Image token

DALL-E in practice

Input: Sentence A

Text BPE vocabuolary of size
16384

Image learned vocabulary of size
8192



Text token						Image token					
1	2	3	4	5	6	1	2	3	4	5	...

Text token are connected to all input tokens via attention

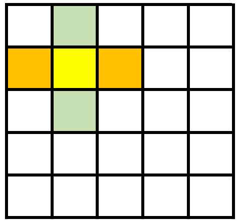


Image token in addition to seeing text tokens have colum and row attention

Text token

Image token

DALL-E in practice

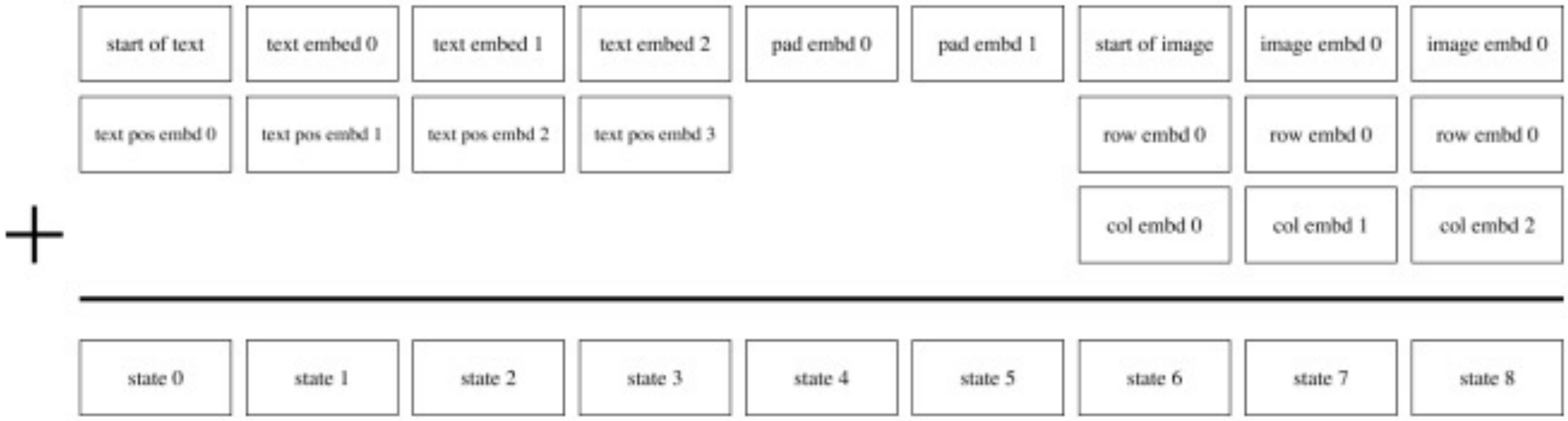


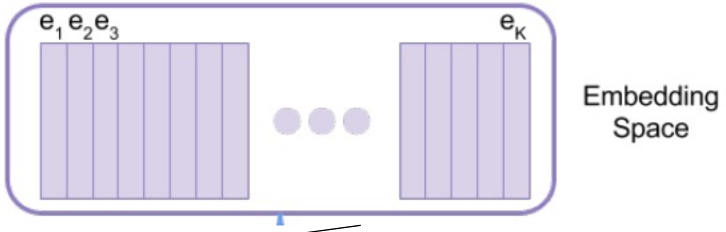
Figure 10. Illustration of the embedding scheme for a hypothetical version of our transformer with a maximum text length of 6 tokens. Each box denotes a vector of size $d_{\text{model}} = 3968$. In this illustration, the caption has a length of 4 tokens, so 2 padding tokens are used (as described in Section 2.2). Each image vocabulary embedding is summed with a row and column embedding.

DALL-E in practice

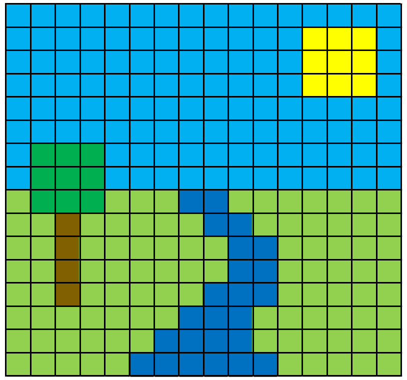
Input: Sentence A

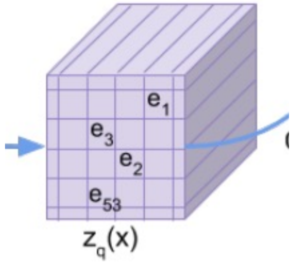
Text BPE vocabuolary of size 16384

Image learned vocabulary of size 8192



1	2	3	4	5	6	1	2	3	4	5	...





VQ-VAE

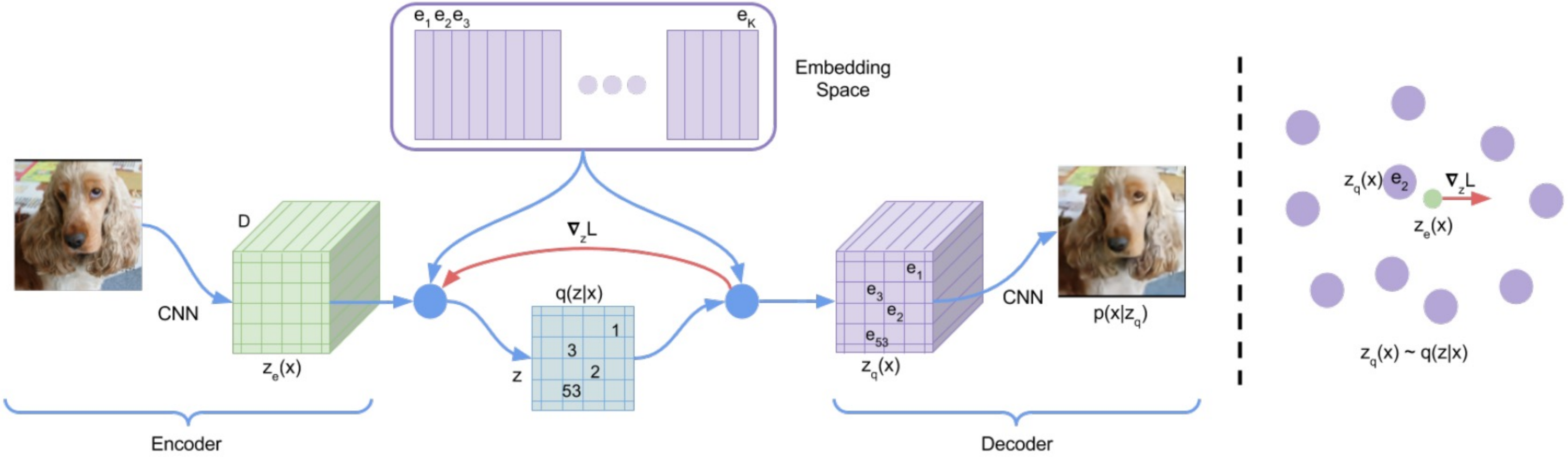
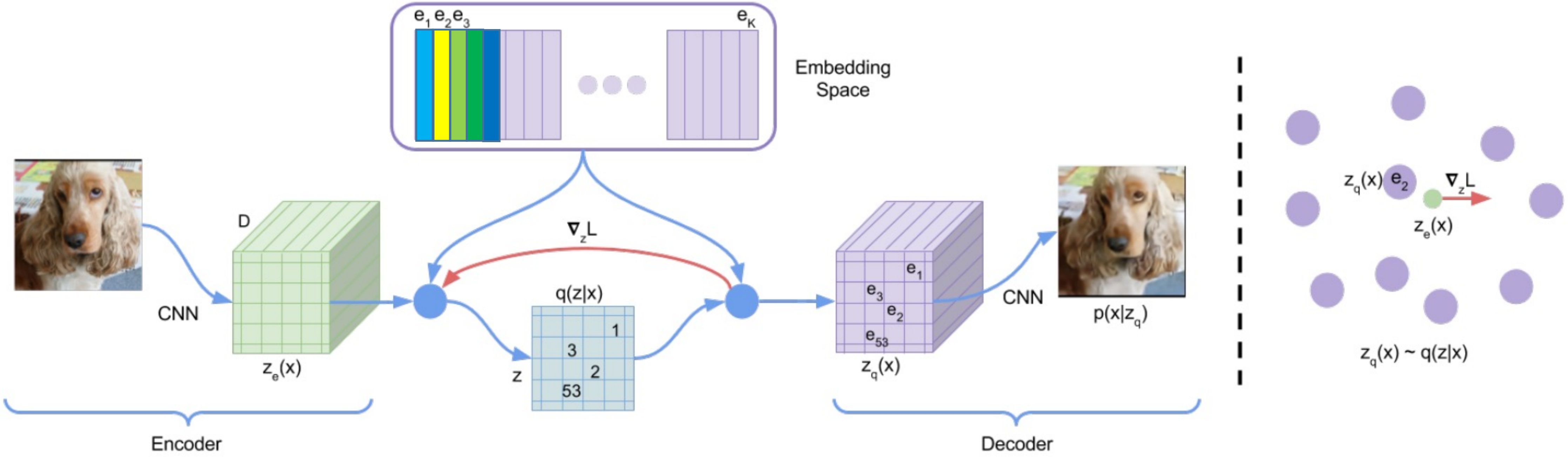
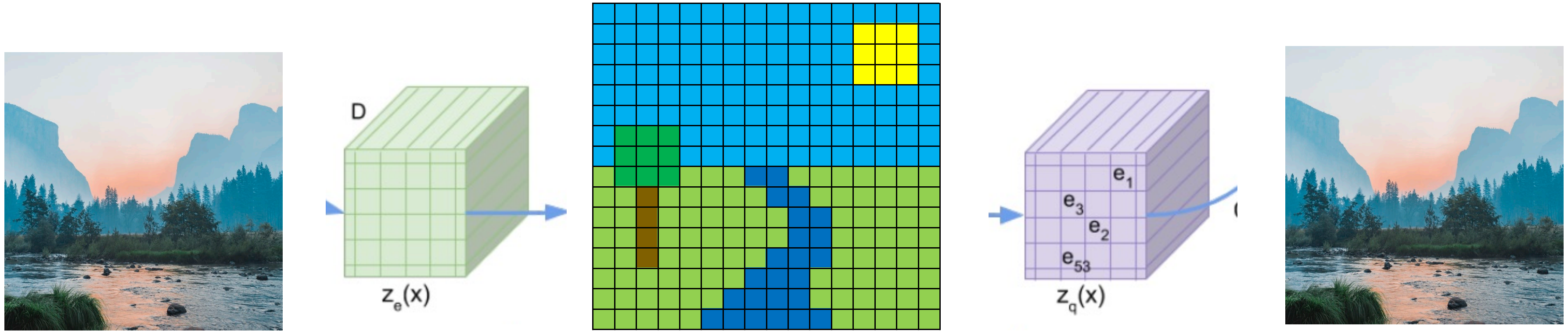


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

Source: https://arxiv.org/pdf/1711.00937.pdf

VQ-VAE





Source: https://arxiv.org/pdf/1711.00937.pdf

Diffusion Models

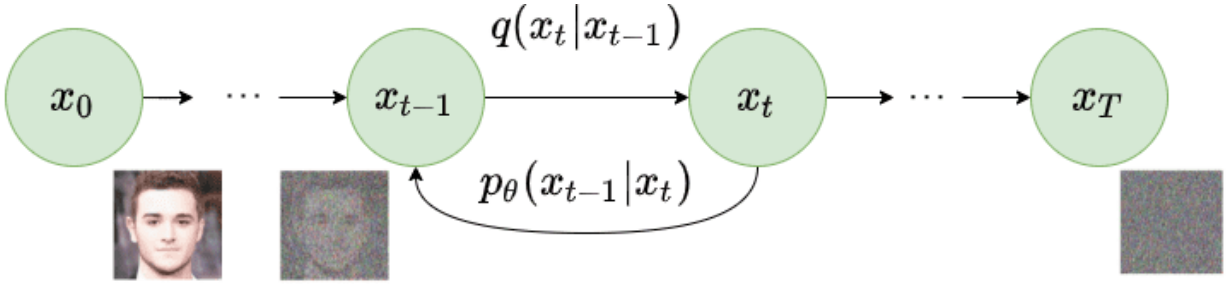
Diffusion Models

Models:

- Diffusion models have emerged as a new state-of-the-art family of deep generative models.
- They broke the long domination of Generative Adversary Networks (GANs) and Transformers.
- Over the past year 2022, numerous architectures have been developed:
  - Glide [1],
  - Dalle2 [2],
  - Imagen [3]
  - StableDiffusion [4]
    - v 1.0
    - v 2.0, 2.1

Principles:

- Diffusion models aim to decompose the image generation process (sampling) in many small "denoising" steps.
- In foward process, diffusion models take the input image and gradually add Gaussian noise to it through a series of T steps (Markov chain).
- Neural network is then trained to recover the original data by reversing the noising process.



Stable Diffusion

Stable Diffusion – What are the key components?

ClipText

- Text encoder used to translate prompt into information guiding the process of noise removal
- Input: text.
- Output: 77 token embeddings vectors, each in 768 dimensions.

+

VAE

- Image Encoder/Decoder – runs once at the end of the process to create final image in pixel space
- Input / Output: The processed information array (dimensions: (4,64,64))
- Output / Input: The resulting image (dimensions: (3, 512, 512) which are (red/green/blue, width, height))

+

UNet

- Image information creator – runs iteratively for N steps to produce the result. It works in the latent space (image information space)
- Input: text embeddings and a starting multi-dimensional array (structured lists of numbers, also called a *tensor*) made up of noise.
- Output: A processed information array

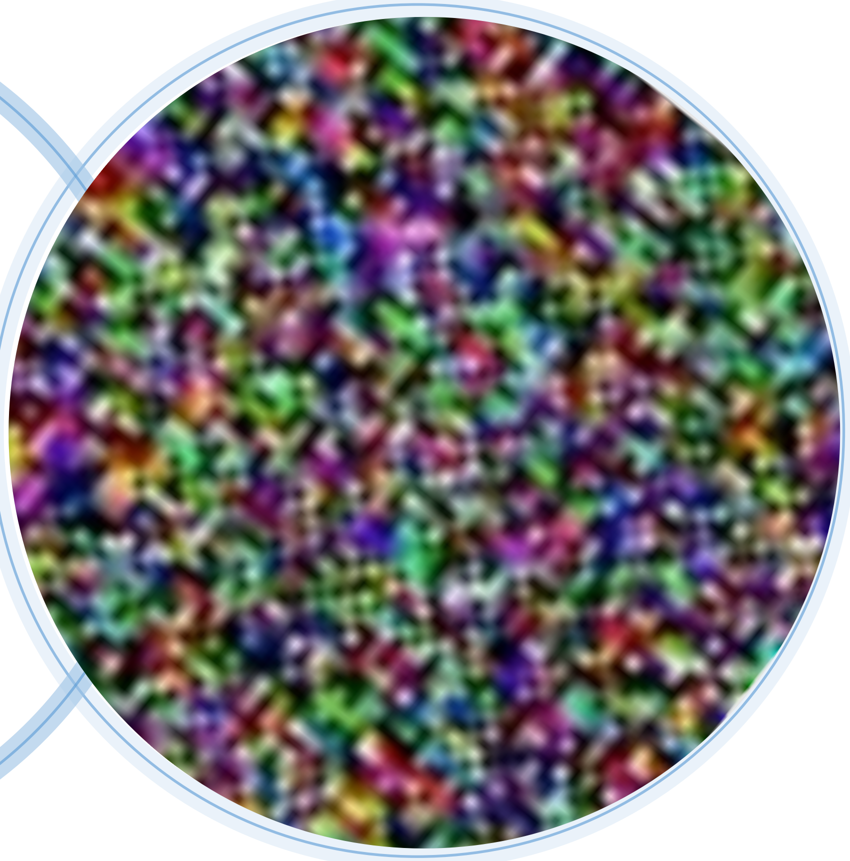
Intuition behind each component

Prompt: "a handsome cat dressed like Lincoln, trending art."

ClipText

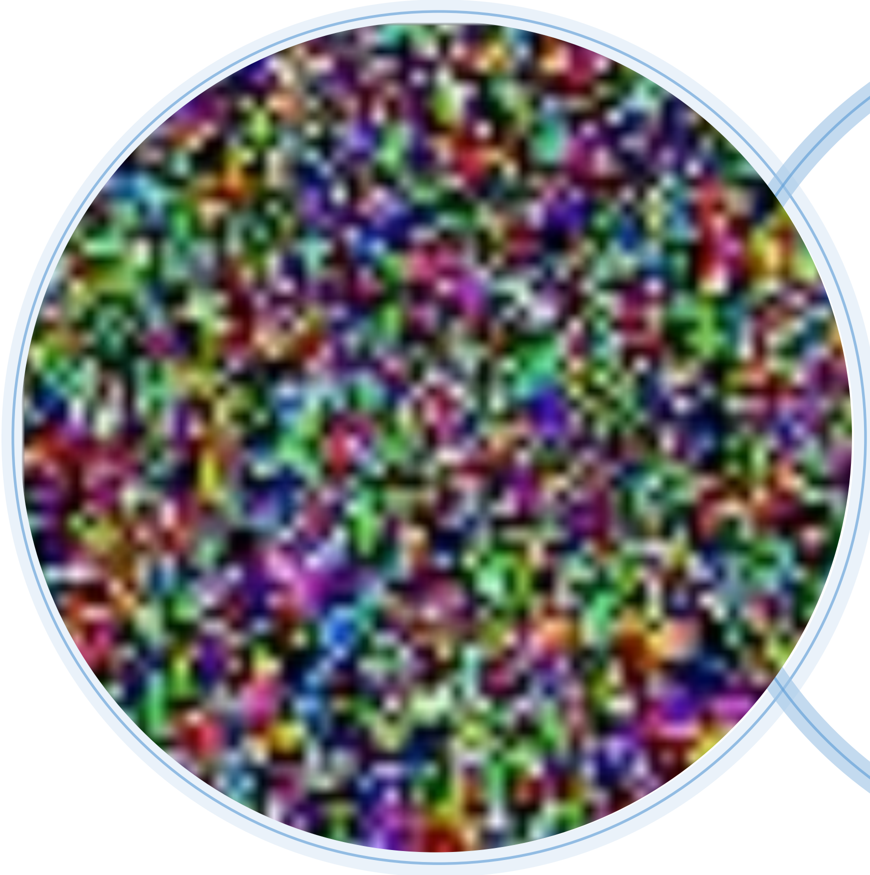
Text encoder used to translate prompt into information guiding the process of noise removal

- Input: text.
- Output: 77 token embeddings vectors, each in 768 dimensions.



Source: own material

Intuition behind each component



UNet

Image information creator – runs iteratively for N steps to produce the result. It works in the latent space (image information space)

- Input: text embeddings and a starting multi-dimensional array (structured lists of numbers, also called a *tensor*) made up of noise.
- Output: A processed information array

Source: own material

Intuition behind each component

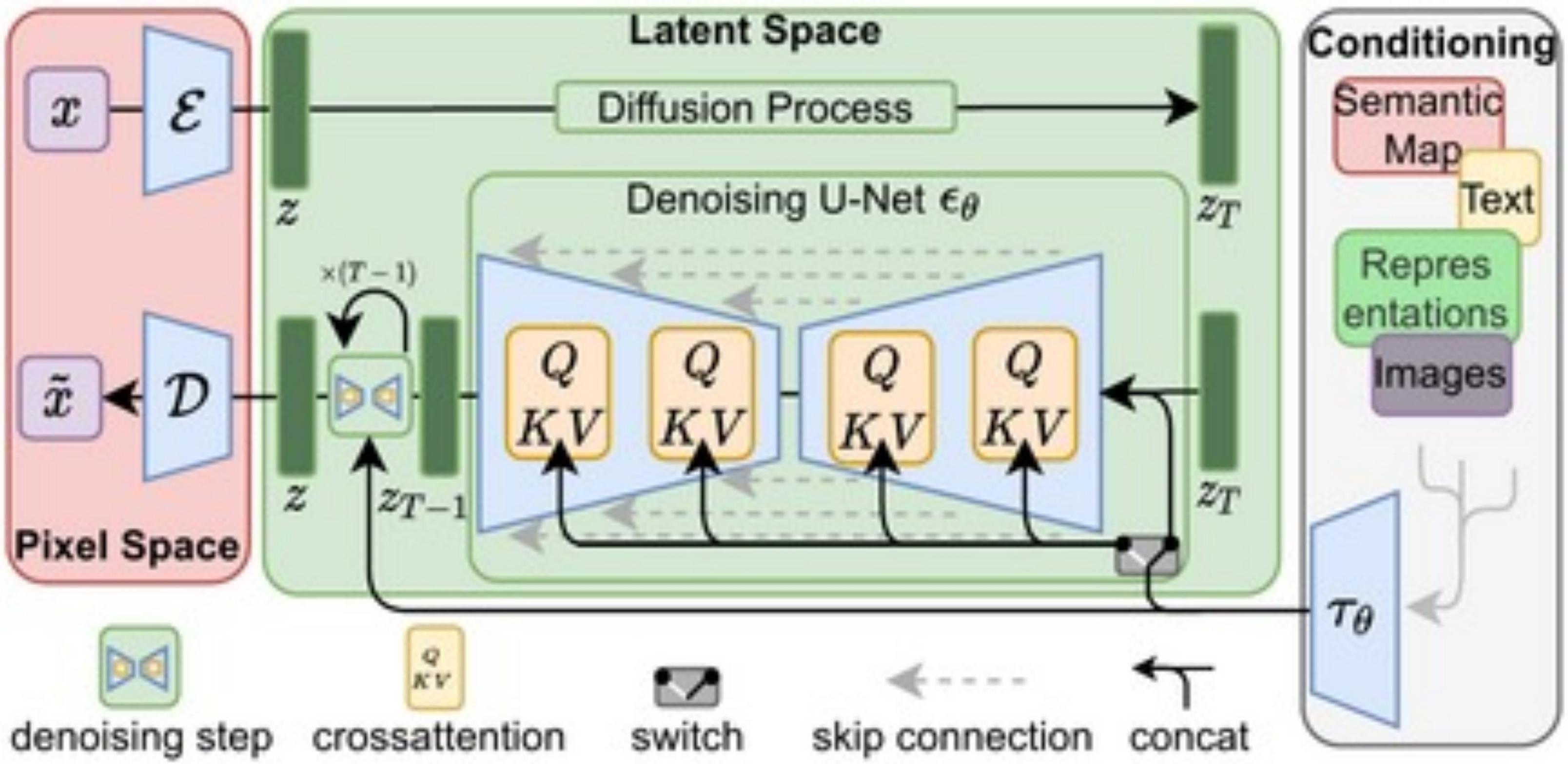
VAE

Image Encoder/Decoder – runs once at the end of the process to create final image in pixel space

- Input / Output: The processed information array (dimensions: (4,64,64))
- Output / Input: The resulting image (dimensions: (3, 512, 512) which are (red/green/blue, width, height))

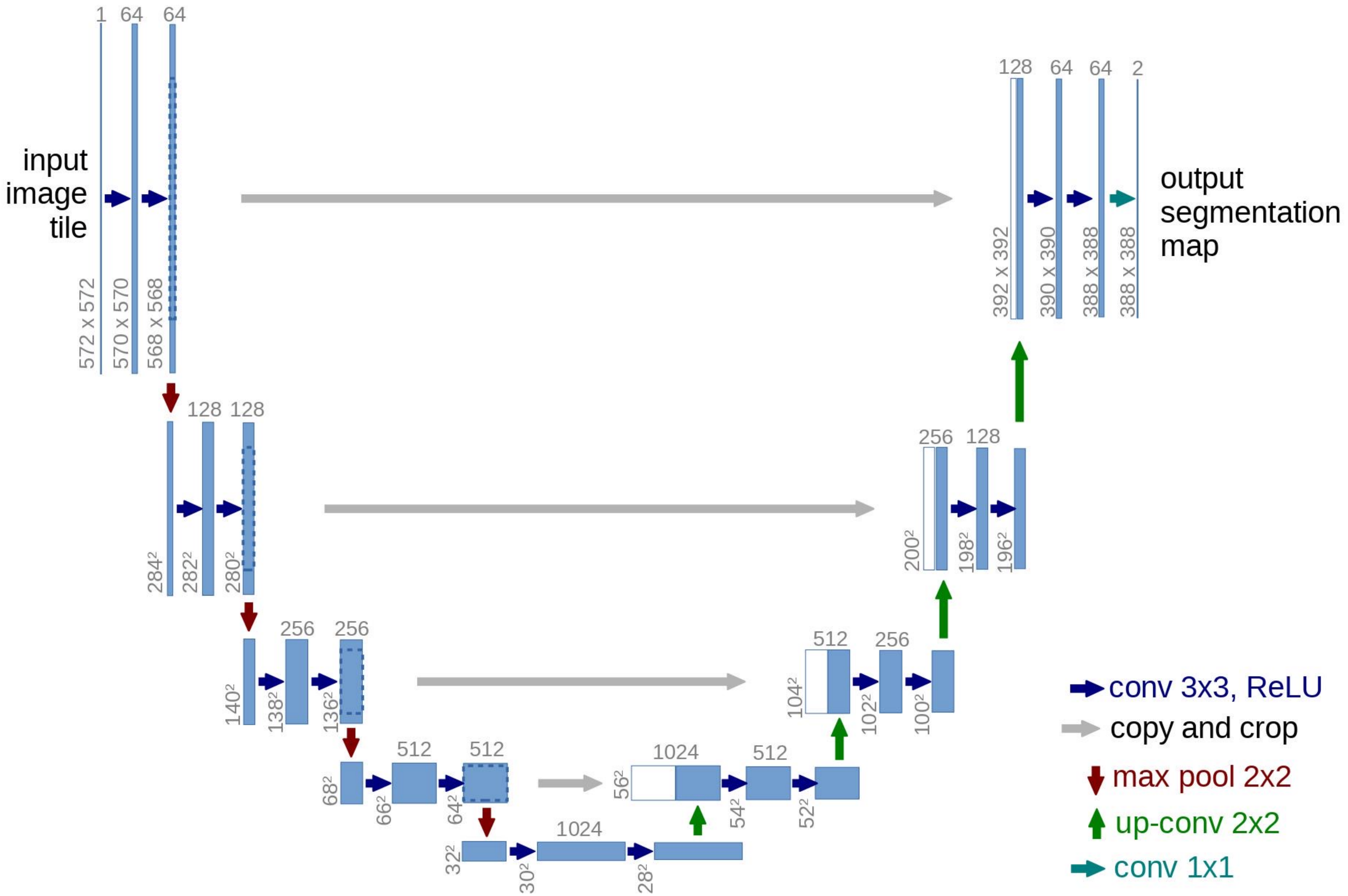
Source: own material

Stable Diffusion



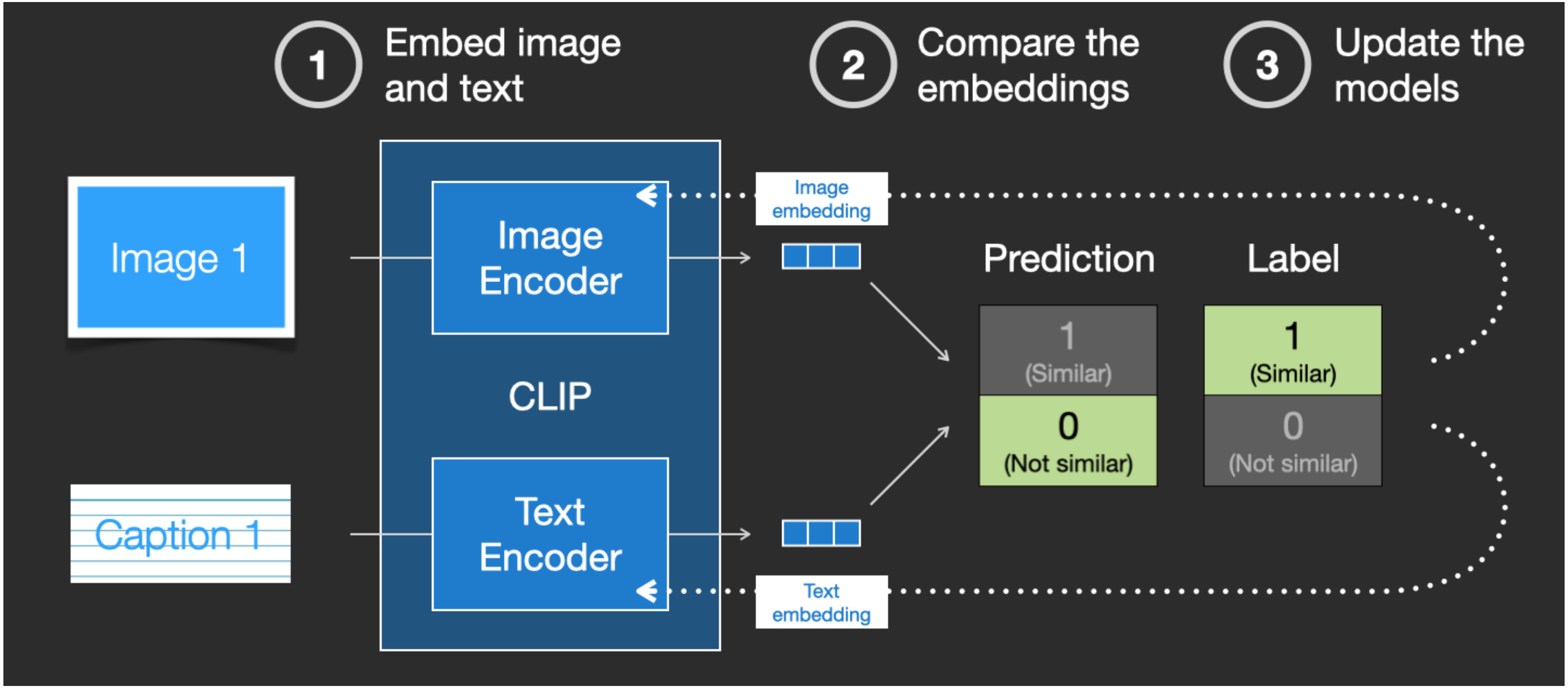
Source: Rombach & Blattmann, et al. “High-Resolution Image Synthesis with Latent Diffusion Models." CVPR 2022.

Denoising UNet



Source: https://scholar.harvard.edu/binxuw/classes/machine-learning-scratch/materials/stable-diffusion-scratch

CLIP text model



Source: https://jalammar.github.io/illustrated-stable-diffusion/

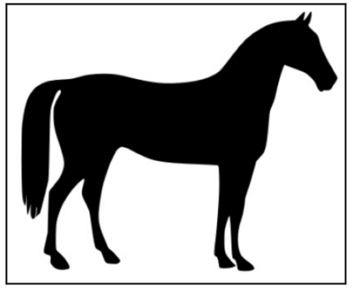
CLIP text model

chelsea.png
a facial photo of a tabby cat

rocket.jpg
a rocket standing on a launchpad

camera.png
a person looking at a camera on a tripod

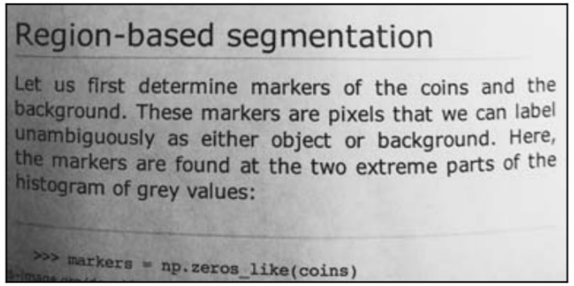
horse.png
a black-and-white silhouette of a horse



coffee.png
a cup of coffee on a saucer

astronaut.png
a portrait of an astronaut with the American flag

page.png
a page of text about segmentation



Region-based segmentation

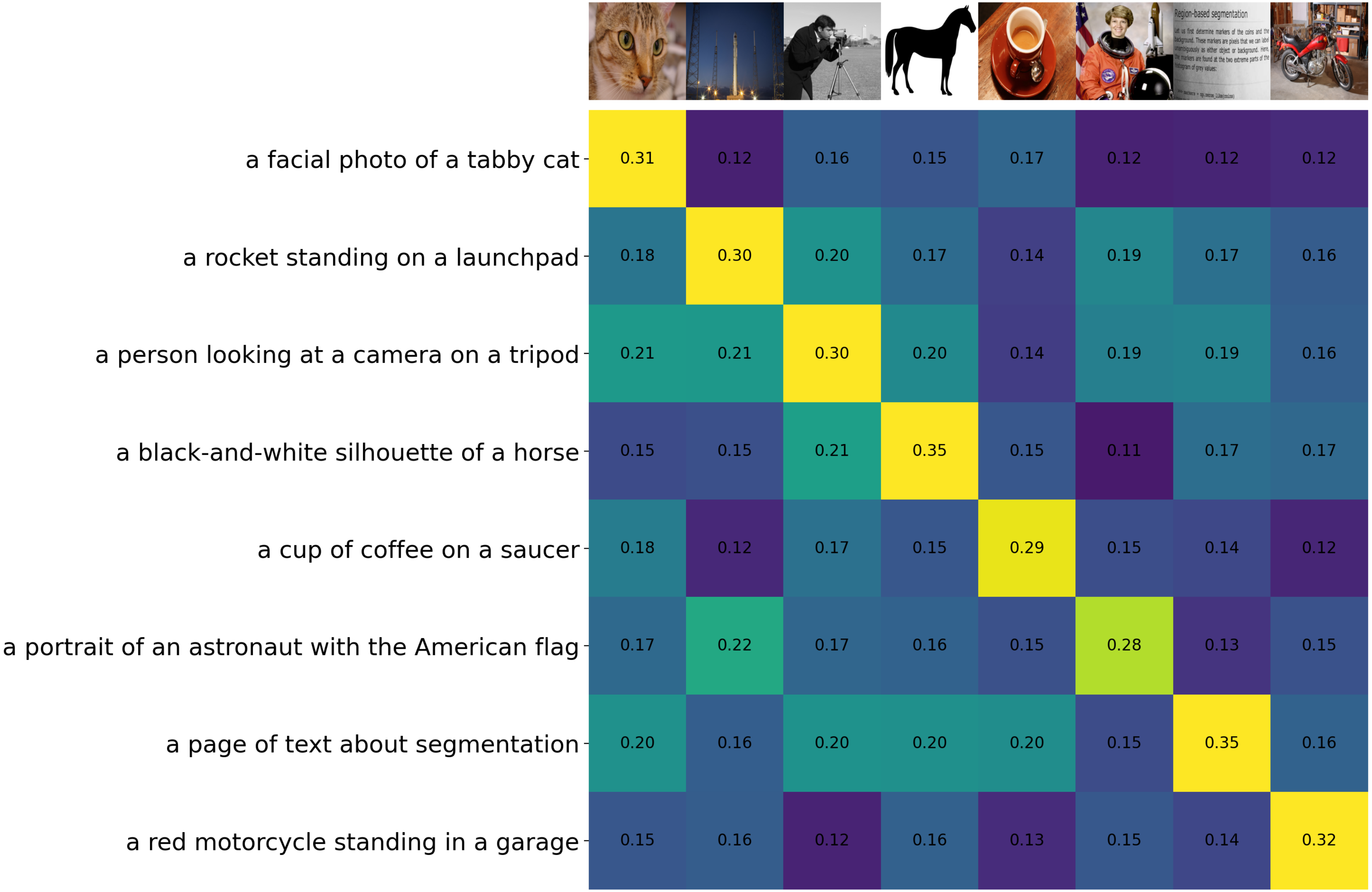
Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here, the markers are found at the two extreme parts of the histogram of grey values:

```
>>> markers = np.zeros_like(coins)
```

motorcycle_right.png
a red motorcycle standing in a garage

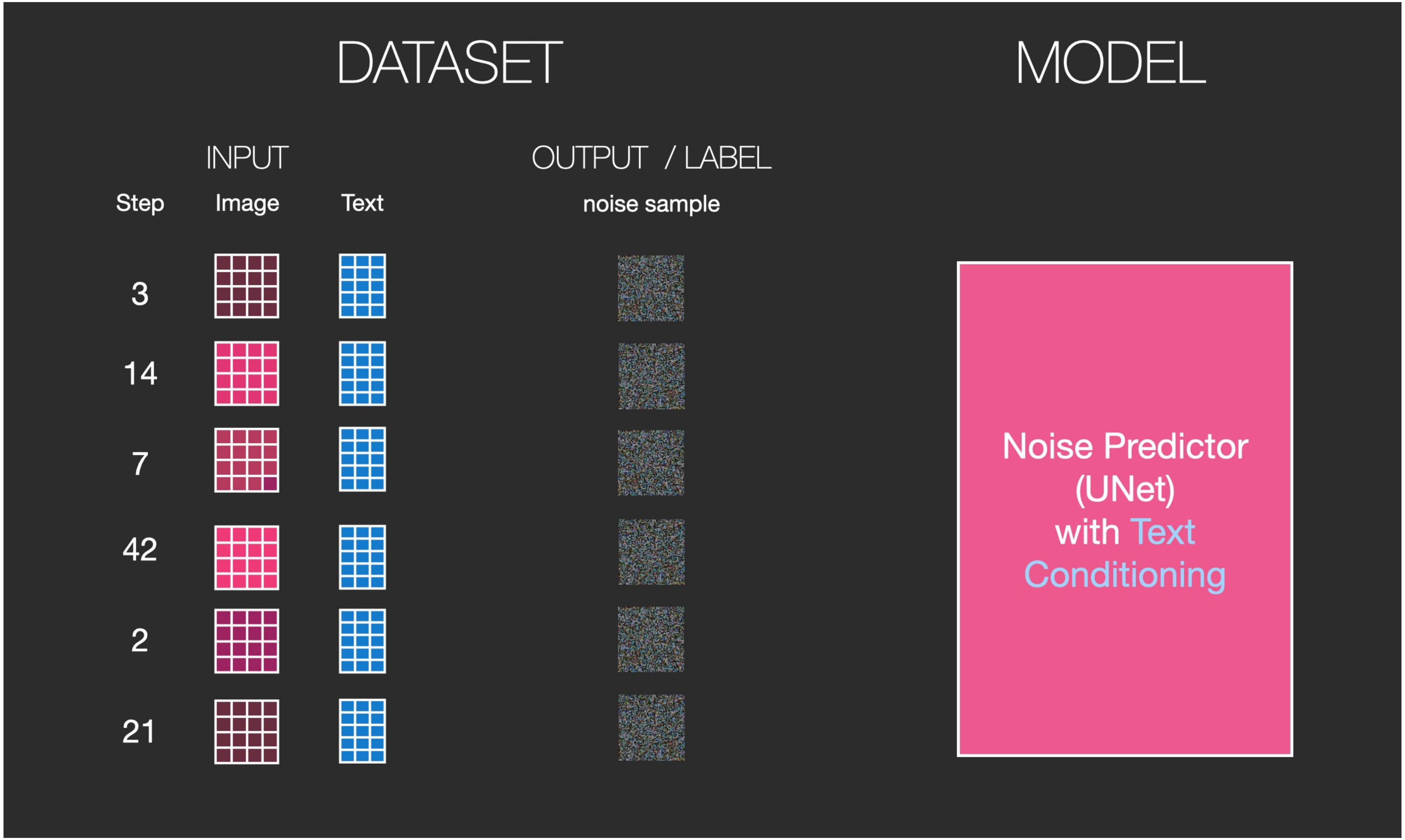
Source: https://colab.research.google.com/github/openai/clip/blob/master/notebooks/Interacting_with_CLIP.ipynb#scrollTo=C5zvMxh8cU6m

CLIP text model



Source: https://colab.research.google.com/github/openai/clip/blob/master/notebooks/Interacting_with_CLIP.ipynb#scrollTo=C5zvMxh8cU6m

CLIP text model



Source: https://jalammar.github.io/illustrated-stable-diffusion/

Encoder / Decoder

Training:

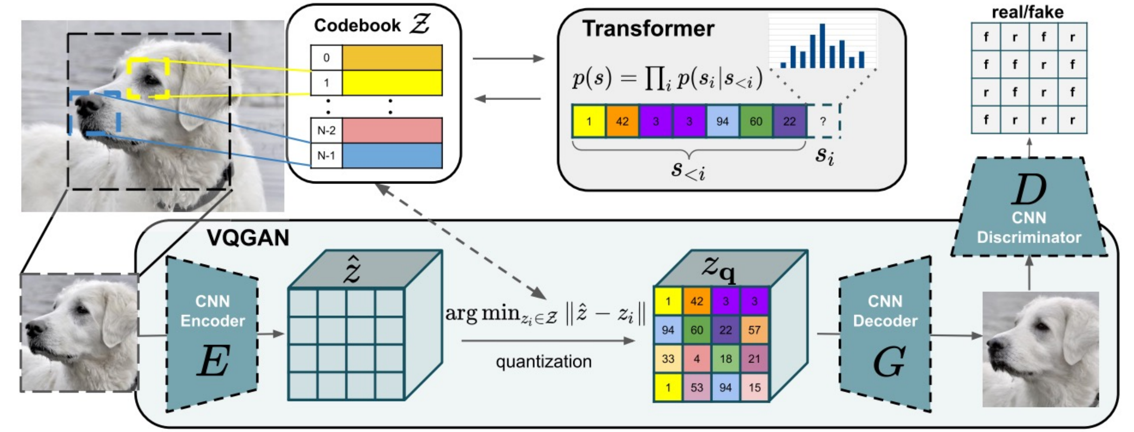


Figure 2. Our approach uses a convolutional *VQGAN* to learn a codebook of context-rich visual parts, whose composition is subsequently modeled with an autoregressive transformer architecture. A discrete codebook provides the interface between these architectures and a patch-based discriminator enables strong compression while retaining high perceptual quality. This method introduces the efficiency of convolutional approaches to transformer based high resolution image synthesis.

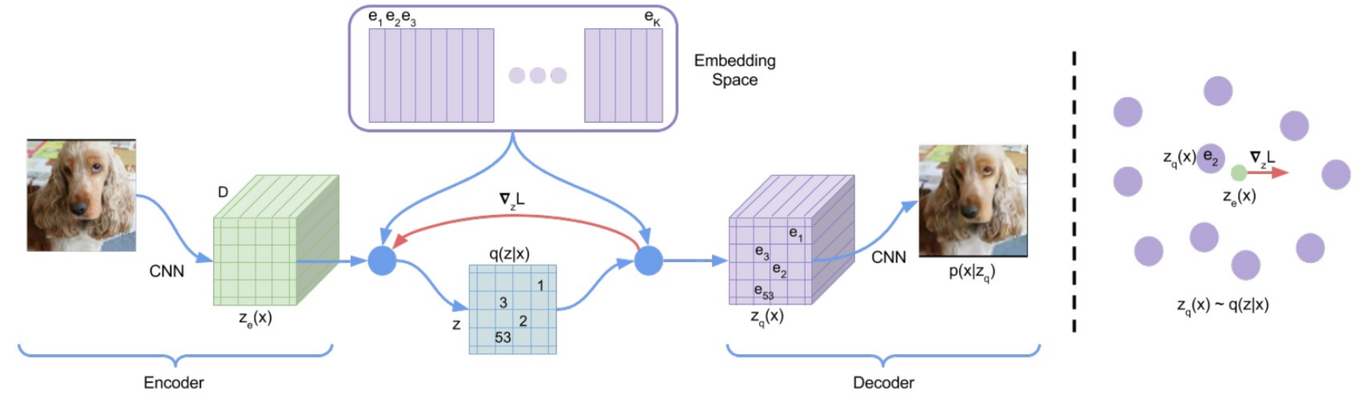
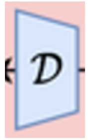


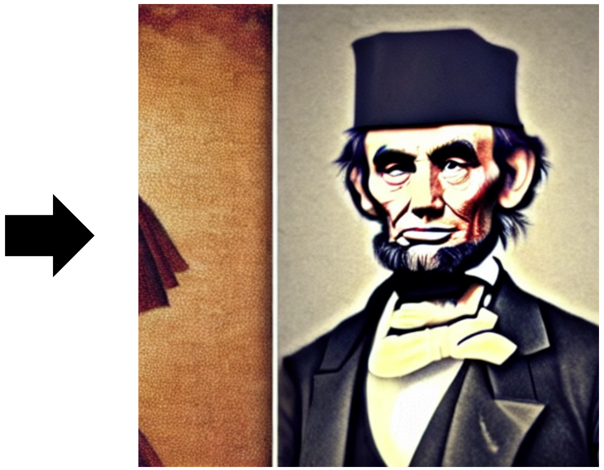
Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

SD usage:



x is $(3, 512, 512)$ image tensor

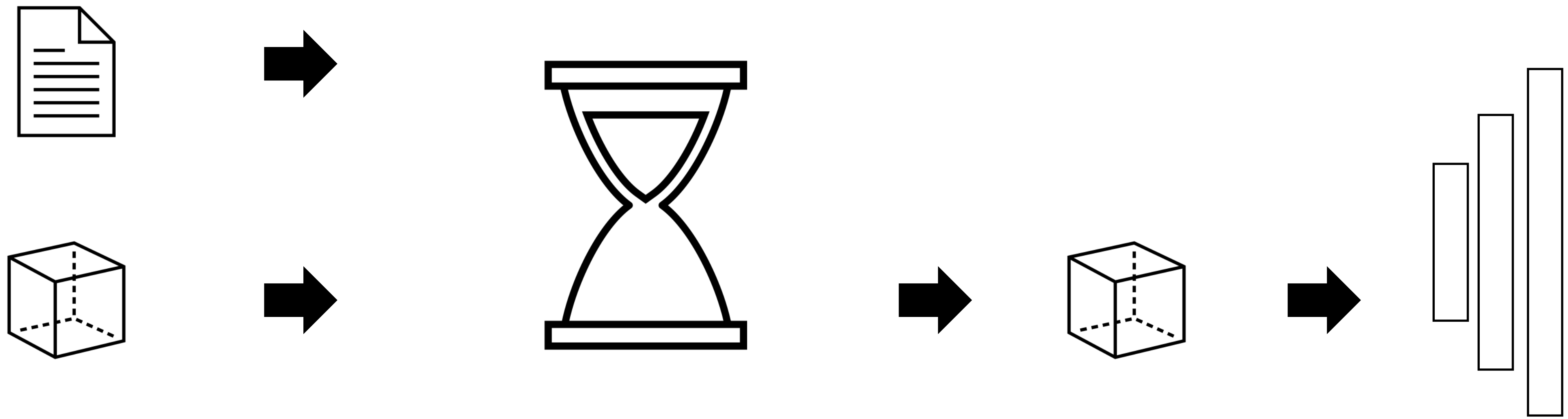
z is $(4, 64, 64)$ latent tensor



Source: https://colab.research.google.com/drive/1TvOlX2_l4pCBOKjDI672JcMm4q68sKrA?usp=sharing#scrollTo=xE07xceRUVO9

Inference pipeline

Text prompt



We start in the latent space z. A random noise of diemnsion $(4, 64, 64)$ is generated. This is controlled by seed.

Unet takes latent noise and text prompt as input and predicts noise in latent space.

Latent space with noise iteratively subtracted

VEA decoder

Source: https://scholar.harvard.edu/binxuw/classes/machine-learning-scratch/materials/stable-diffusion-scratch

Limitations

Limitations

- Limitations of Stable Diffusion
  - Not ideal for fine-grained tasks
  - Societal Impact
  - Revealing training data
  - Lacking ability to render a particular subject

Exploration – a dog

Adaptation – “A red SUV car with a surfing board painting on the side”

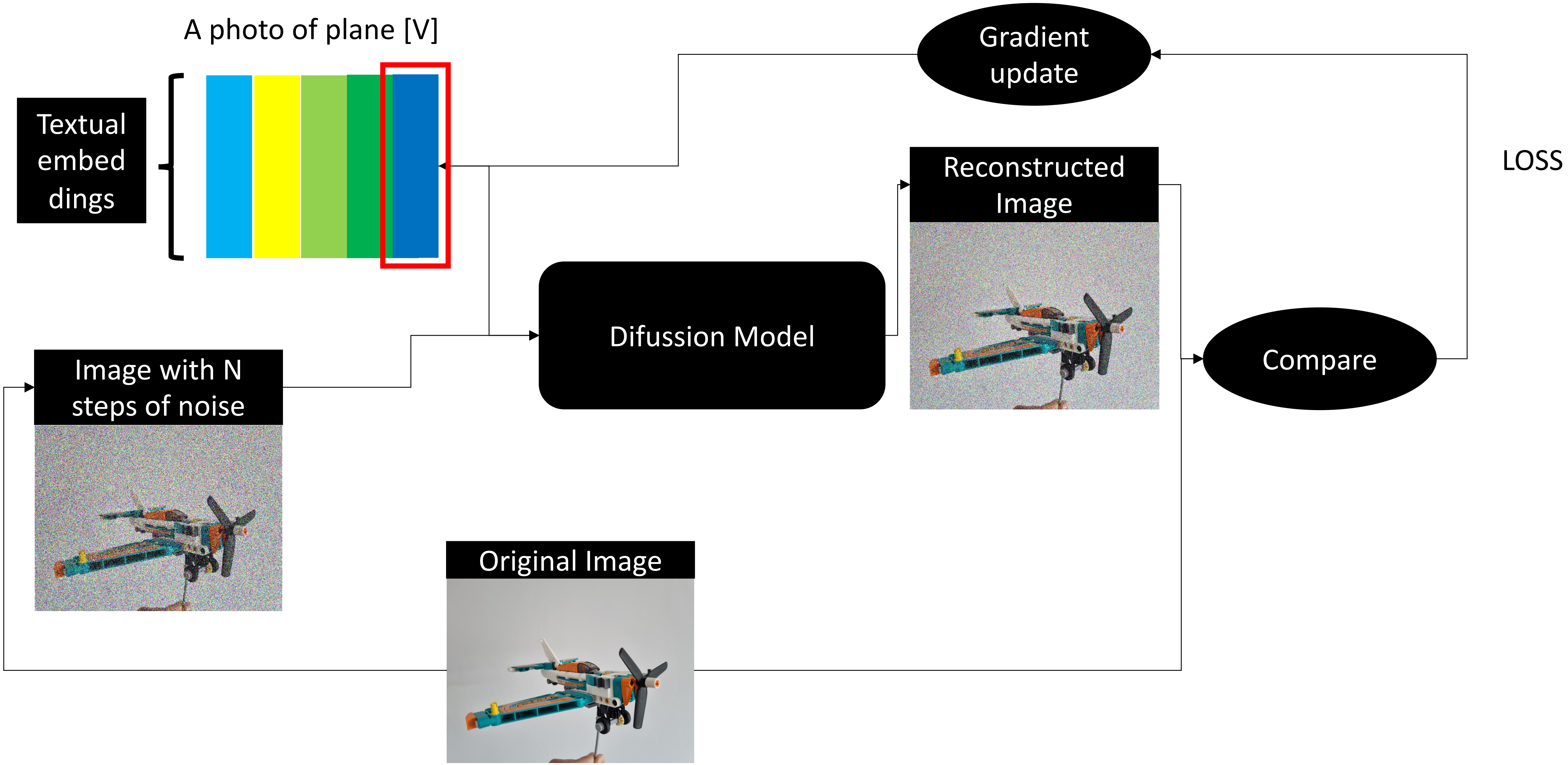




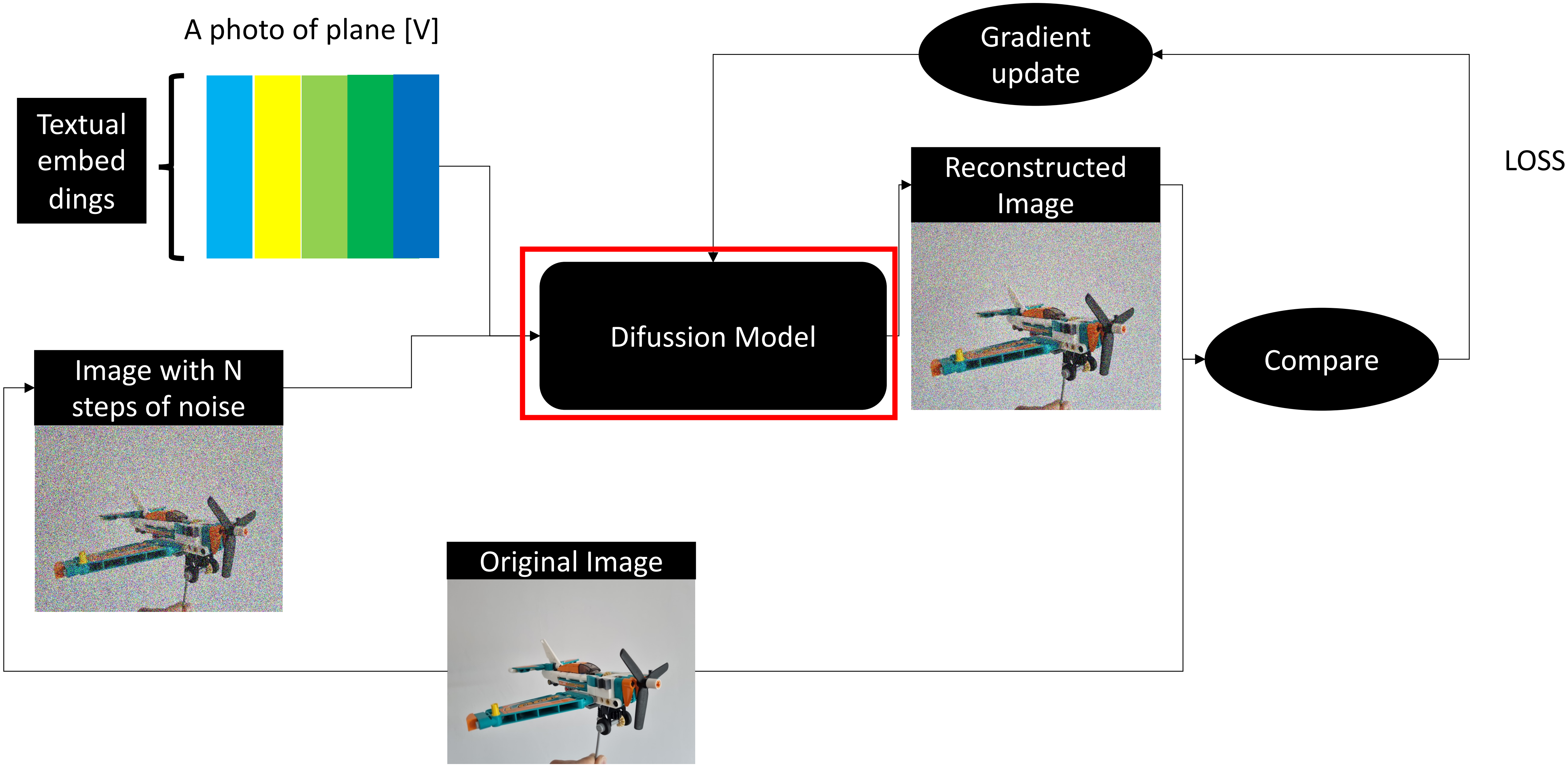



Fine-tuning approach

Textual Inversion



DreamBooth

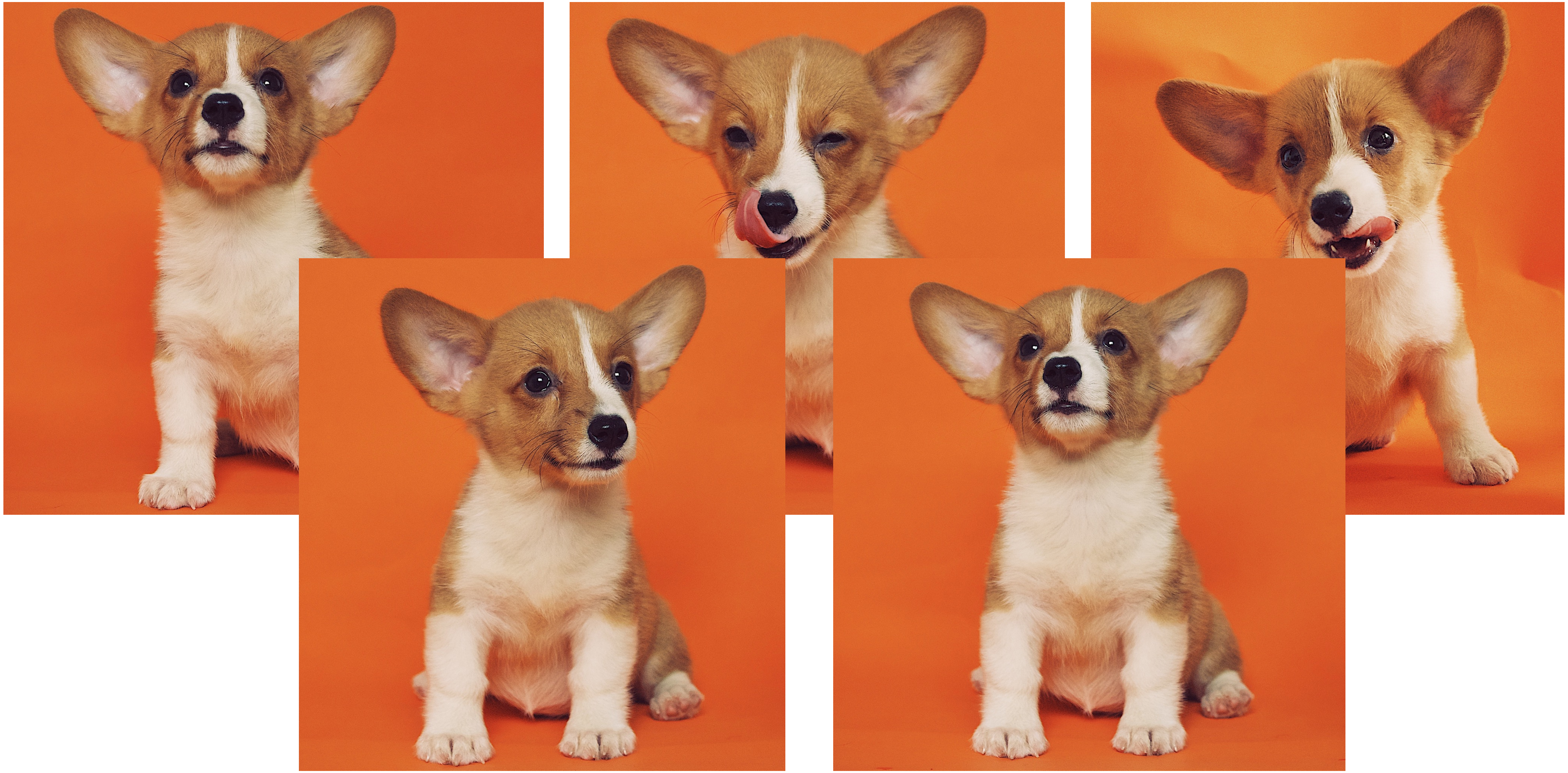

A photo of plane [V]
Textual embed dings
Gradient update
Reconstructed Image
LOSS
Difussion Model
Compare
Image with N steps of noise
Original Image

Textual Inversion vs DreamBooth

	Textual Inversion	**DreamBooth**
Effectiveness	Quite good	Subjectively better than Textual Inversion
Storage	Efficient when it comes to starage management since base model is not changed and it is only the small vector representing the learned concept that needs to be sotred.	For every new concept new big model files is created, so in case many concepts need to be learned we end up with bunch of models.

Example – DreamBooth

DreamBooth – experiment 2



Source: own material

DreamBooth – experiment 2

```
accelerate launch train_dreambooth.py \
--pretrained_model_name_or_path="stabilityai/stable-diffusion-2-1-base" \
--instance_data_dir="./data/dog" \
--output_dir="./results/dog" \
--instance_prompt="a sks dog" \
--with_prior_preservation \
--class_prompt="a dog" \
--class_data_dir="./results/dog_class" \
--num_class_images=1000 \
--resolution=512 \
--train_batch_size=2 \
--gradient_accumulation_steps=1 \
--learning_rate=5e-6 \
--lr_scheduler="constant" \
--lr_warmup_steps=0 \
--max_train_steps=1000
```

DreamBooth – experiment 2

```
accelerate launch train_dreambooth.py \
--pretrained_model_name_or_path="stabilityai/stable-diffusion-2-1-base" \
--instance_data_dir="./data/dog" \
--output_dir="./results/dog" \
--instance_prompt="a sks dog" \
--with_prior_preservation \
--class_prompt="a dog" \
--class_data_dir="./results/dog_class" \
--num_class_images=1000 \
--resolution=512 \
--train_batch_size=2 \
--gradient_accumulation_steps=1 \
--learning_rate=5e-6 \
--lr_scheduler="constant" \
--lr_warmup_steps=0 \
--max_train_steps=1000
```

DreamBooth – experiment 2

Prompt: SKS dog in a forest

Prompt: dog in a forest

Source: own material

DreamBooth – experiment 2

Prompt: SKS dog over pyramids

Prompt: dog over pyramids

Source: own material

DreamBooth – experiment 2

Prompt: SKS dog underwater

Prompt: SKS dog sleeping

Source: own material

DreamBooth – experiment 2

Prompt: SKS dog in red dress

Prompt: dog in red dress

Source: own material

DreamBooth – experiment 2

Prompt: sketch of SKS dog

Prompt: SKS dog by Picasso

Prompt: dog by Picasso

Source: own material