# Images & Text

Rafail Fridman for DL4CV

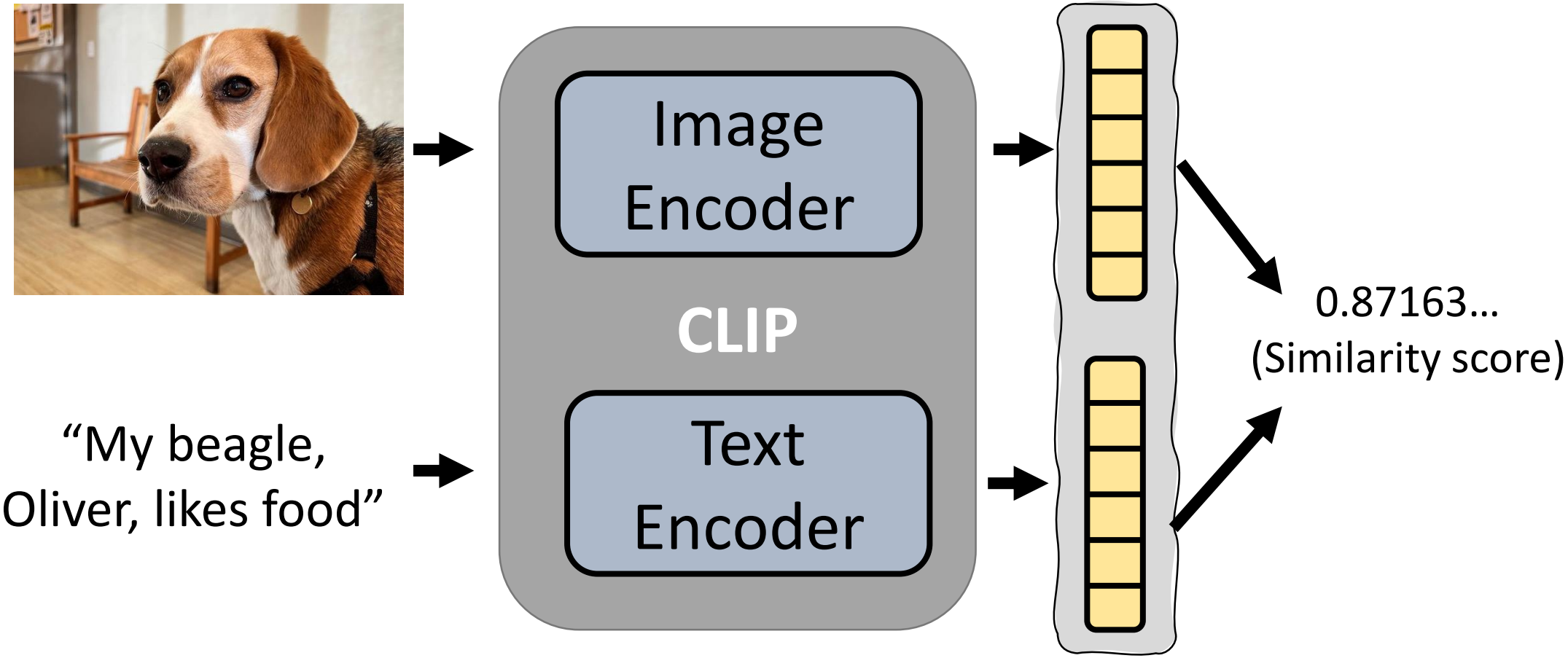
08.02.2023

# Topics

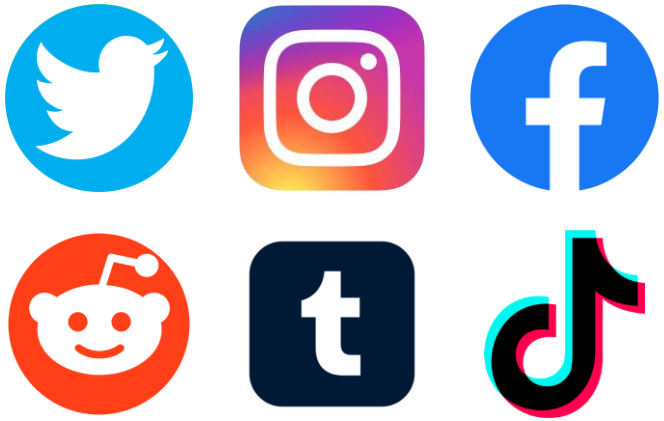
1. CLIP-guided optimization:
    - VQ-GAN + CLIP
    - StyleCLIP
    - Text2LIVE
2. Diffusion Models + text
    - Text conditioning in Diffusion Models
    - Classifier (free) guidance
    - Latent Diffusion models

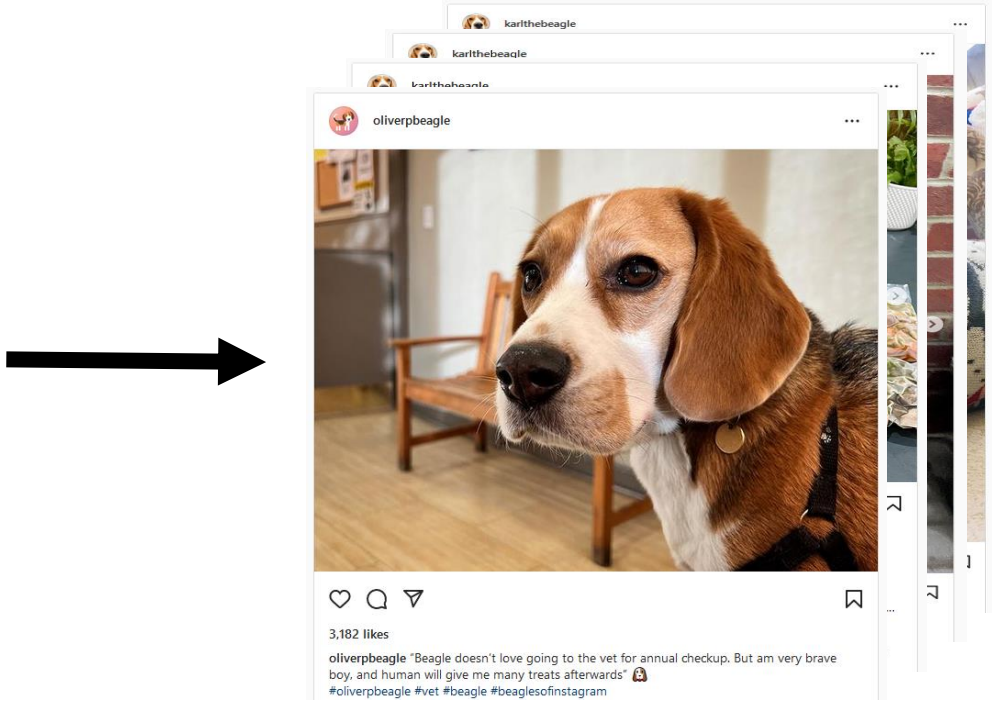
# CLIP - reminder

- **C**ontrastive **L**anguage **I**mage **P**retraining



# CLIP - reminder






(  , **oliverpbeagle** "Beagle doesn't love going to the vet for annual checkup. But am very brave boy, and human will give me many treats afterwards" 🐶 #oliverpbeagle #vet #beagle #beaglesofinstagram ) $\times 400$ Million

# CLIP for generative tasks?

"My beagle, Oliver, likes food"

Image Encoder

CLIP

Text Encoder

$E_{im}$

$E_{txt}$

$$\frac{E_{im} \cdot E_{txt}}{||E_{im}||||E_{txt}||}$$

# Using CLIP for generative tasks

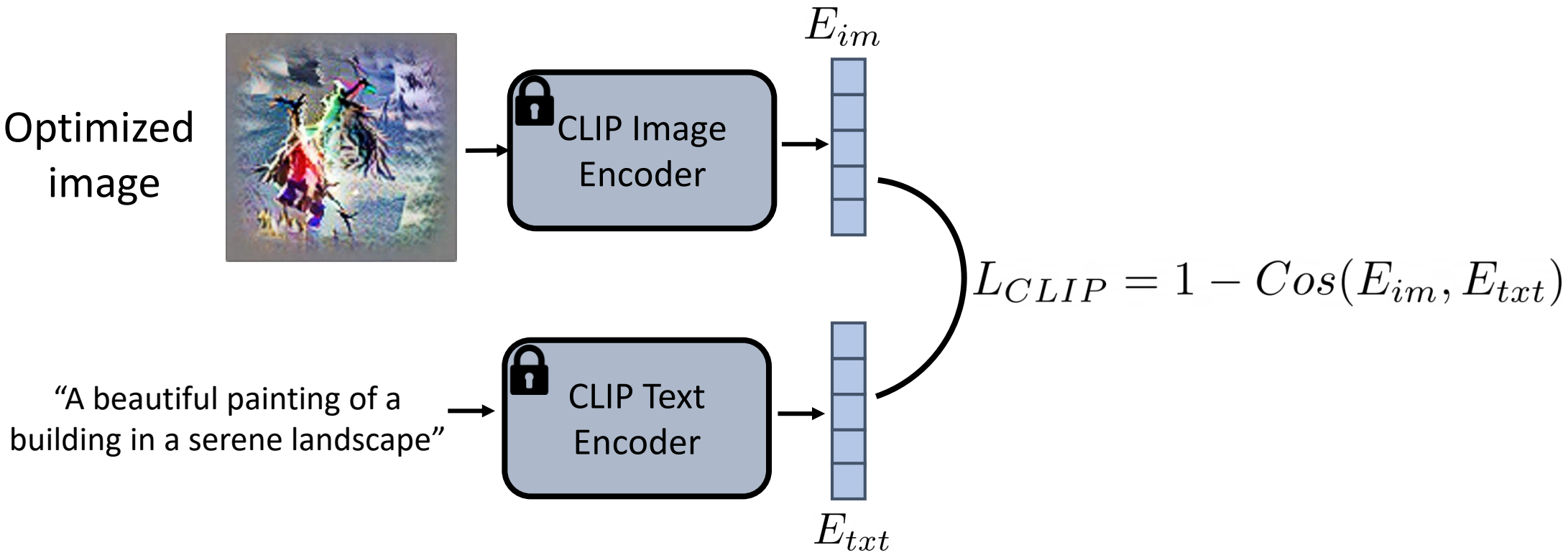
**Generation**

A beautiful painting of a building in a serene landscape

**Editing**

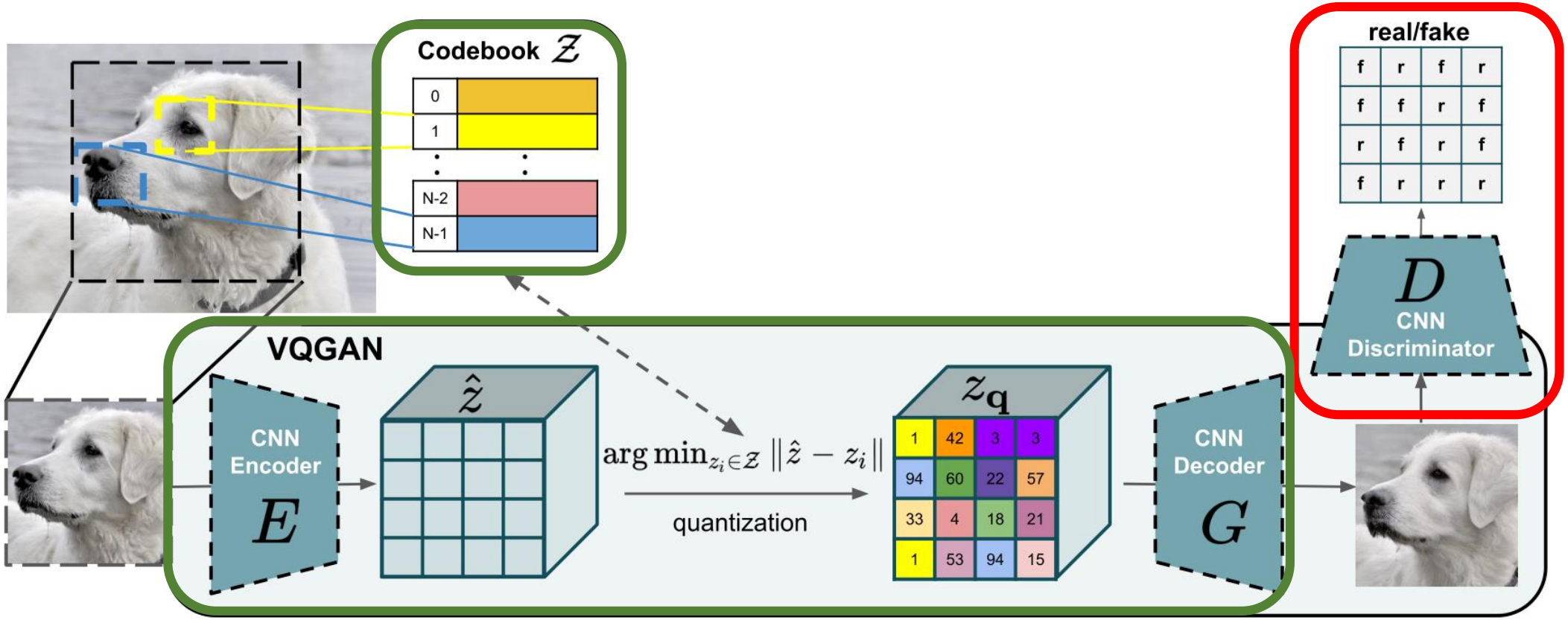
"A cake made of ice"

# Naive approach



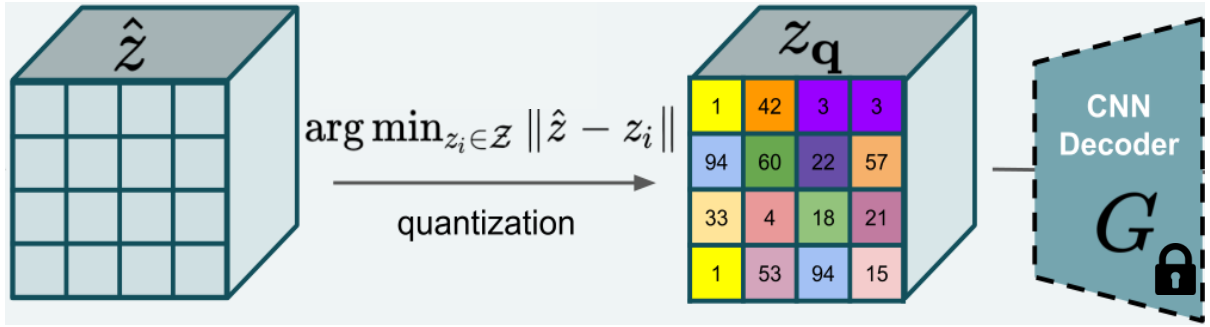

**Utilize CLIP to steer the generation towards the desired text prompt**

# VQ-GAN - reminder



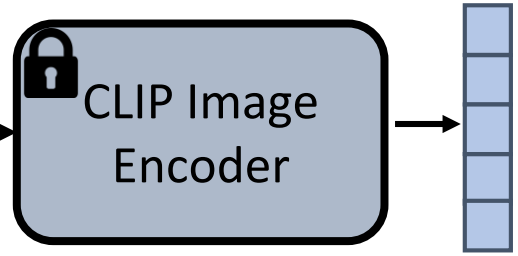

VQ-VAE + GAN

# VQGAN + CLIP




Optimized latent

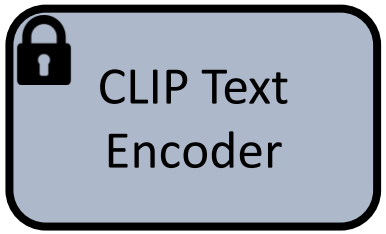



$E_{im}$

$L_{CLIP}$

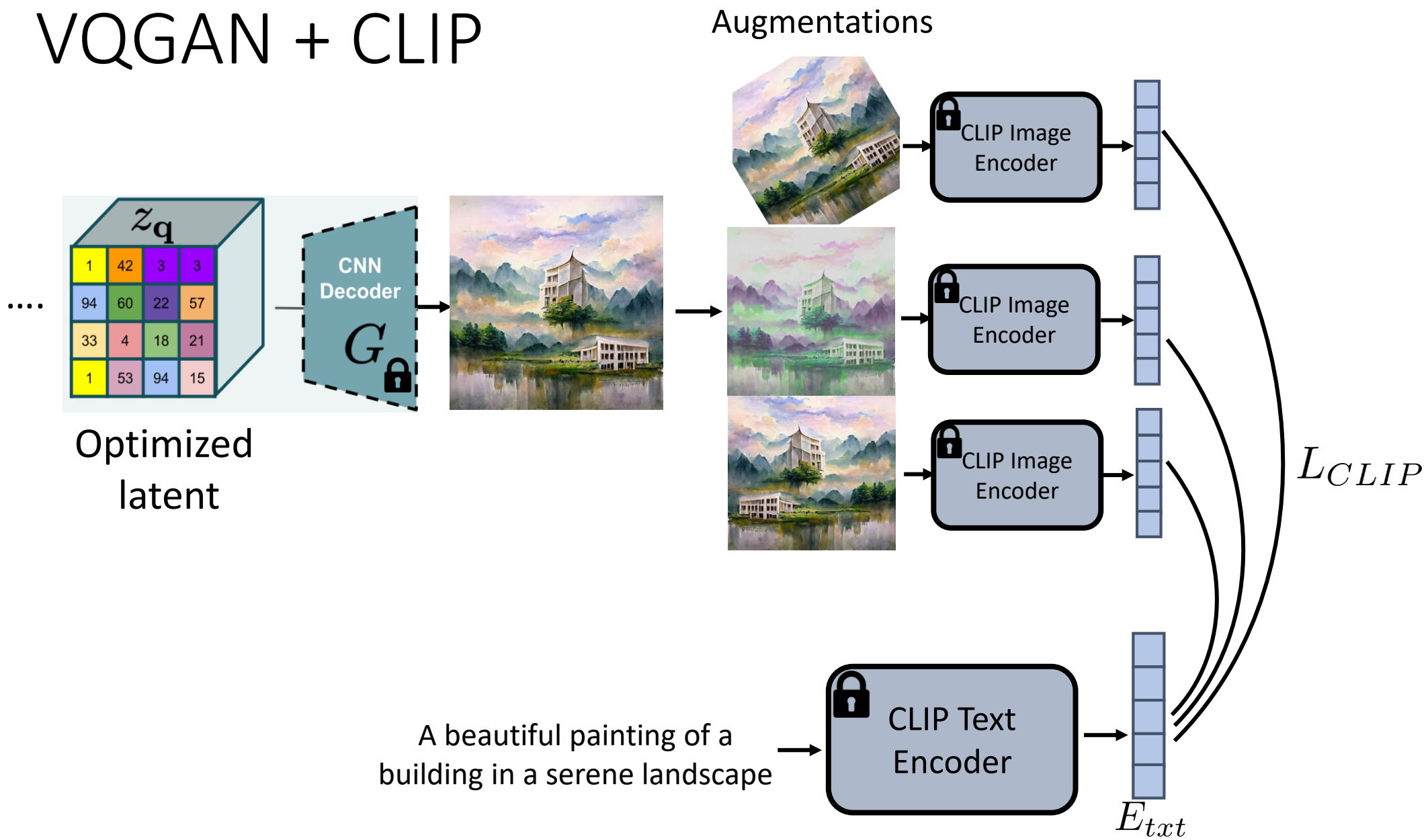
A beautiful painting of a building in a serene landscape



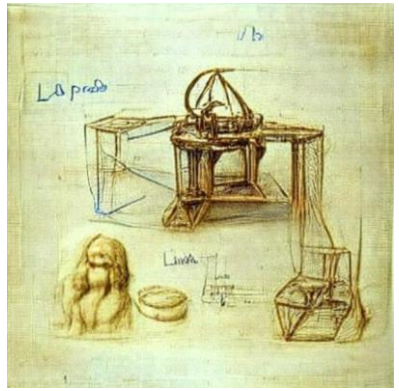

$E_{txt}$

$$L_{CLIP} = 1 - Cos(E_{im}, E_{txt})$$

# VQGAN + CLIP


Augmentations
....
$z_q$
1 42 3 3
94 60 22 57
33 4 18 21
1 53 94 15
CNN Decoder
$G$
Optimized latent
CLIP Image Encoder
CLIP Image Encoder
CLIP Image Encoder
$L_{CLIP}$
A beautiful painting of a building in a serene landscape
CLIP Text Encoder
$E_{txt}$

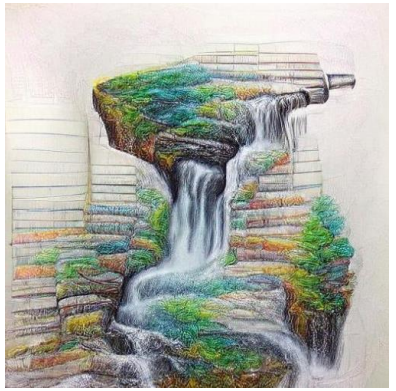
# VQGAN + CLIP results



"A sketch of 3D printer by da Vinci"

"An autogyro flying, artstation"

"A futuristic city in synthwave style"



"A colored pencil drawing of a waterfall"

"A painting of a city in a deep valley"

"Baba Yaga's house, fantasy art"

# Using CLIP for generative tasks

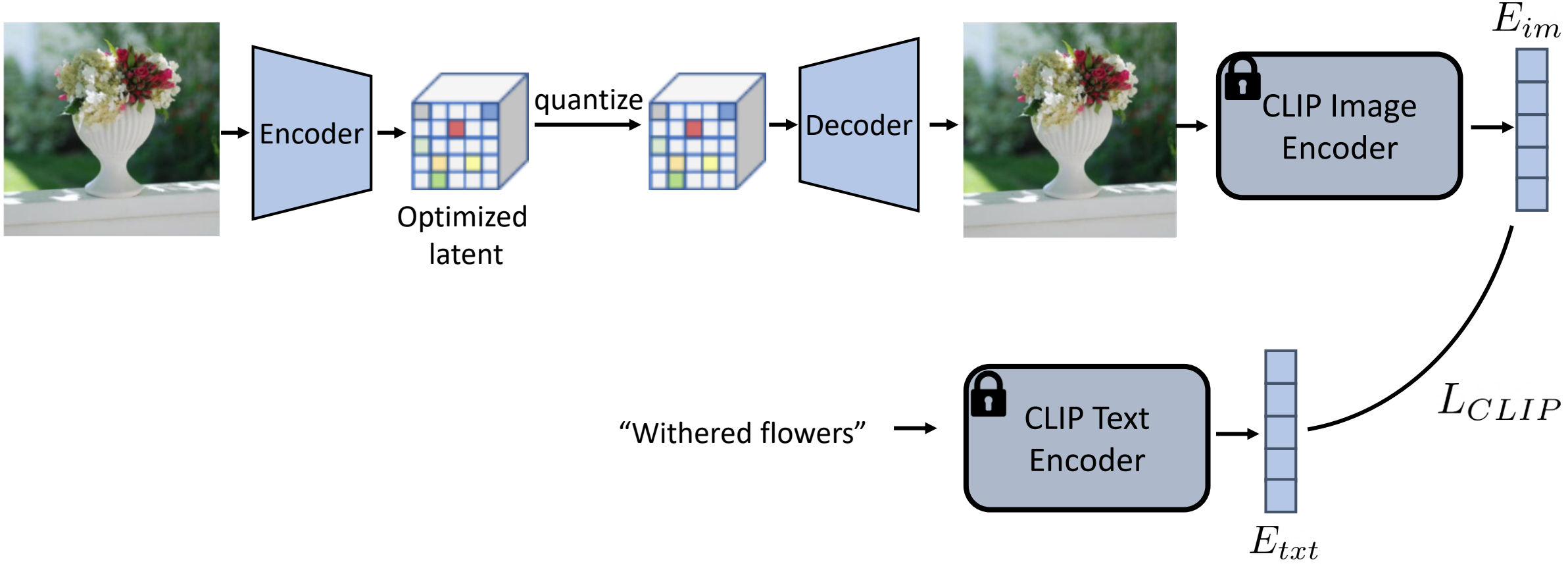
**Generation**

A beautiful painting of a building in a serene landscape

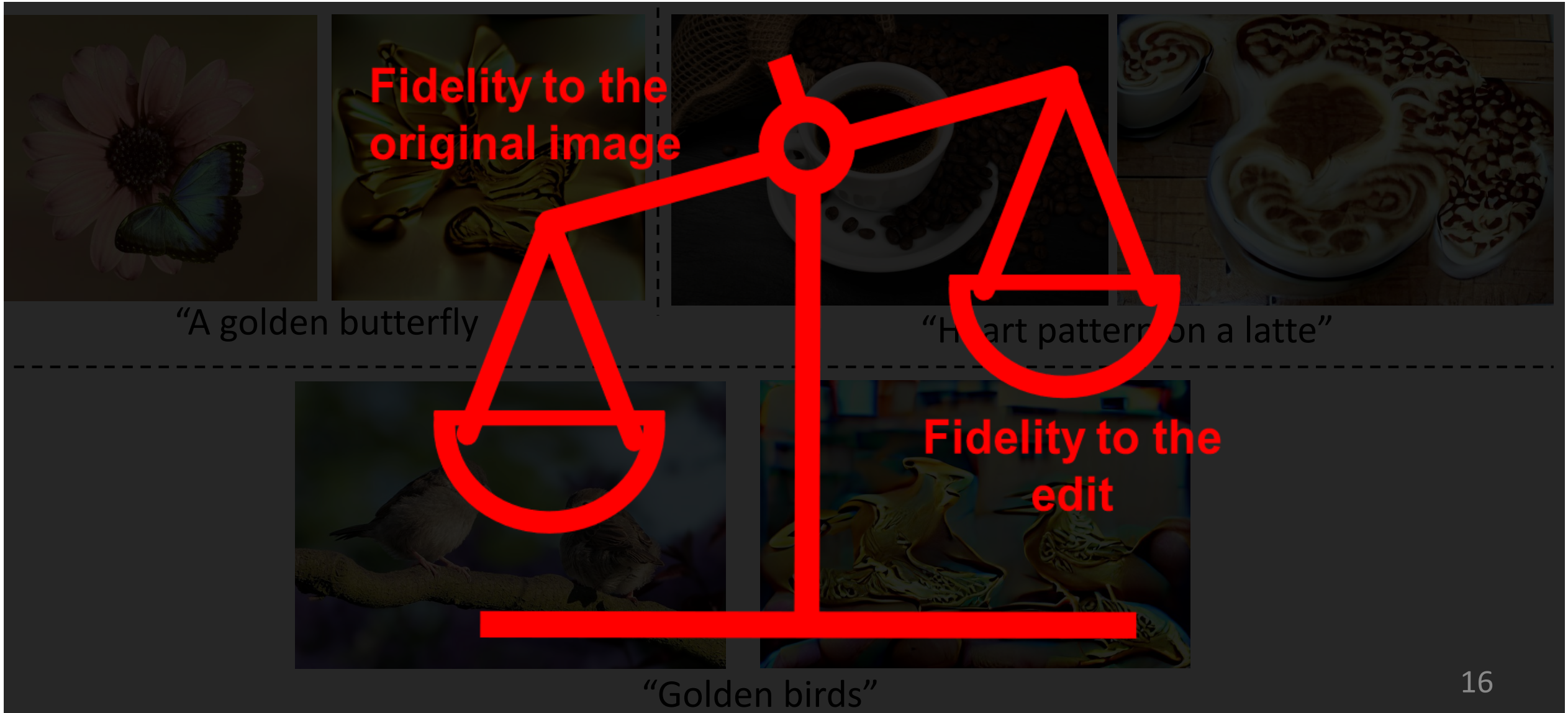
**Editing**

"A cake made of ice"

# VQGAN + CLIP editing



# VQGAN + CLIP editing



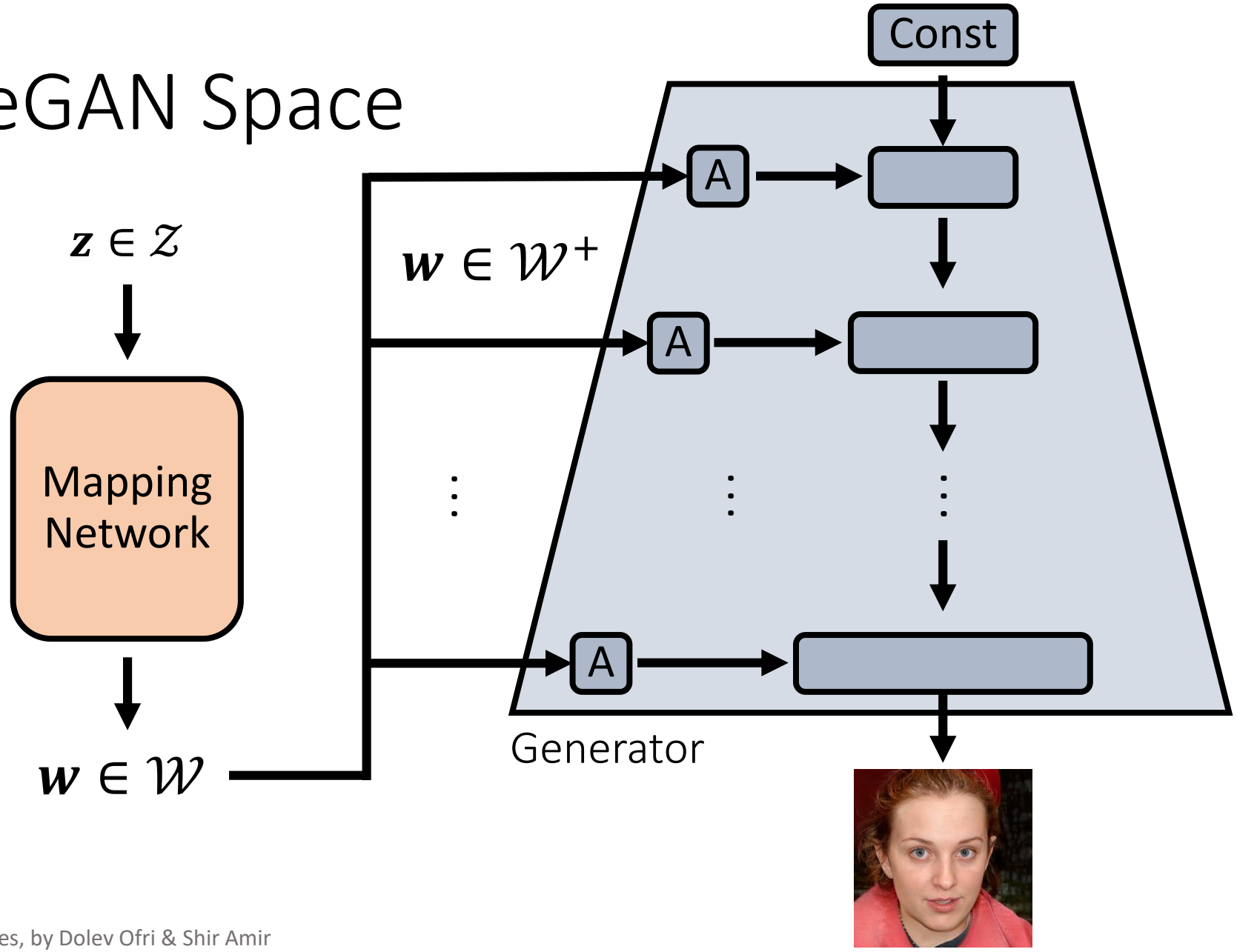
# StyleCLIP – goal

"curly hair" "hi-top fade hair" "fringe hair" "black hair" "makeup"

"happy" "big eyes"

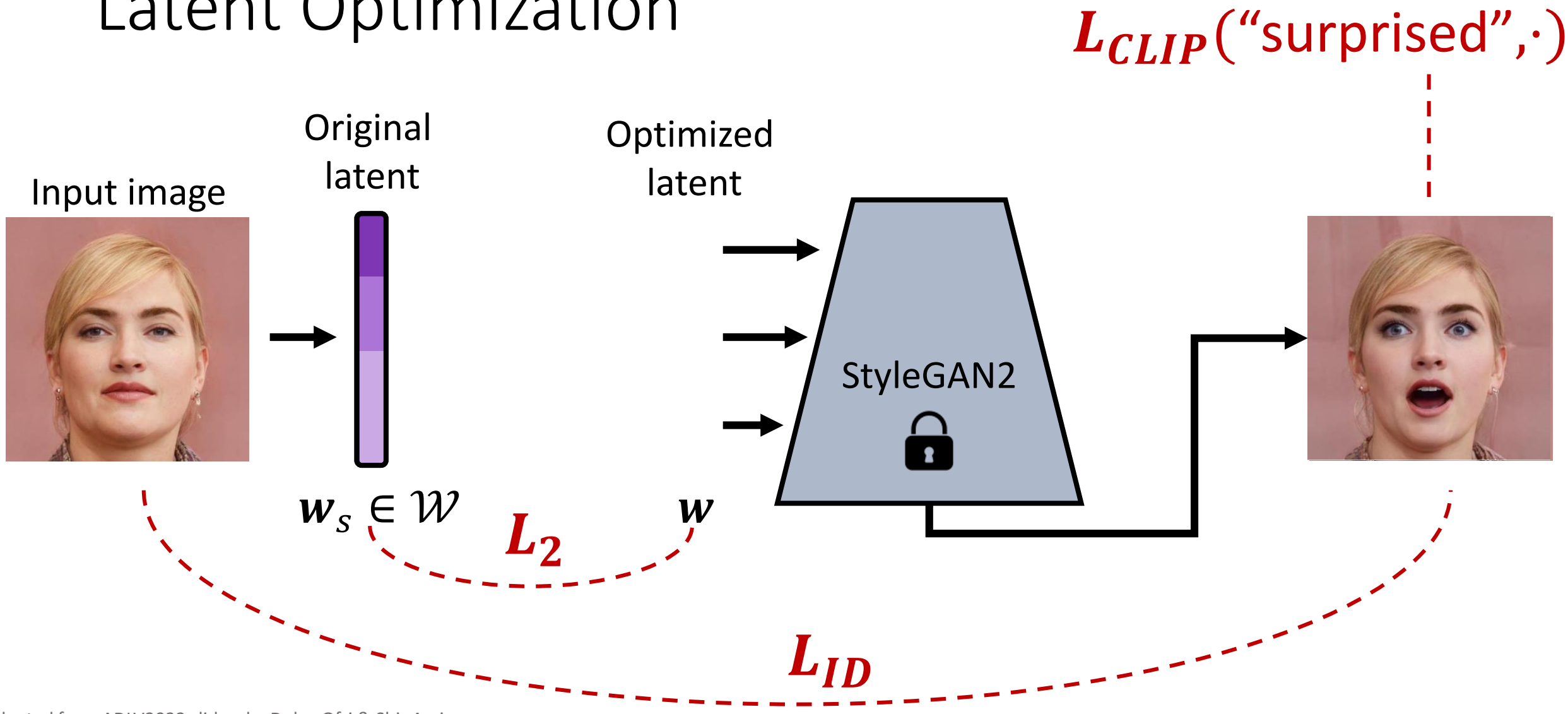
"cloud" "spires"

# StyleGAN Space




Adapted from ADLV2022 slides, by Dolev Ofri & Shir Amir

# Latent Optimization




Adapted from ADLV2022 slides, by Dolev Ofri & Shir Amir

# StyleCLIP - results

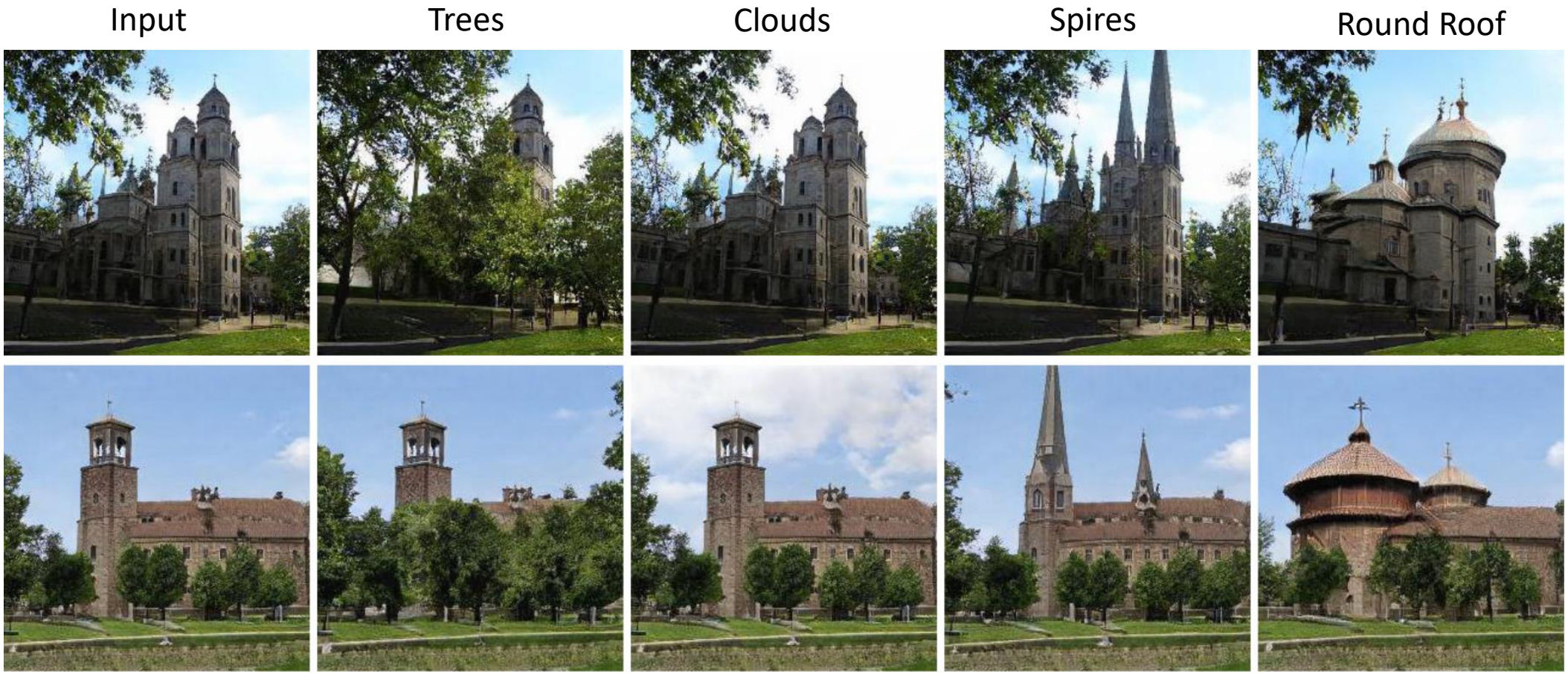
Input | "Beyonce" | "A woman without makeup" | "Elsa from Frozen"

Input | "A man with a beard" | "A blonde man" | "Donald Trump"

CLIP knows about these!

# StyleCLIP – results, different domains

# StyleCLIP – results, different domains



# Text2LIVE – goal

Original image

*“oreo cake”*

*“brioche”*

*“ice”*

*“spinach moss cake”*

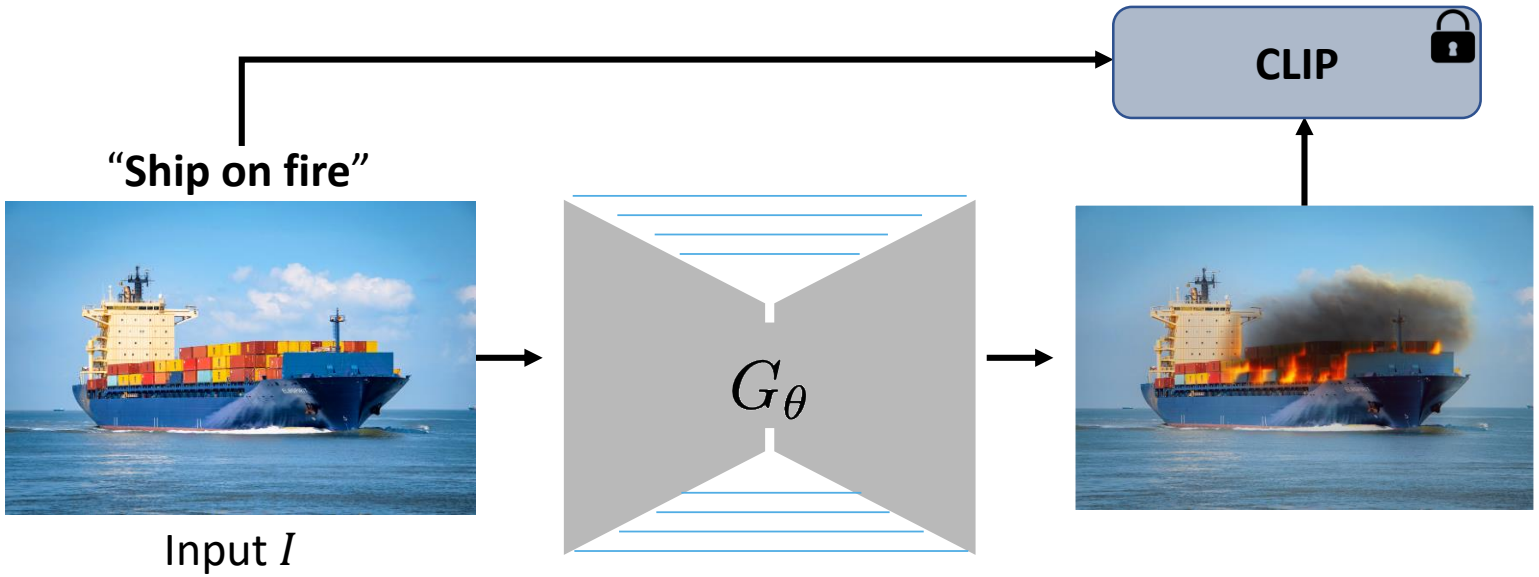
Original video

“stained glass giraffe”

“giraffe with neck warmer”

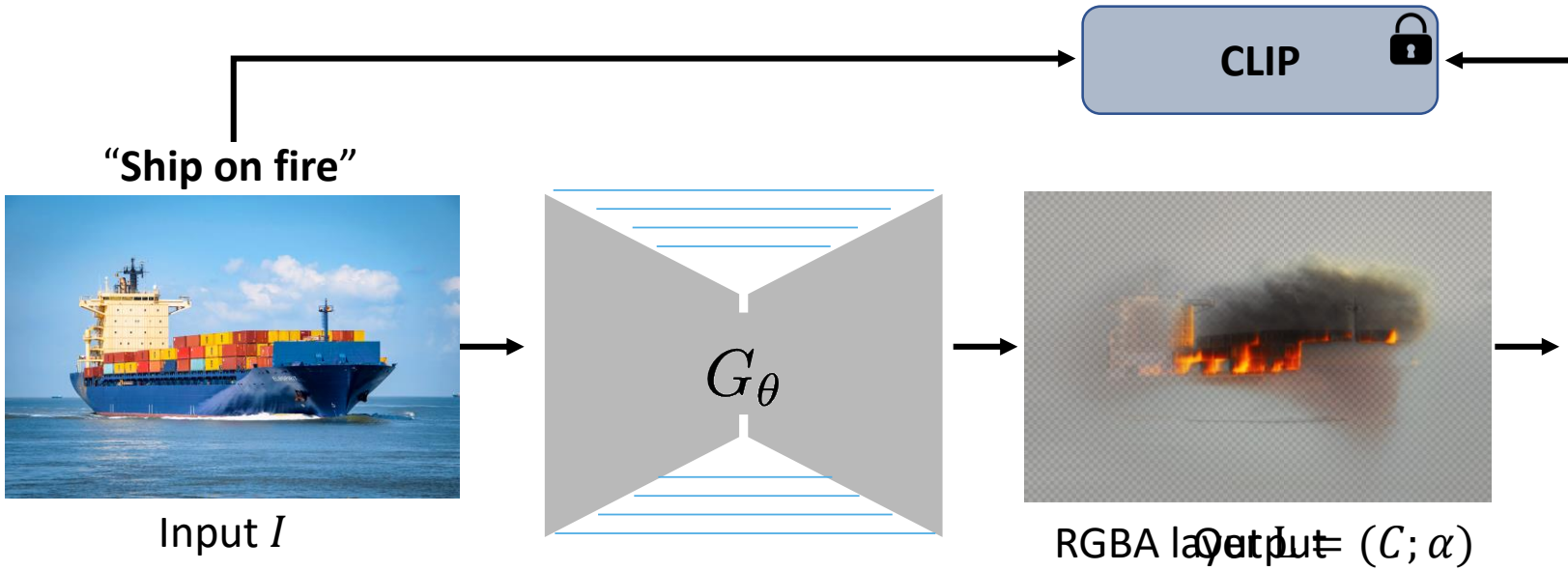
“giraffe with hairy colorful mane”

# Text2LIVE – method




**Augmentations**

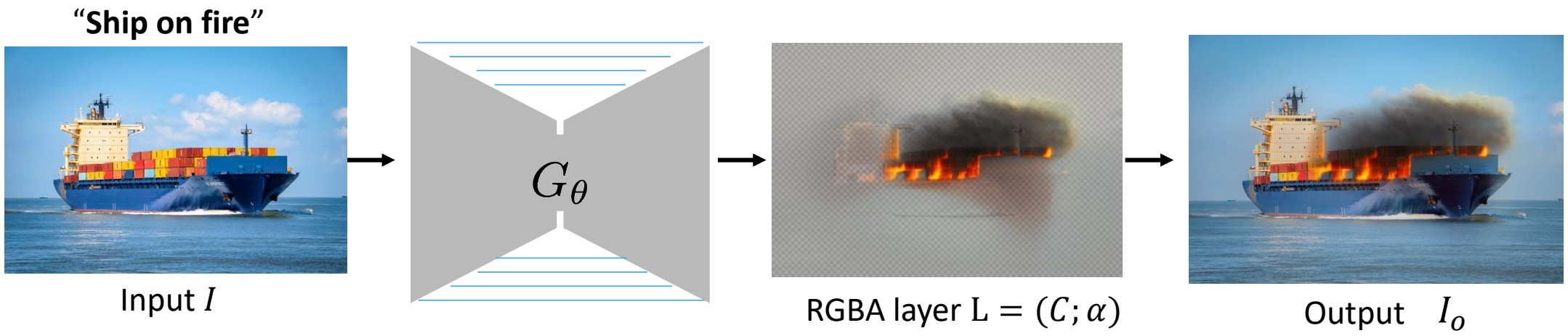
# Text2LIVE – method



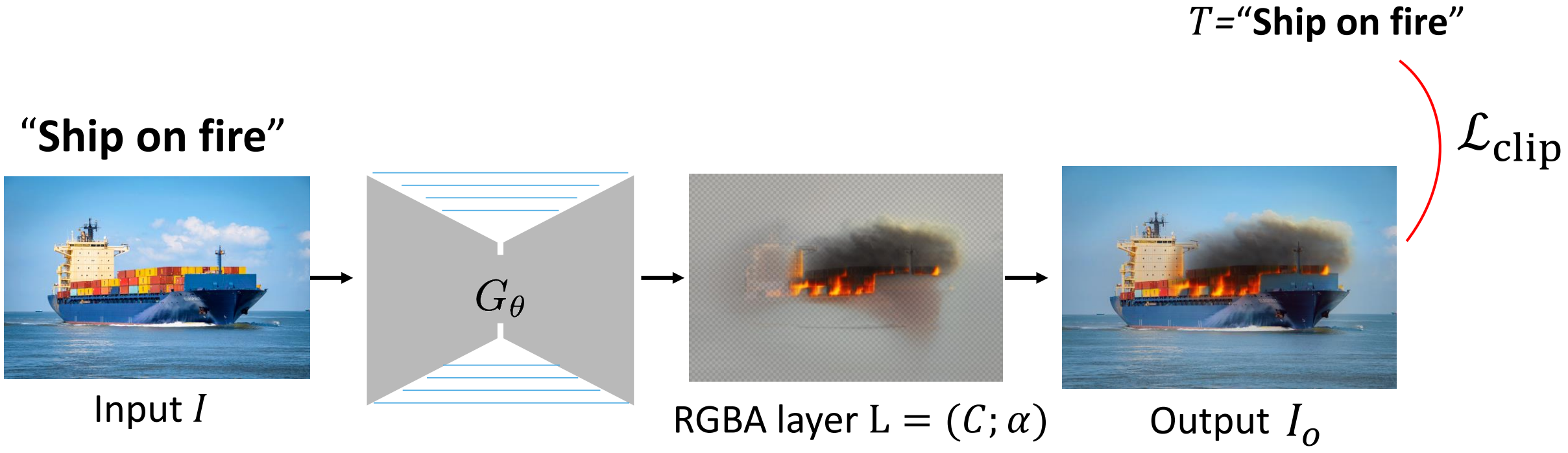

$$I_o = \alpha \cdot C + (1 - \alpha) \cdot I$$

# Text2LIVE – losses

$$\mathcal{L}_{Text2LIVE} = \underbrace{\mathcal{L}_{\text{comp}}(I_o)}_{\text{Final composite}} + \underbrace{\alpha\mathcal{L}_{screen}(C,\alpha) + \gamma\mathcal{L}_{\text{sparsity}}(\alpha)}_{\text{Edit layer}}$$



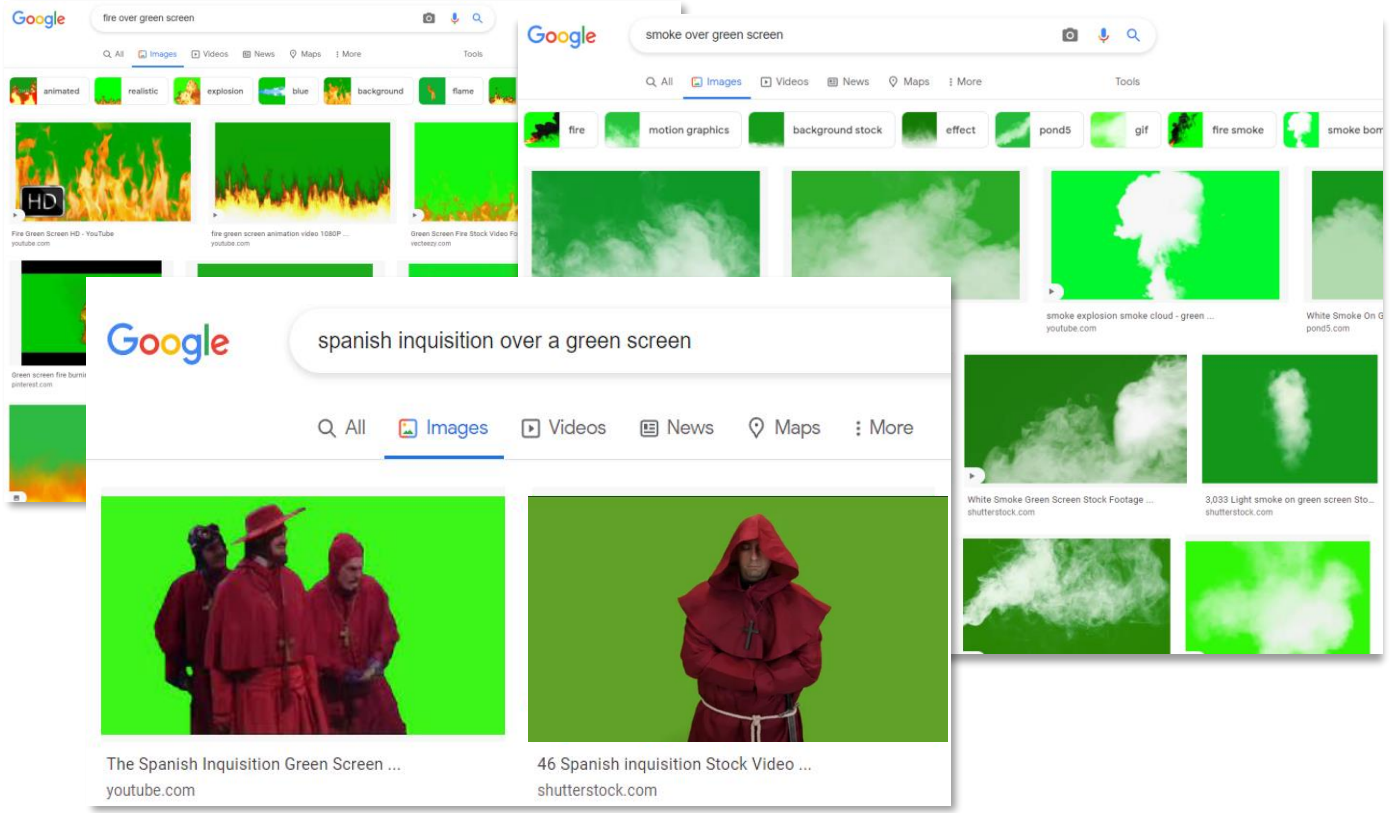
# Loss on the composition



# Losses on the edit layer

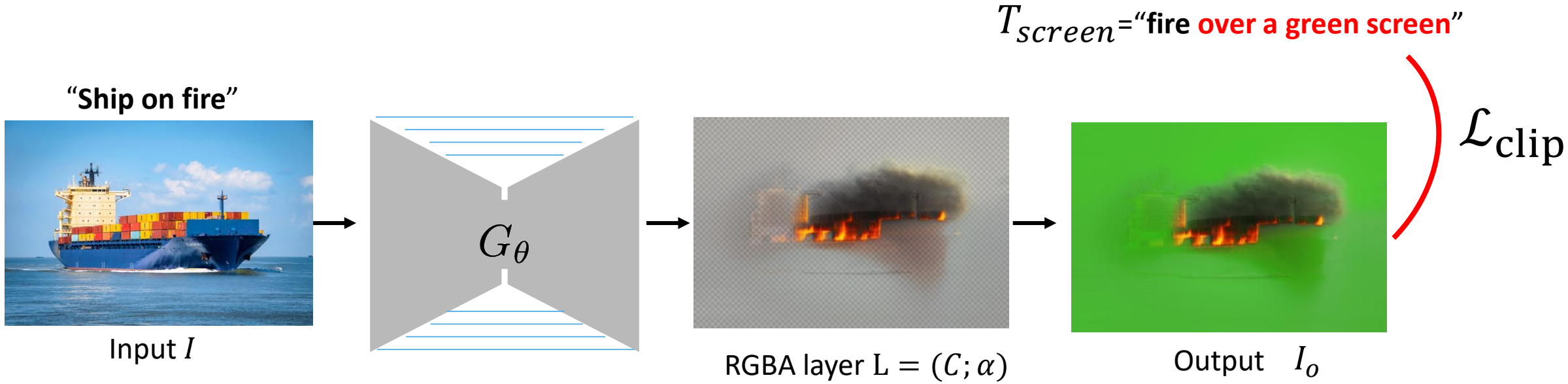
$$\mathcal{L}_{Text2LIVE} = \mathcal{L}_{\text{comp}}(I_o) + \boldsymbol{\alpha}\boldsymbol{\mathcal{L}_{screen}}\,(\boldsymbol{C}, \boldsymbol{\alpha}) + \gamma\mathcal{L}_{\text{sparsity}}(\alpha)$$

Chroma keying



# Text2LIVE – losses

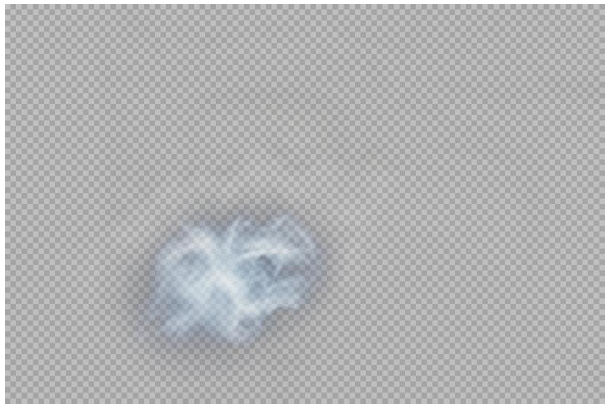
$$\mathcal{L}_{Text2LIVE} = \mathcal{L}_{\text{comp}}(I_o) + \boldsymbol{\alpha}\boldsymbol{\mathcal{L}_{screen}}\,(\boldsymbol{C}, \boldsymbol{\alpha}) + \gamma\mathcal{L}_{\text{sparsity}}(\alpha)$$



# Losses on the edit layer

"***A man smoking a cigar***"

w/ $\mathcal{L}_{screen}$ ($T$ ="smoke")

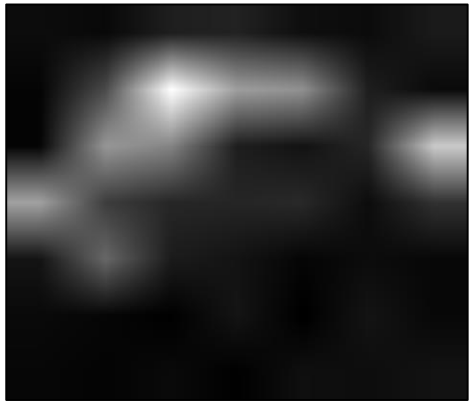


w/o $\mathcal{L}_{screen}$

# Losses on the edit layer

Input Image

“woman wearing a red hat”



Relevancy map*

“**hat**”

Text2LIVE output matte

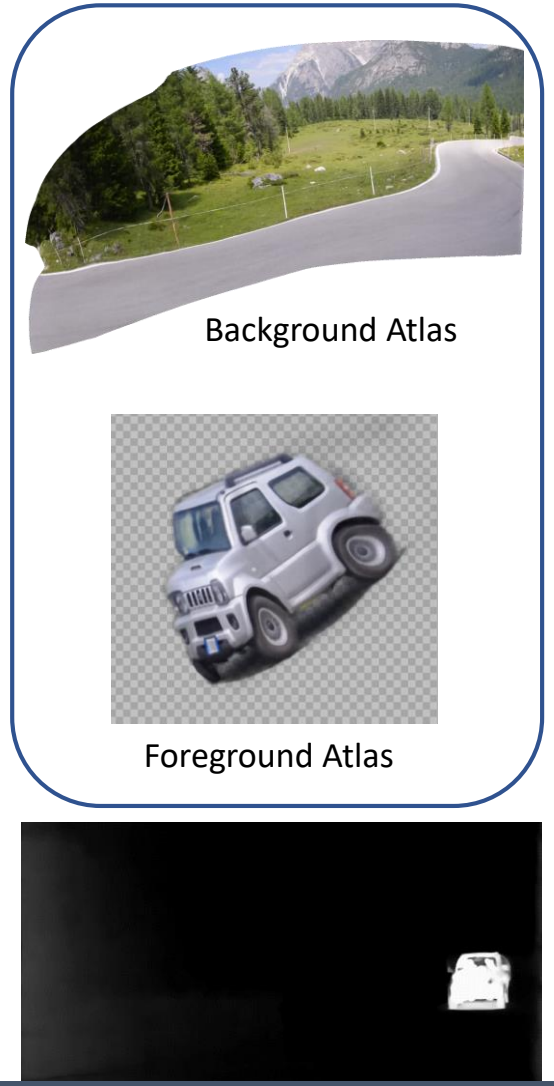
$MSE$

Text2LIVE result

# Text2LIVE – results

# NLA - reminder

Input video




Render Video

$$c^{\boldsymbol{p}} = (1 - \alpha^{\boldsymbol{p}})c_b^{\boldsymbol{p}} + \alpha^{\boldsymbol{p}} c_f^{\boldsymbol{p}}$$

Video Reconstruction

# NLA - reminder

Input video

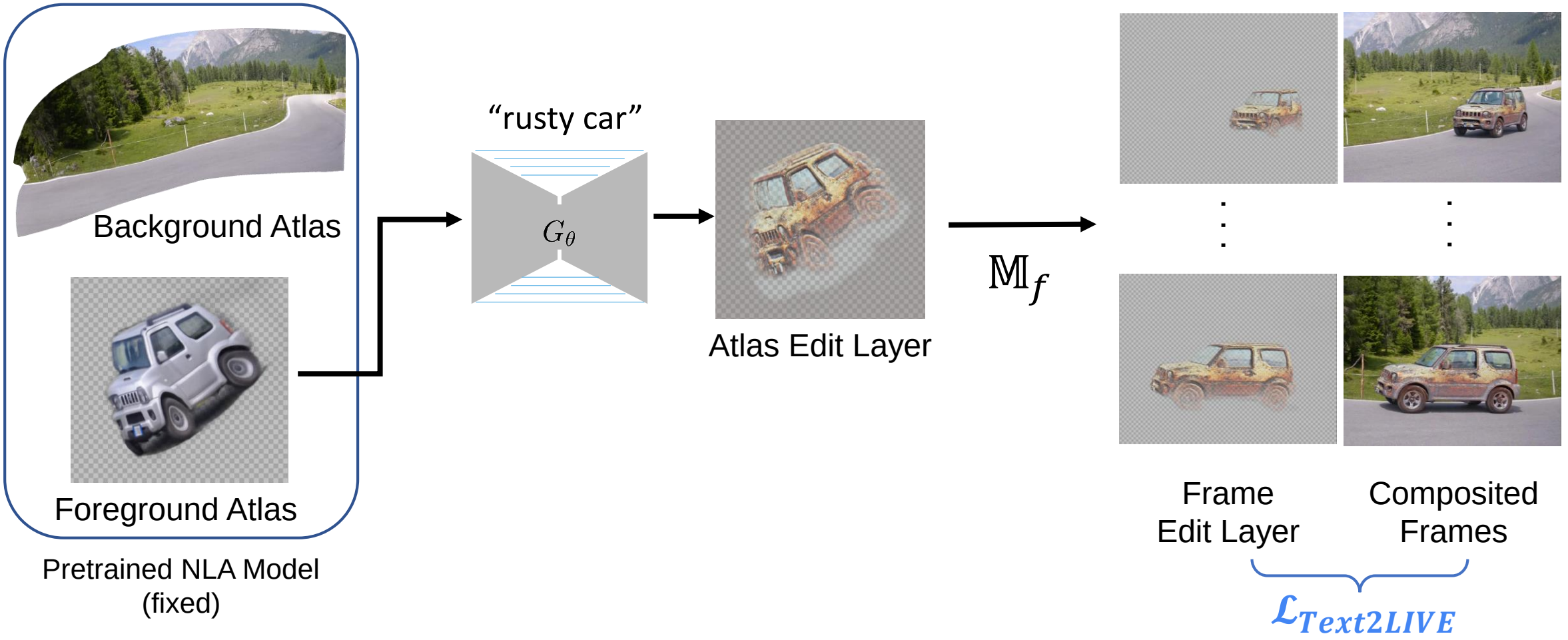

Render Video

$$c^{\boldsymbol{p}} = (1 - \alpha^{\boldsymbol{p}})c_b^{\boldsymbol{p}} + \alpha^{\boldsymbol{p}} c_f^{\boldsymbol{p}}$$

Video Reconstruction

*Editing **huge pixel volume** → Editing a **single 2D image***

# Text2LIVE video editing



# Text2LIVE – results

Input Video

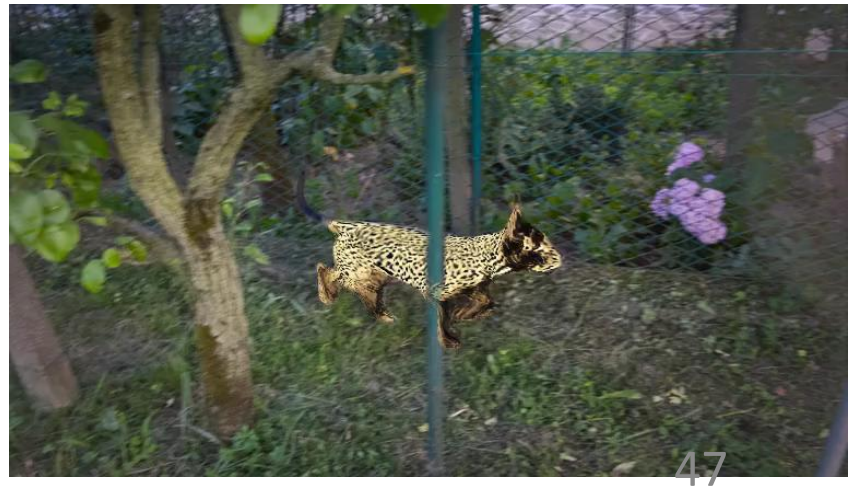
“swarovski blue crystal swan”

Original Video

Input Video

“dalmatian dog”

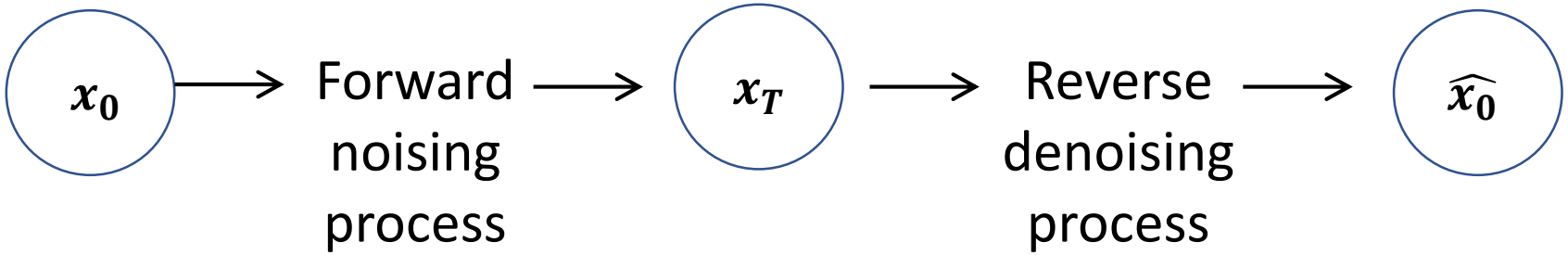
“dog with leopard texture”



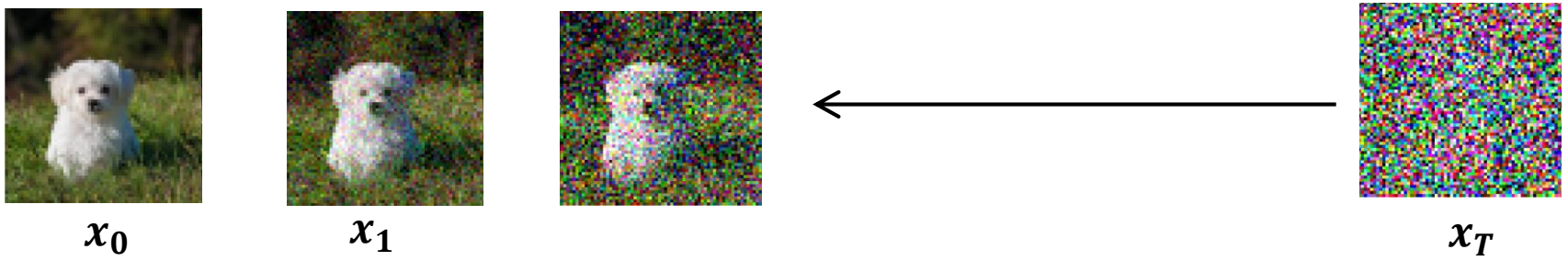
# Topics

1. CLIP-guided optimization:
    - VQ-GAN + CLIP
    - StyleCLIP
    - Text2LIVE
2. **Diffusion Models + text**
    - **Text conditioning in Diffusion Models**
    - **Classifier (free) guidance**
    - **Latent Diffusion models**

# Diffusion Models - reminder



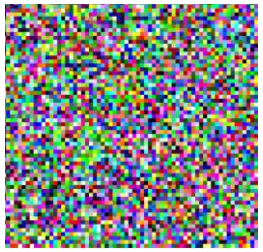

Reverse denoising process




Generating samples by by gradually reducing noise

**Reminder:** $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$

# Sampling Algorithm



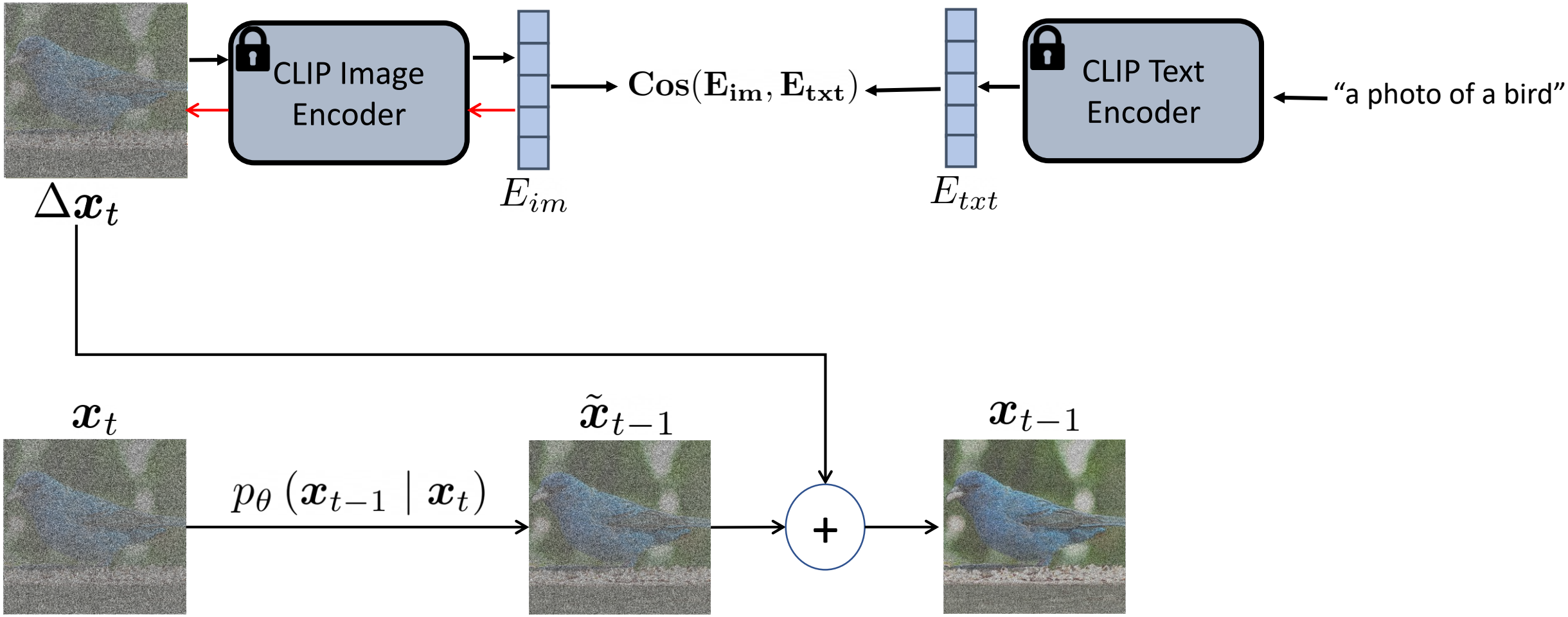
Neural Net

1. Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. For $t = T, \ldots, 1$:
   - "Predict noise" in current image $\mathbf{z}_\theta(\mathbf{x}_t, t)$
   - Sample variance noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
   - Sample from reverse distribution:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right) + \tilde{\beta}_t \mathbf{z}$$

3. Return $\mathbf{x}_0$

# Conditional image generation **with CLIP**



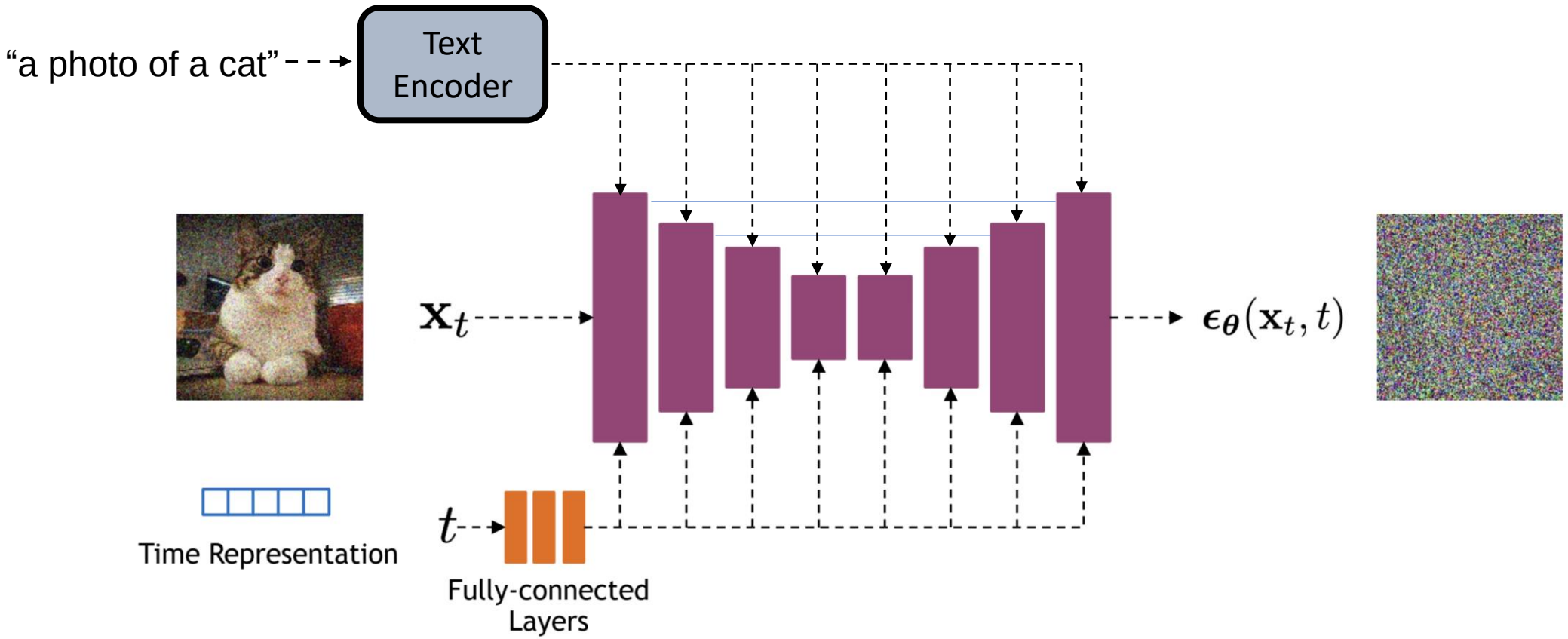
# Conditional sampling **with CLIP model**

1. Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, text prompt $c$, text encoder $f$, image encoder $g$
2. For $t = T, \ldots, 1$:
   - "Predict noise" in current image $\mathbf{z}_\theta(\mathbf{x}_t, t)$
   - Sample variance noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
   - Sample from reverse distribution:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right) + \tilde{\beta}_t\mathbf{z} + s\,\nabla_{x_t}\left(f(x_t)\cdot g(c)\right)$$

3. Return $\mathbf{x}_0$

# Text conditioning implementation




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

# Text conditioning in Diffusion Models

Noisy image $x_t$

"A photo of a cat"

Prompt $P$

Text tokens $\psi(P)$

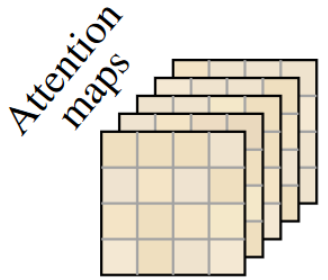
$l_K(\psi(P))$

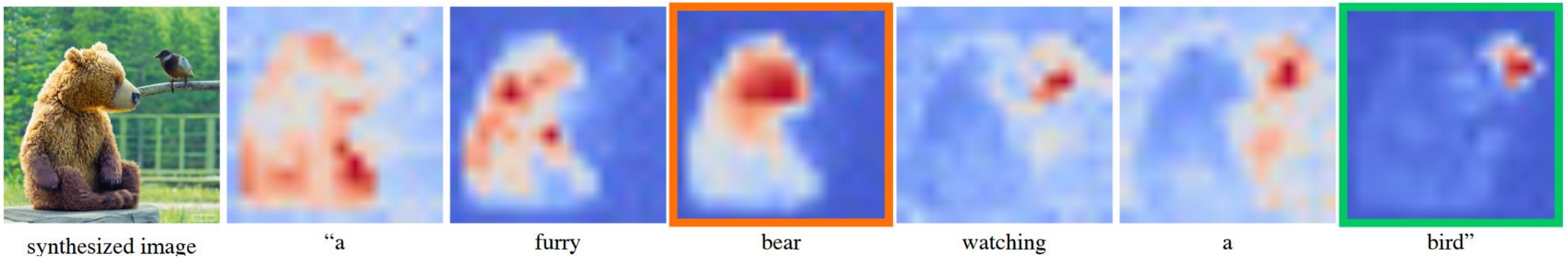
$l_V(\psi(P))$

$\phi(x_t)$

$\hat{\phi}(x_t)$

# Text conditioning in Diffusion Models

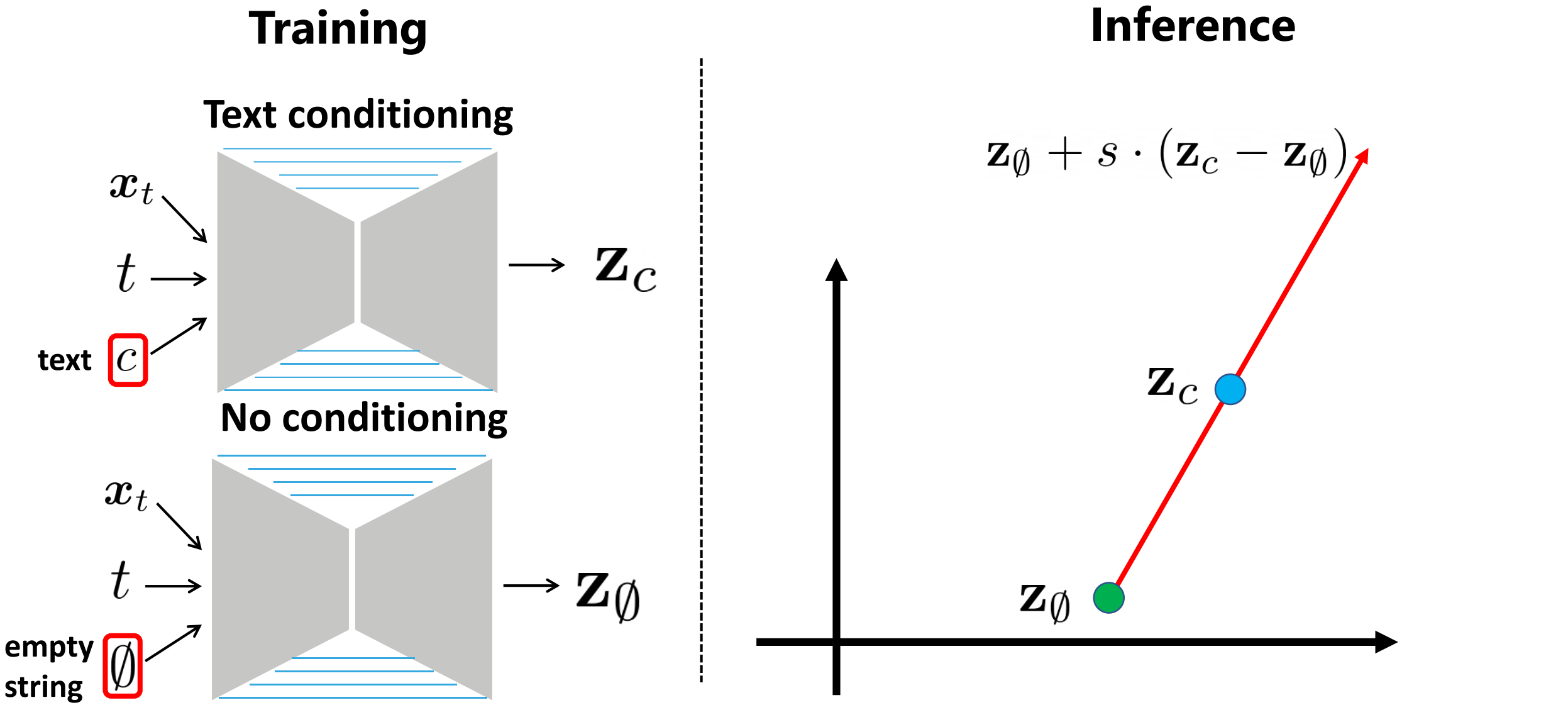



Average attention maps across all timestamps

Visual features attend to the text prompt tokens

# Classifier-free guidance



# Effect of Classifier-free guidance

**"A stained glass window of a panda eating bamboo"**

$s = 1$

$s = 3$

$$\mathbf{z}_\emptyset + s \cdot (\mathbf{z}_c - \mathbf{z}_\emptyset)$$

# Guidance comparison

**A monkey eating a banana**

**CLIP guidance**

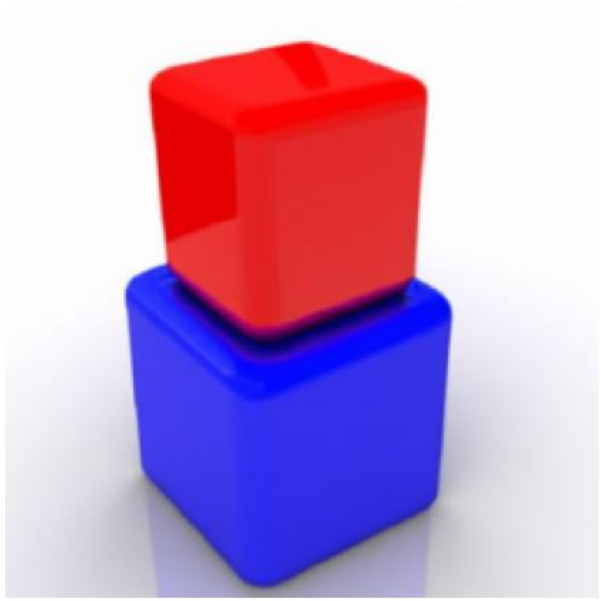
**Noised CLIP guidance**

**Classifier-free guidance**

# Results

“a hedgehog using a calculator”

“a corgi wearing a red bowtie and purple party hat”



“a red cube on top of a blue cube”

“a high-quality oil painting of a psychedelic hamster dragon”

# One small problem

- Diffusion models operate in pixel space
- Training/inference on high-resolution images:
  - Long training time
  - A lot of GPU memory

**Find a different space for diffusion models ☺**

256 x 256

OK

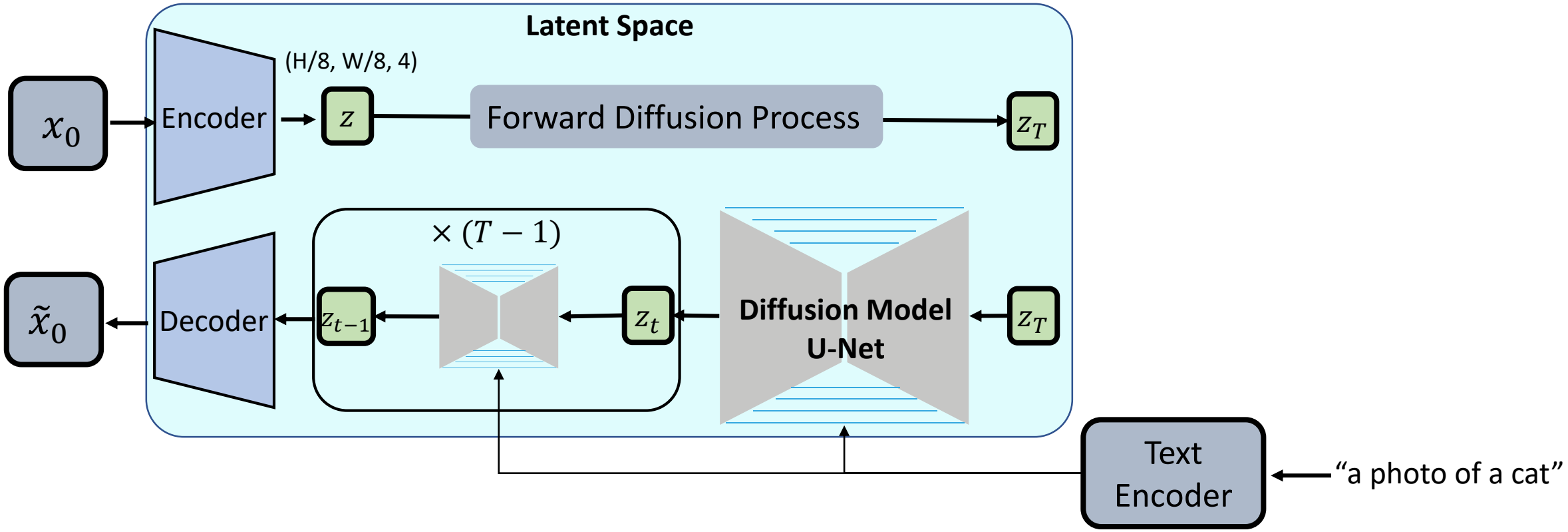
512 x 512

Fine

2048 x 2048

Does not fit in memory!

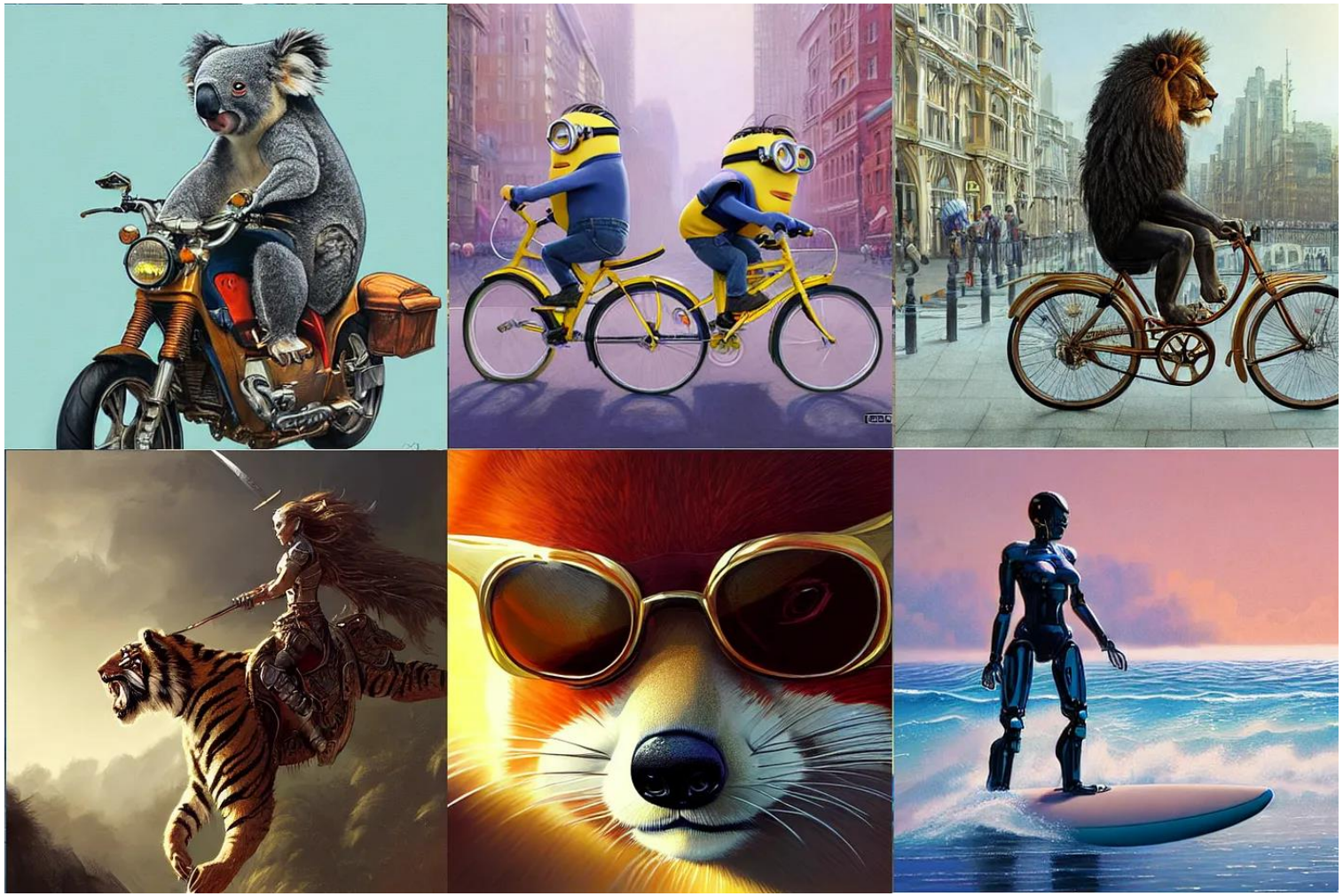
# Latent Diffusion Models




Compress images with encoder → diffusion steps → decompress with a decoder

Everything as usual, except for training – it is done in **latent space**

# Results



# Summary

- A cross-modal network (CLIP) trained for discriminative tasks, can be used for generation as well.

- To generate images from text we need a generative prior:
  - VQ-GAN
  - StyleGAN
  - DIP
  - Diffusion Models

# What we had today

1. CLIP-guided optimization:
   - VQ-GAN + CLIP
   - StyleCLIP
   - Text2LIVE
2. Diffusion Models + text
   - Text conditioning in Diffusion Models
   - Classifier (free) guidance
   - Latent Diffusion models

**Questions?**