Images & Text

Rafail Fridman

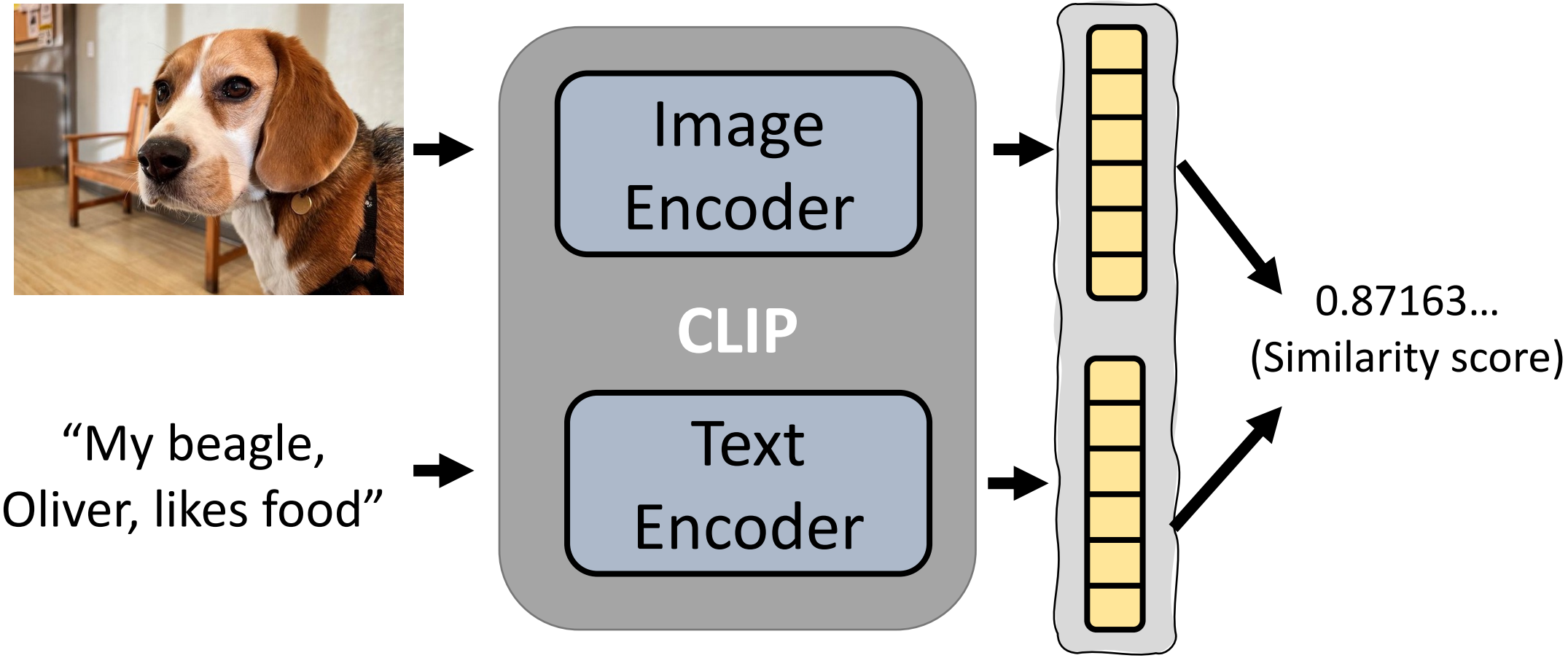
2.02.2024

Topics

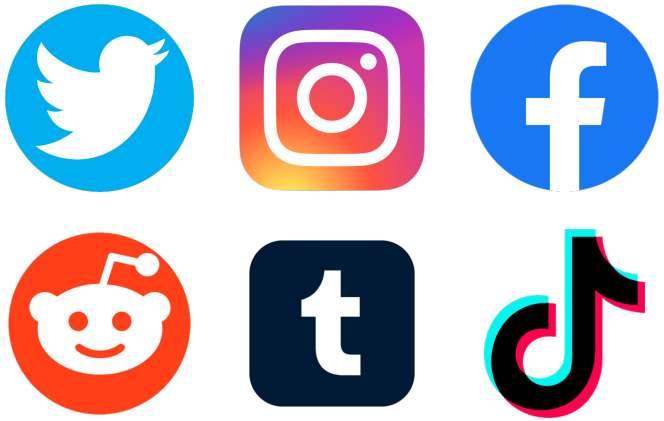
1. CLIP-guided optimization:
    - VQ-GAN + CLIP
    - StyleCLIP
    - Text2LIVE
2. Diffusion Models + text
    - Text conditioning in Diffusion Models
    - Classifier (free) guidance
    - Latent Diffusion models

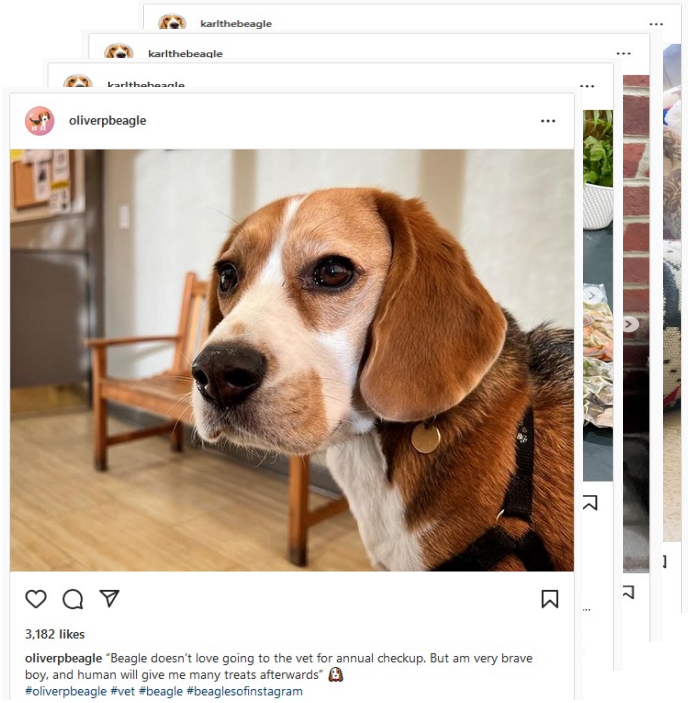
CLIP - reminder

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



CLIP - reminder

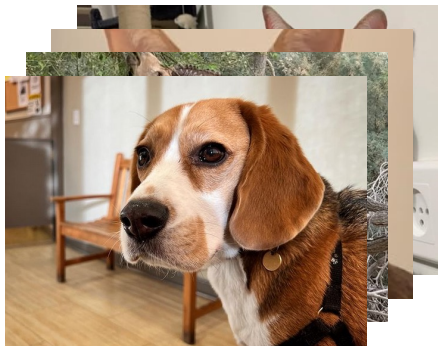




( , **oliverpbeagle** "Beagle doesn't love going to the vet for annual checkup. But am very brave boy, and human will give me many treats afterwards" 🐶 #oliverpbeagle #vet #beagle #beaglesofinstagram) ×400 Million

CLIP - reminder

$$\mathcal{L}_{InfoNCE} = \sum_{i=1}^{N} -\log \frac{\exp(I_i T_i/\tau)}{\sum_{j=1}^{N} \exp(I_i T_j/\tau)}$$



$I_{1\ldots N}$

oliverpbeagle
"Beagle doesn't love going to the vet for annual checkup. But am very brave boy…"

$T_{1\ldots N}$

encode images

encode texts

	T_1	T_2	T_3	$\cdots$	T_N
I_1	I_1T_1	I_1T_2	I_1T_3	$\cdots$	I_1T_N
I_2	I_2T_1	I_2T_2	I_2T_3	$\cdots$	I_2T_N
I_3	I_3T_1	I_3T_2	I_3T_3	$\cdots$	I_3T_N
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	I_NT_1	I_NT_2	I_NT_3	$\cdots$	I_NT_N

joint embedding space

CLIP for generative tasks?

"Beautiful dog, Beagle, looking at the camera"

Image Encoder

CLIP

Text Encoder

E_{im}

E_{txt}

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

Using CLIP for generative tasks

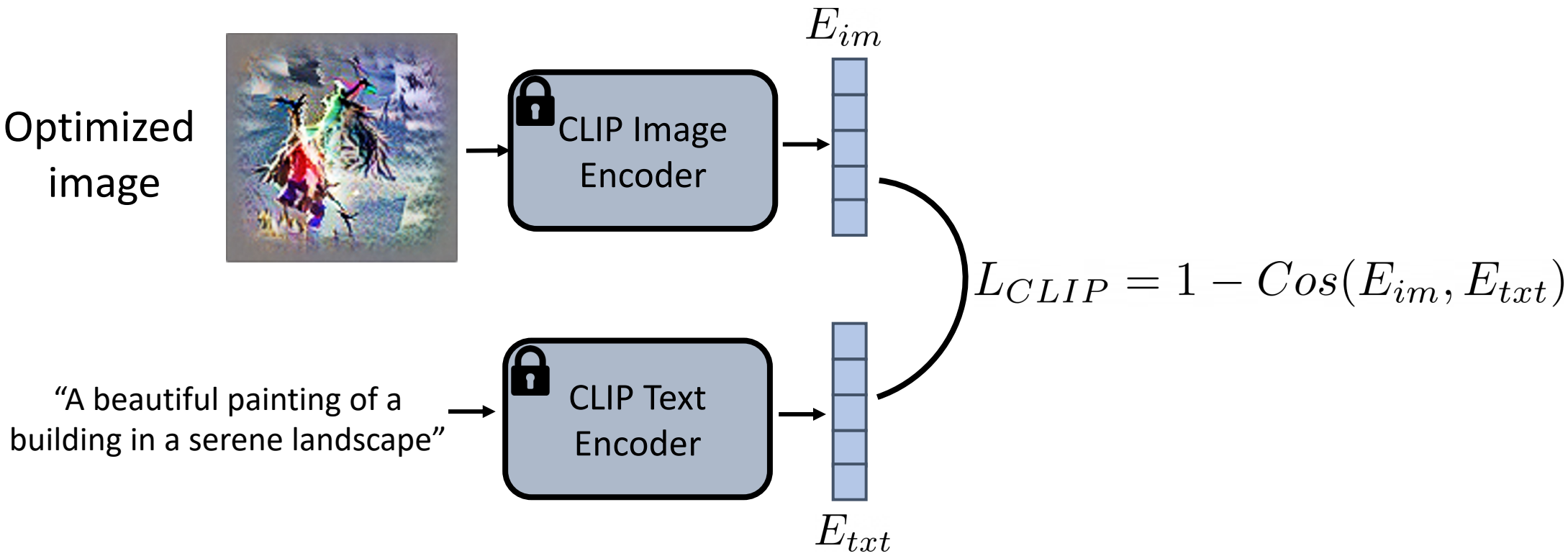
Generation

A beautiful painting of a building in a serene landscape

Editing

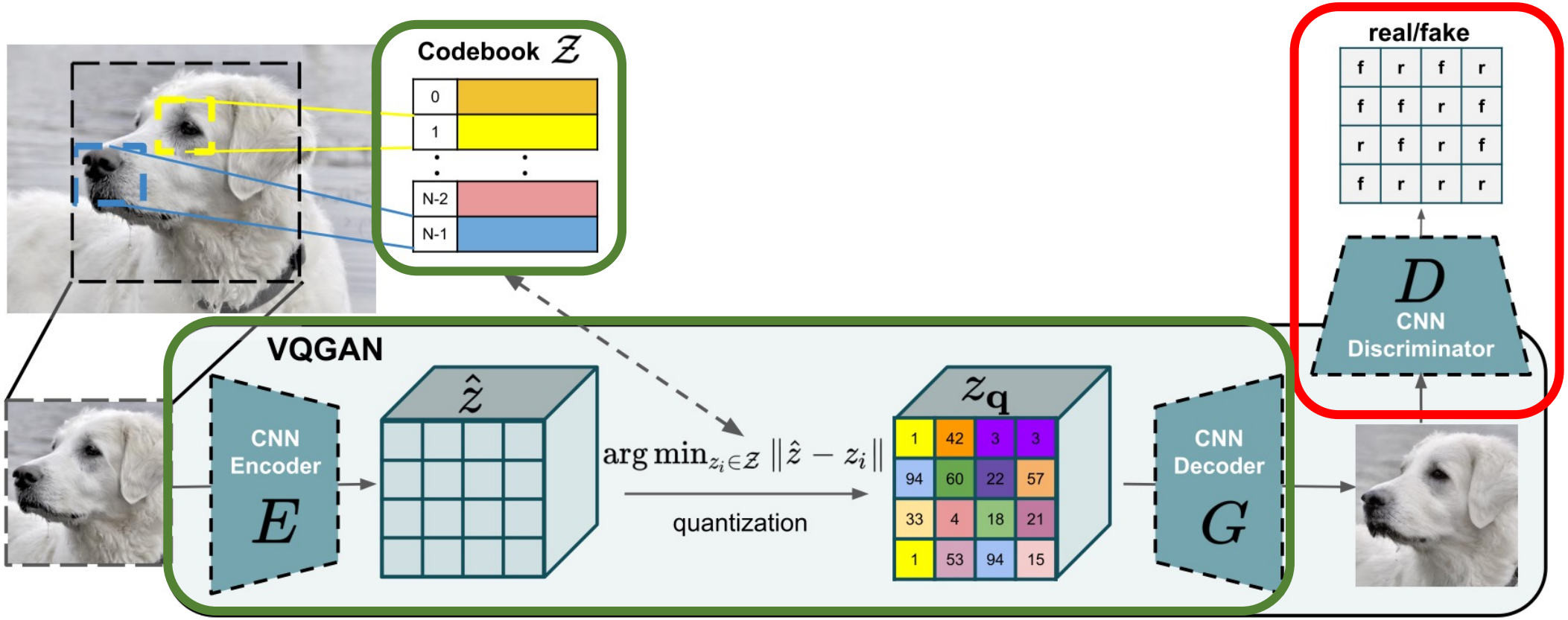
"A cake made of ice"

Naive approach



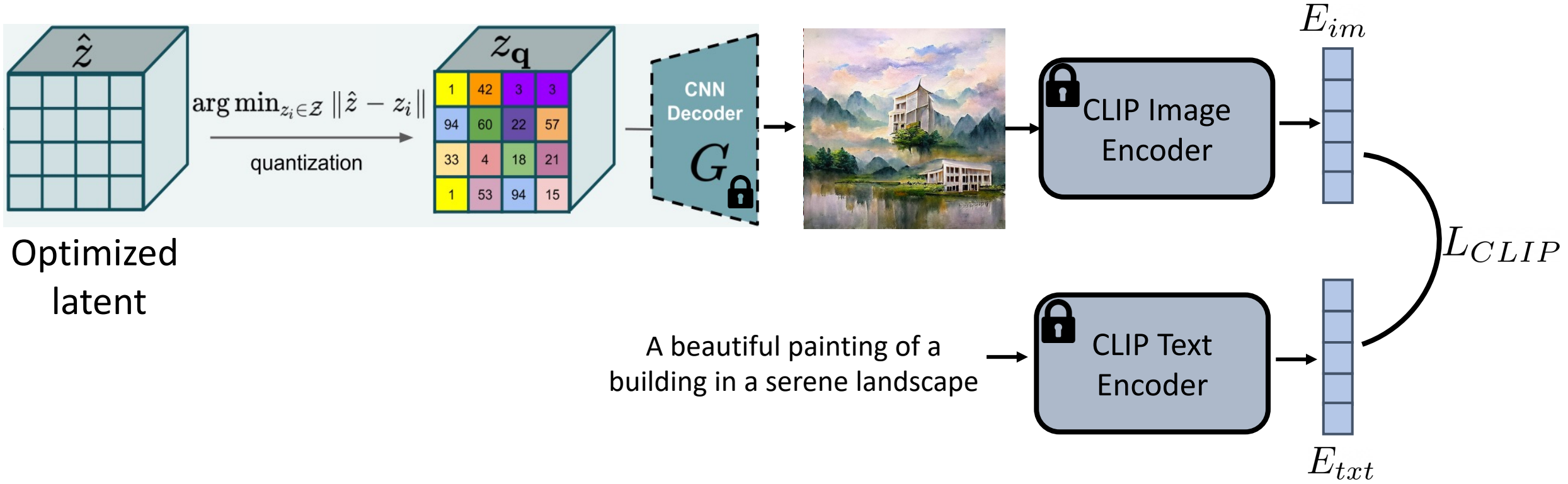
Utilize CLIP to steer the generation towards the desired text prompt

VQ-GAN - reminder



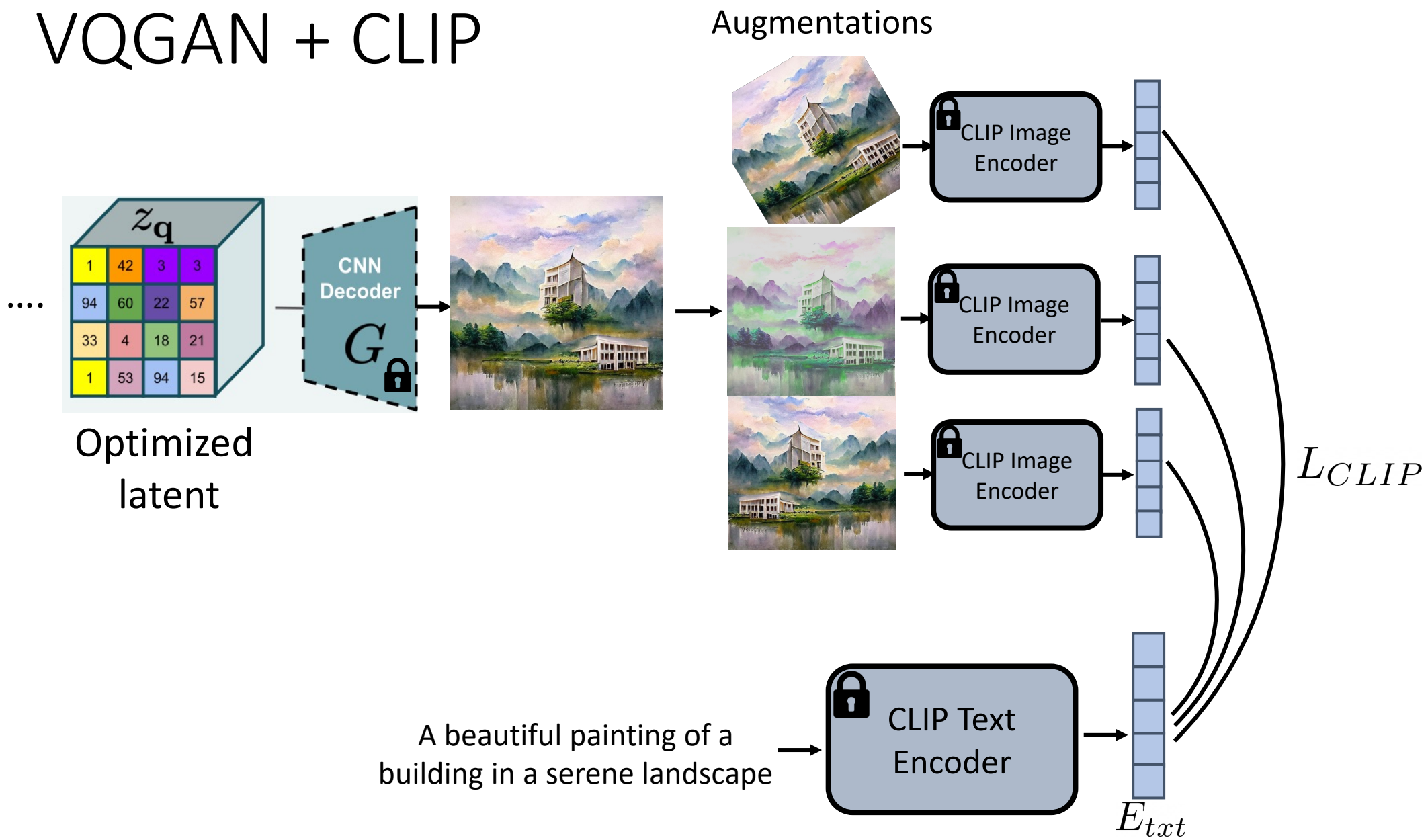
VQ-VAE + GAN

VQGAN + CLIP

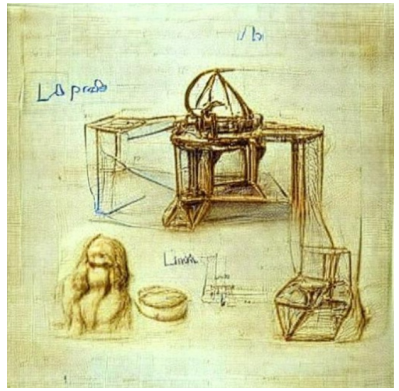


$$L_{CLIP} = 1 - Cos(E_{im}, E_{txt})$$

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}

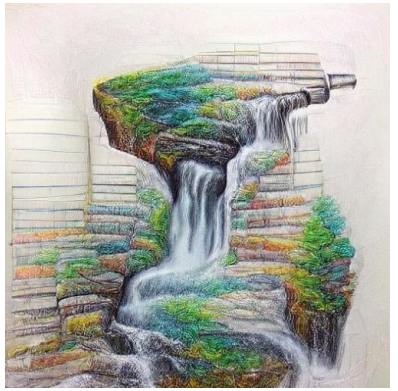
VQGAN + CLIP results



"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

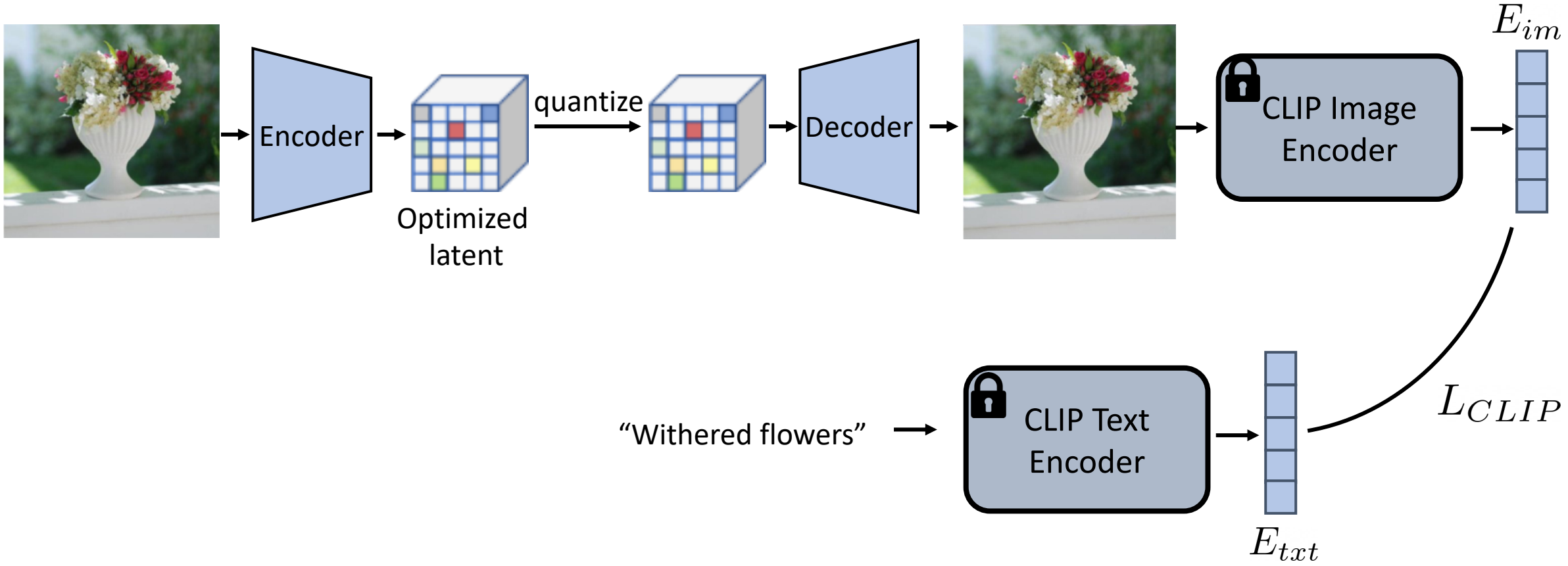
Generation

A beautiful painting of a building in a serene landscape

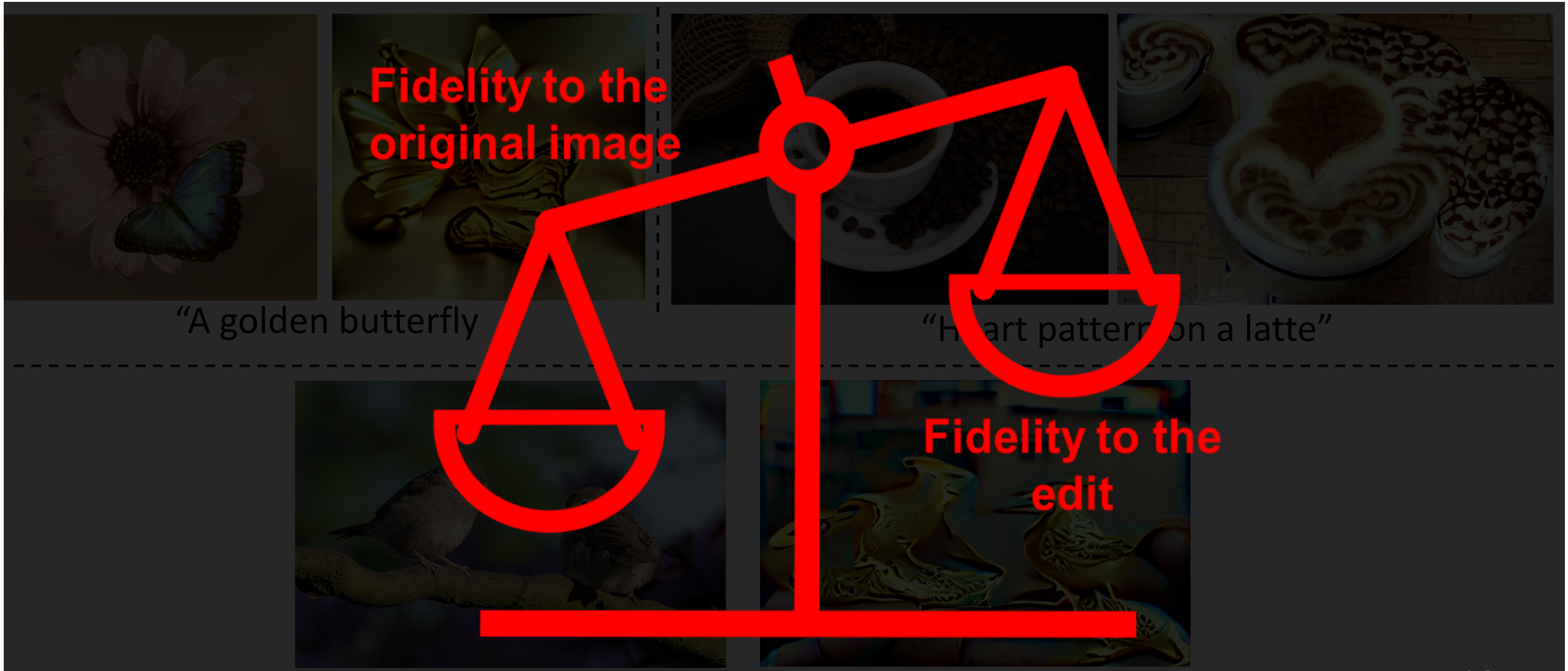
Editing

"A cake made of ice"

VQGAN + CLIP editing

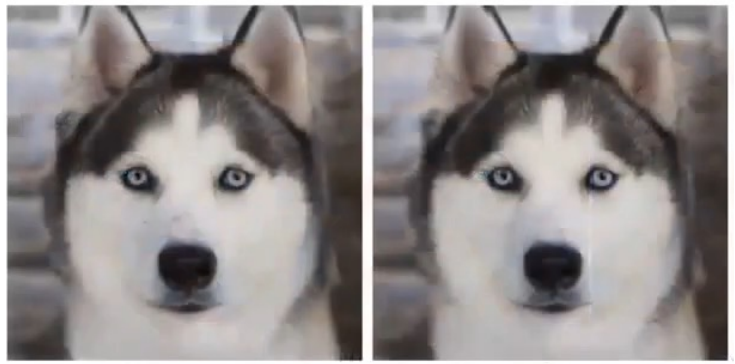


VQGAN + CLIP editing



StyleCLIP – goal

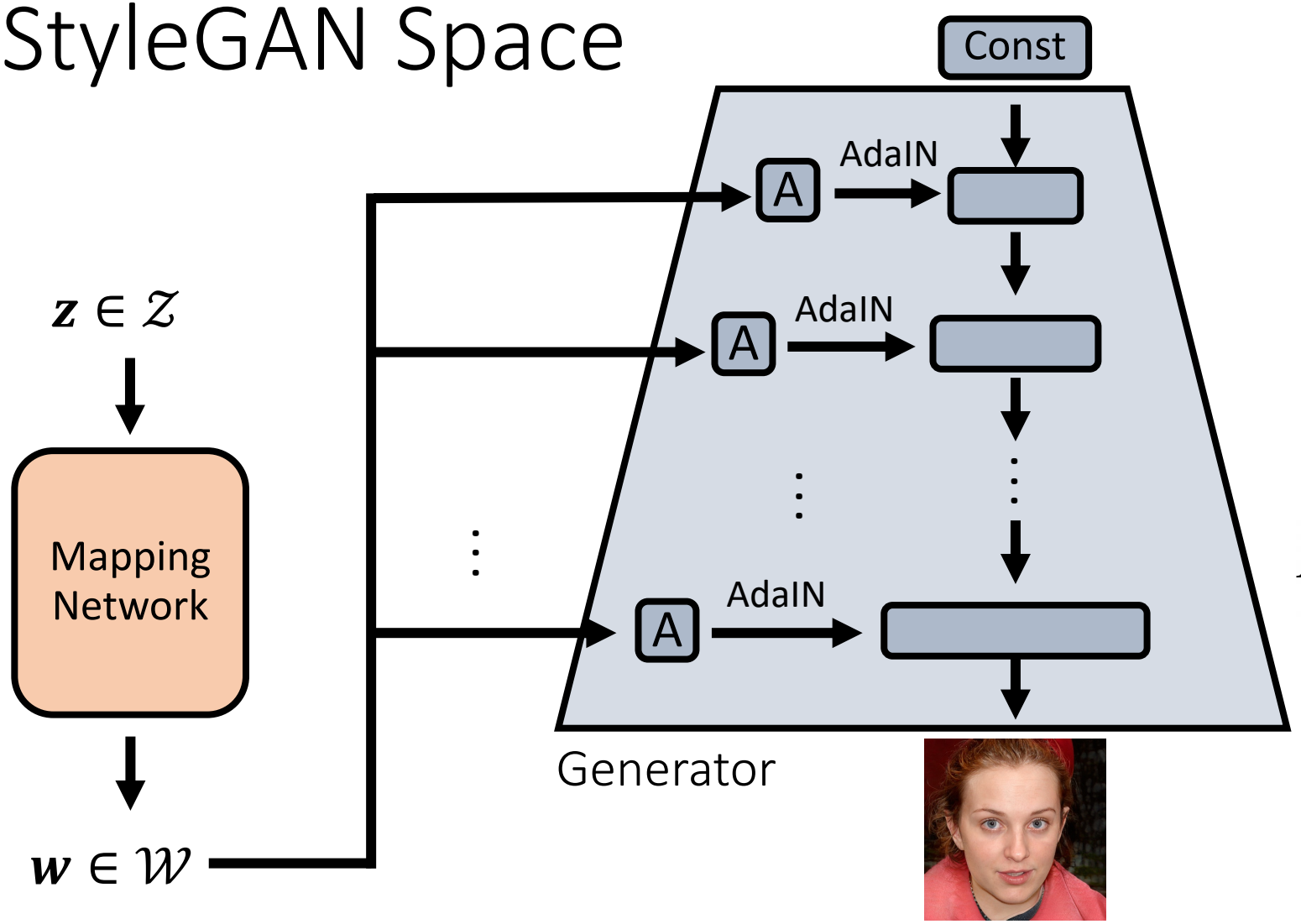
"curly hair" "hi-top fade hair" "fringe hair" "black hair" "makeup"



"happy" "big eyes"

"cloud" "spires"

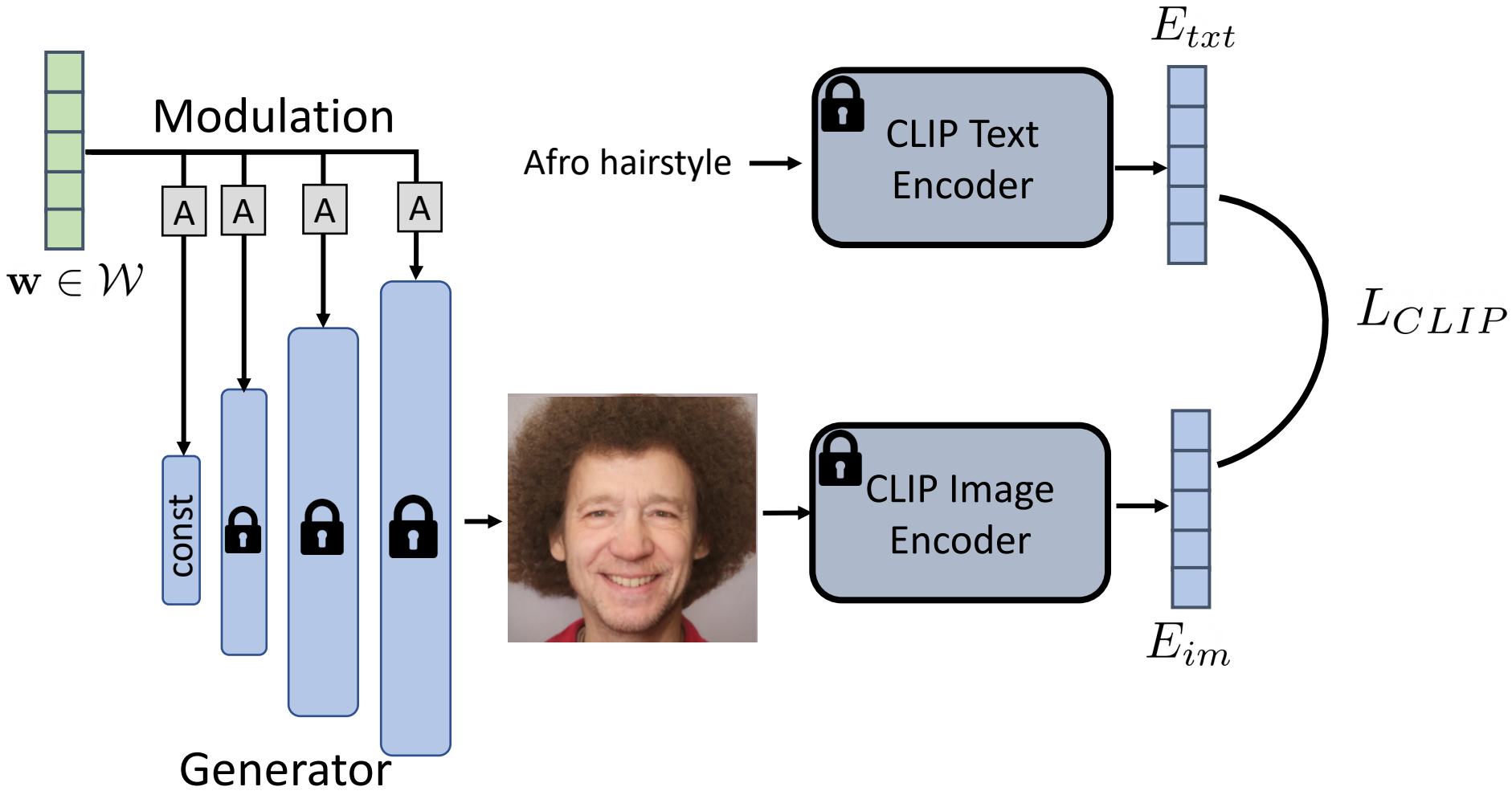
StyleGAN Space



$(\mathbf{y}_{s,i}, \mathbf{y}_{b,i}) = A(\mathbf{w})$ - affine transform

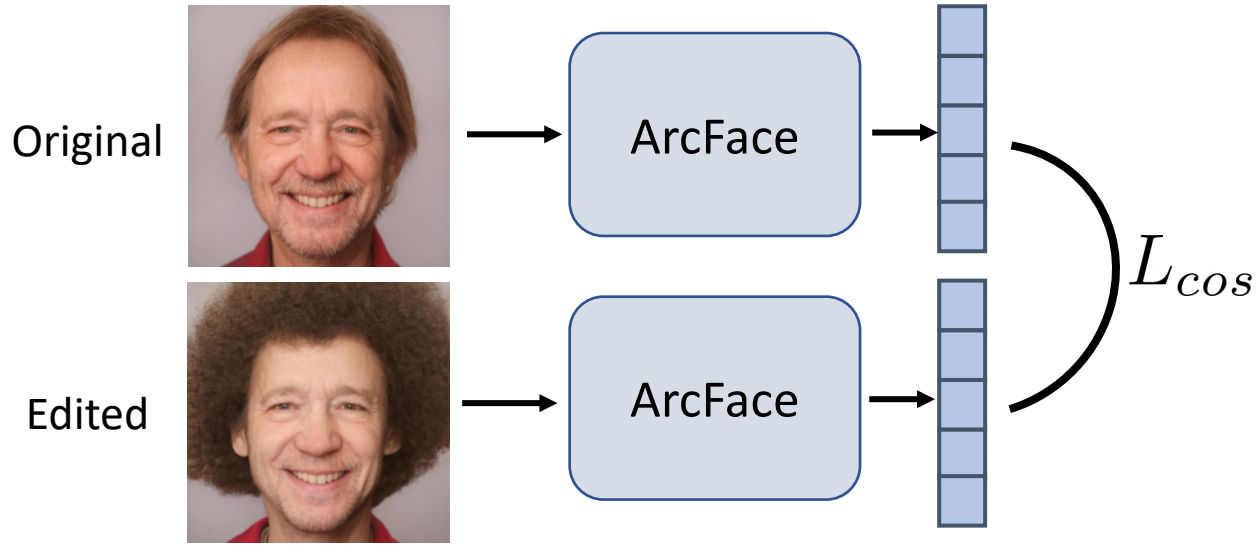
$$\mathrm{AdaIN}\left(\mathbf{x}_i, \mathbf{y}\right) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu\left(\mathbf{x}_i\right)}{\sigma\left(\mathbf{x}_i\right)} + \mathbf{y}_{b,i}$$

StyleCLIP


E_{txt}
Modulation
A
A
A
A
$\mathbf{w} \in \mathcal{W}$
Afro hairstyle
CLIP Text Encoder
L_{CLIP}
const
CLIP Image Encoder
E_{im}
Generator

StyleCLIP - losses

$$\arg\min_{w \in \mathcal{W}} L_{\text{CLIP}}(G(w), T) + \lambda_{\text{L2}} \|w - w_{init}\|_2 + \lambda_{\text{ID}} L_{\text{ID}}(w)$$



StyleCLIP - results

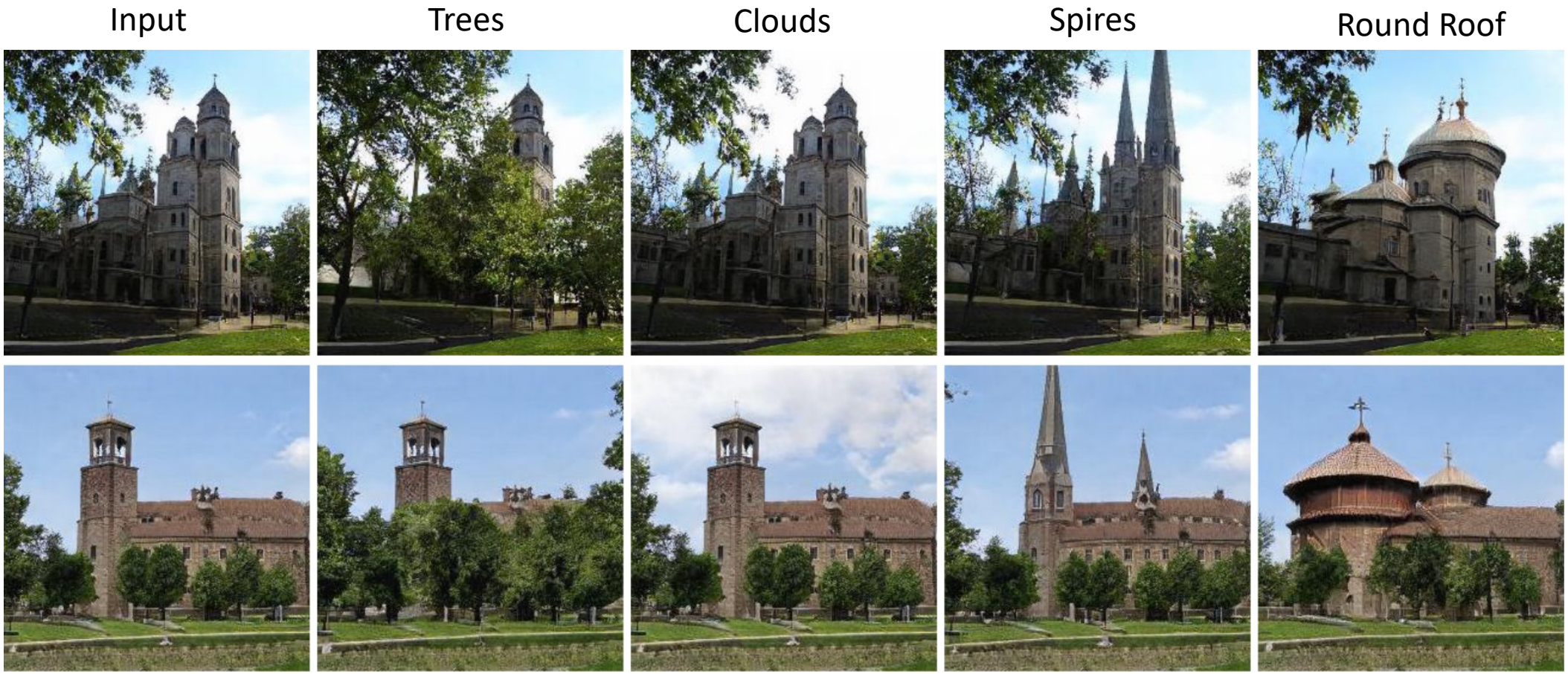
Input | “Beyonce” | “A woman without makeup” | “Elsa from Frozen”

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

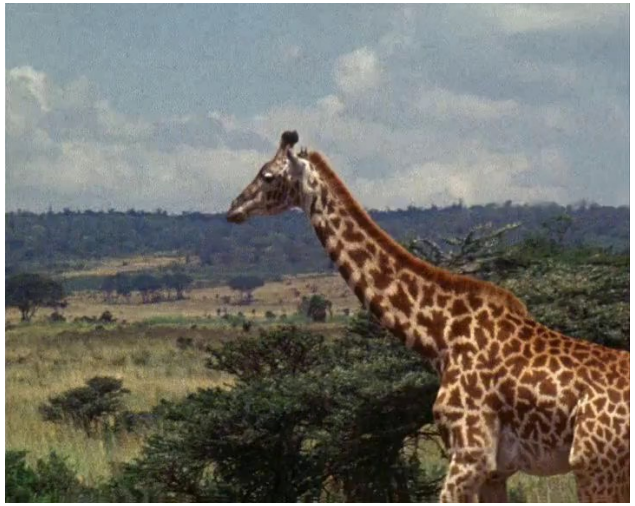
“oreo cake”

“brioche”

“ice”

“spinach moss cake”

Original video

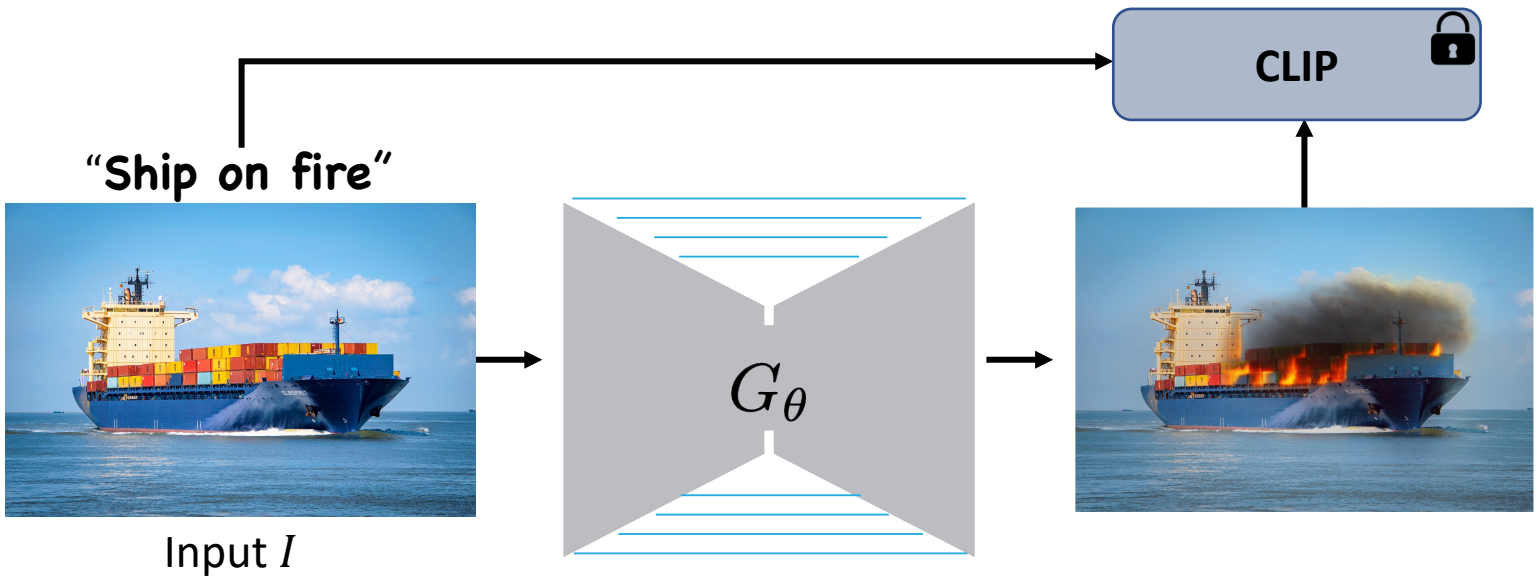


“stained glass giraffe”

“giraffe with neck warmer”

“giraffe with hairy colorful mane”

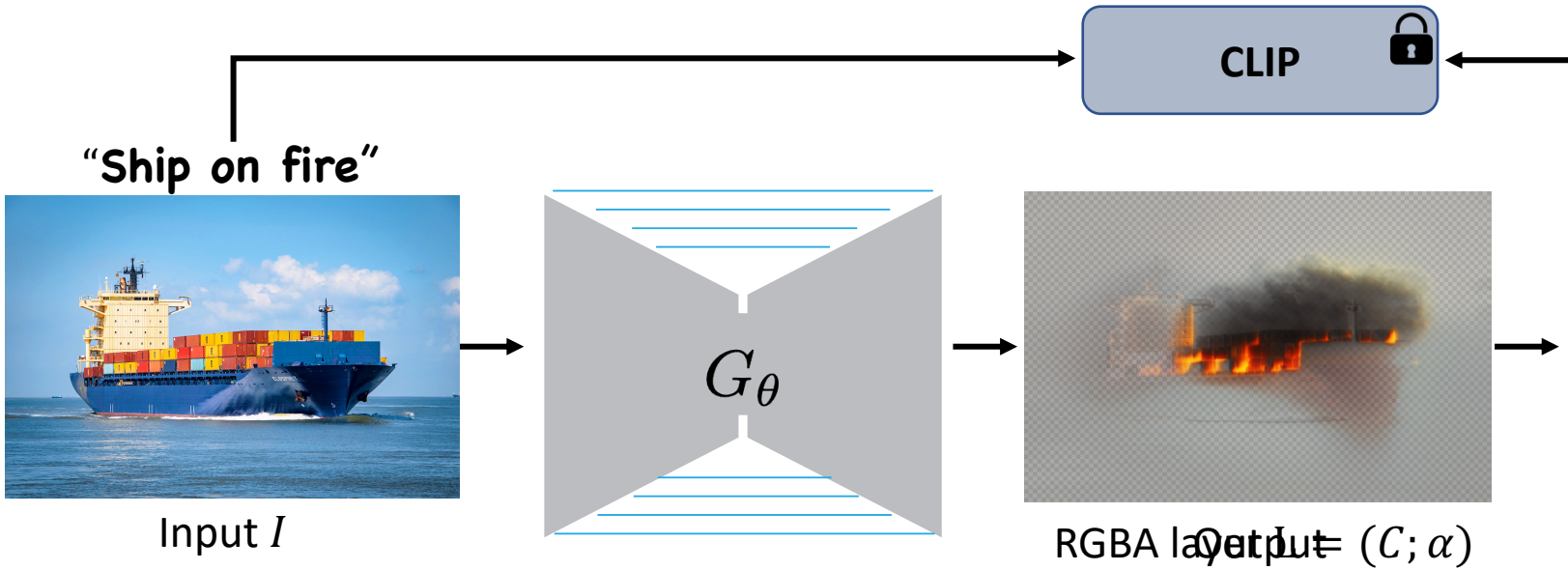
Text2LIVE – method



Augmentations

Training examples

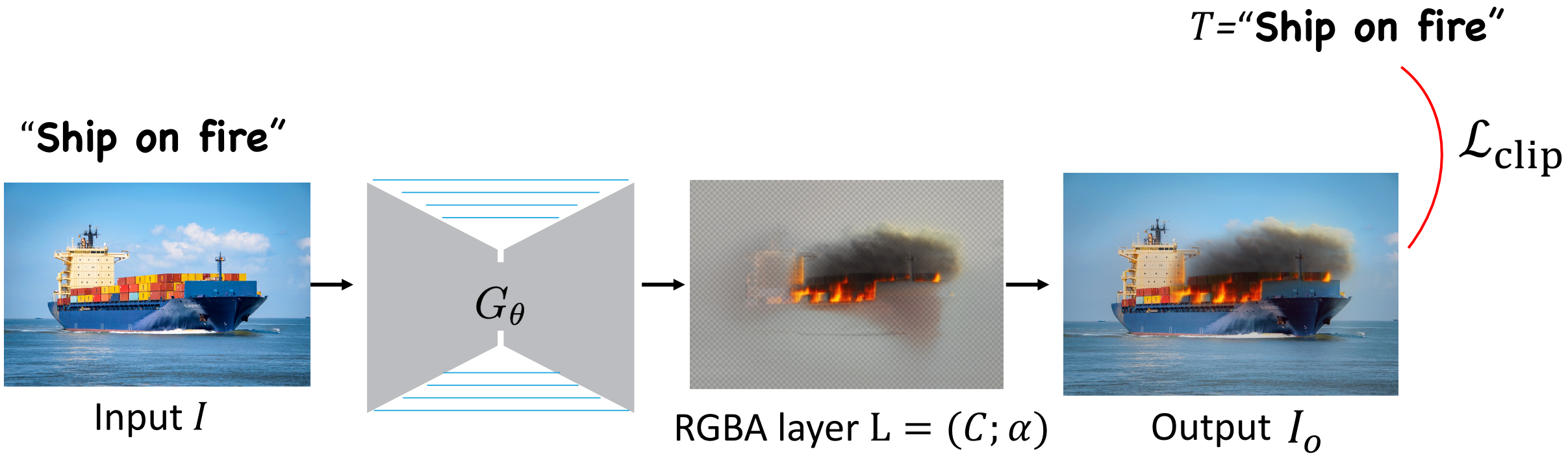
Text2LIVE – method



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

Loss on the composition

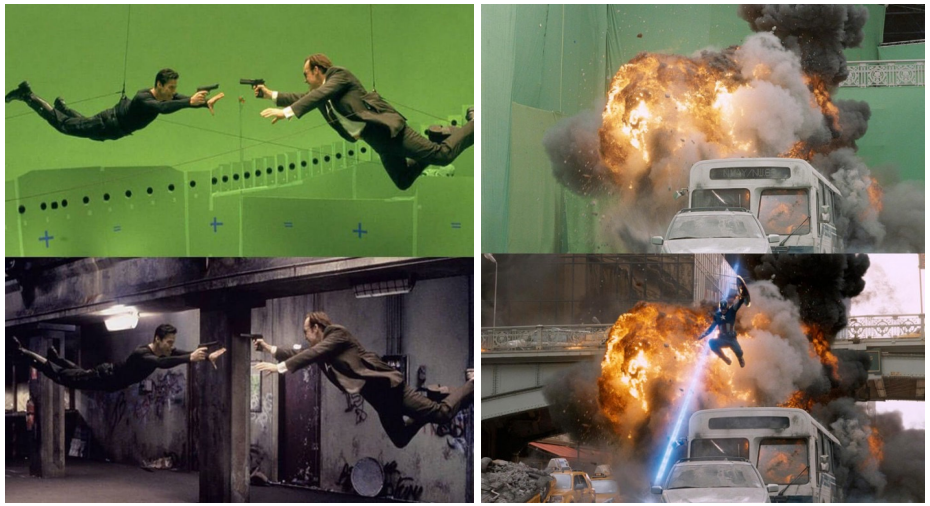
$$\mathcal{L}_{Text2LIVE} = \mathcal{L}_{\mathbf{comp}}(\boldsymbol{I_o}) + \alpha\mathcal{L}_{screen}\,(C, \alpha) + \gamma\mathcal{L}_{\text{sparsity}}(\alpha)$$

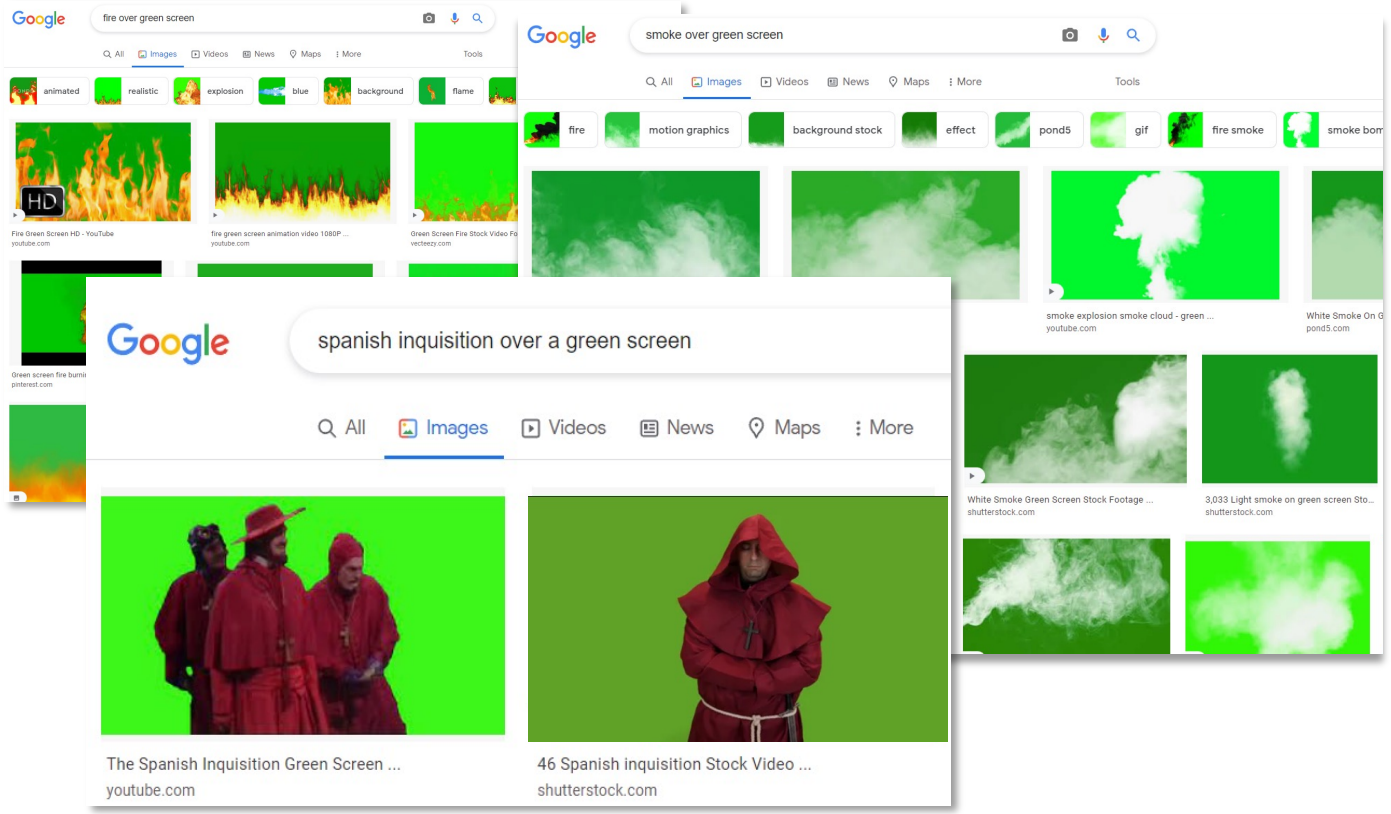


Losses on the edit layer

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

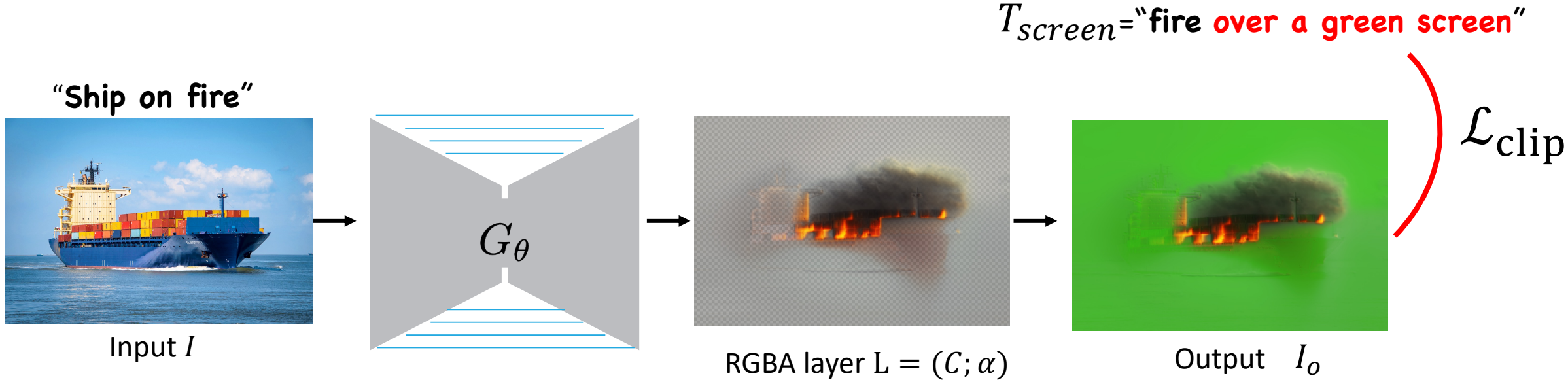
Chroma keying





Text2LIVE – losses

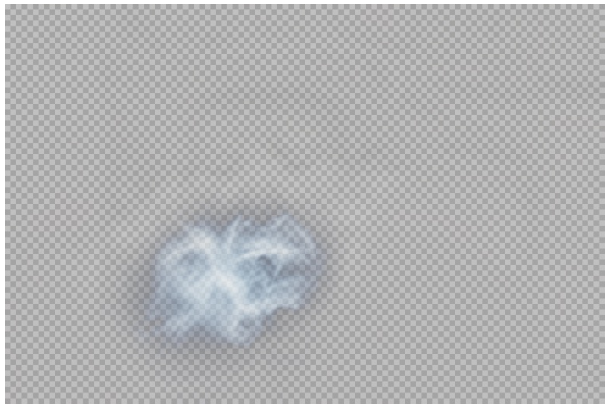
$$\mathcal{L}_{Text2LIVE} = \mathcal{L}_{\text{comp}}(I_o) + \boldsymbol{\alpha}\boldsymbol{\mathcal{L}_{screen}}\,(\boldsymbol{C}, \boldsymbol{\alpha}) + \gamma\mathcal{L}_{\text{sparsity}}(\alpha)$$



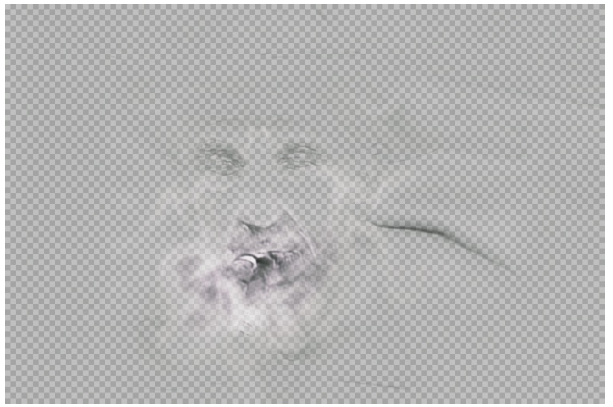
Losses on the edit layer

"*A man smoking a cigar*"

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



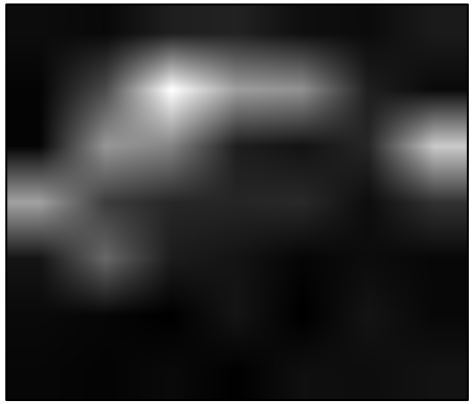
w/o $\mathcal{L}_{screen}$



Losses on the edit layer

Input Image

"woman wearing a red hat"



Relevancy map*

"**hat**"

Text2LIVE output matte

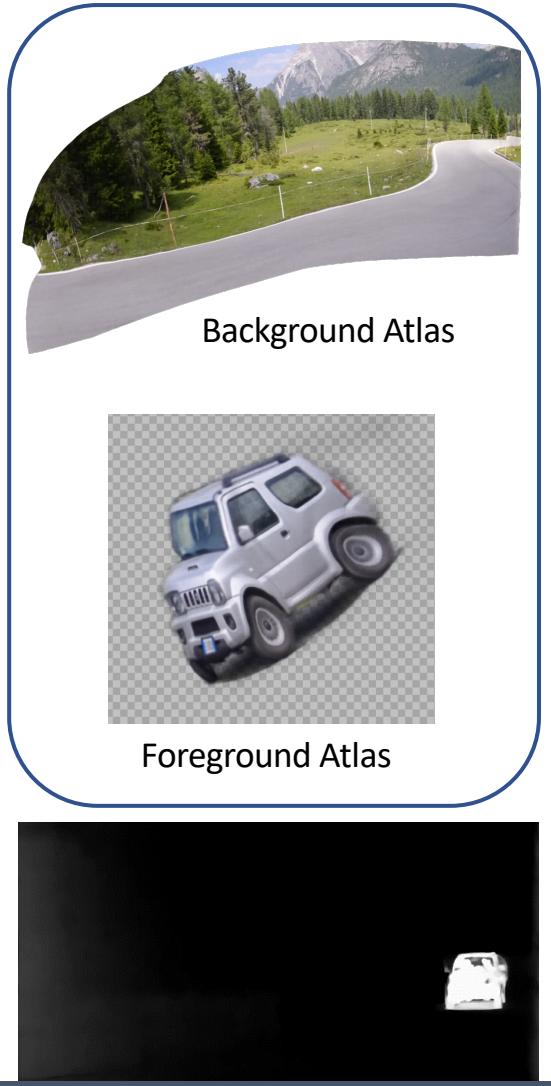
MSE

Text2LIVE result

Text2LIVE – results

NLA - reminder

Input video



Render Video

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

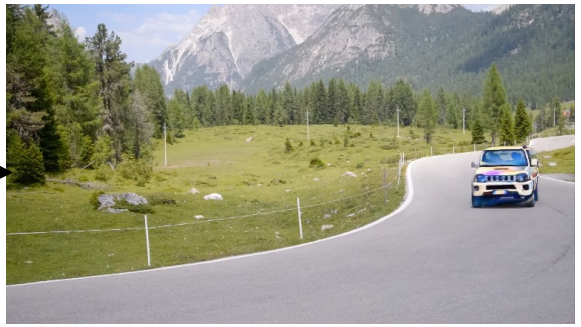
Video Reconstruction

NLA - reminder

Input video

Render Video

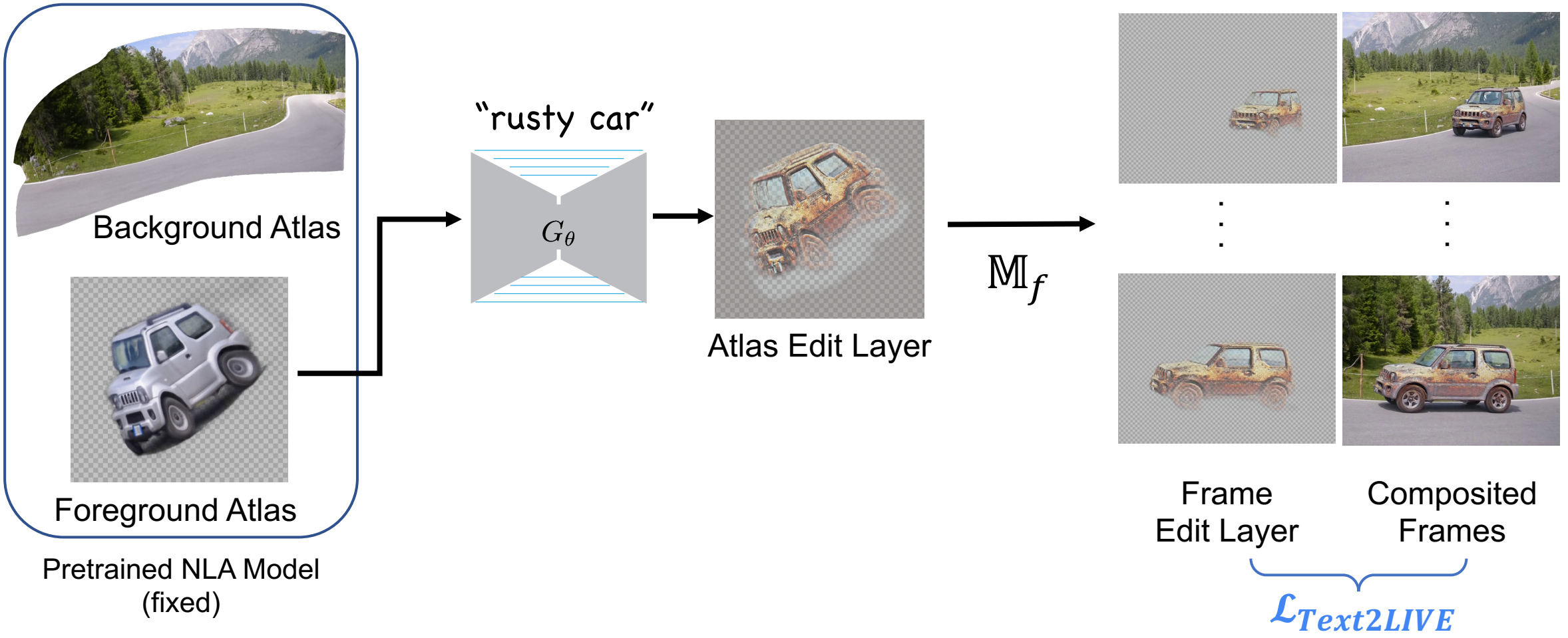
$$c^{\boldsymbol{p}} = (1 - \alpha^{\boldsymbol{p}})c_b^{\boldsymbol{p}} + \alpha^{\boldsymbol{p}} c_f^{\boldsymbol{p}}$$



Video Reconstruction

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

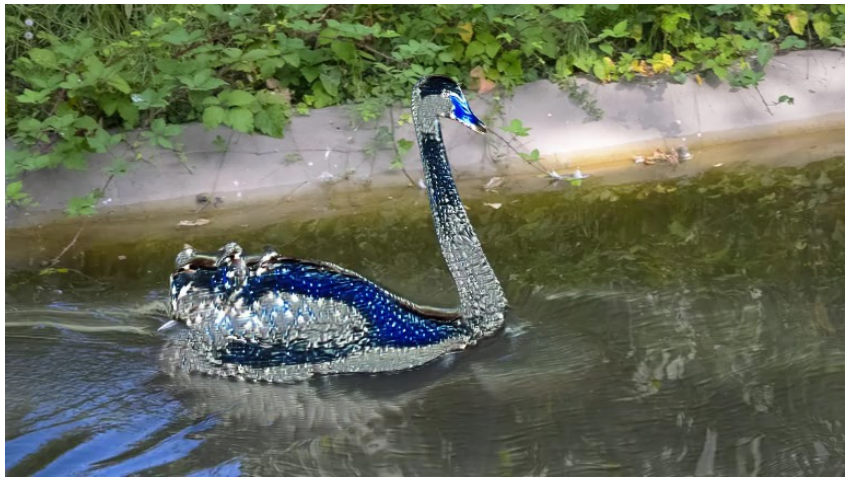
Text2LIVE video editing



Text2LIVE – results

Input Video

"swarovski blue crystal swan"

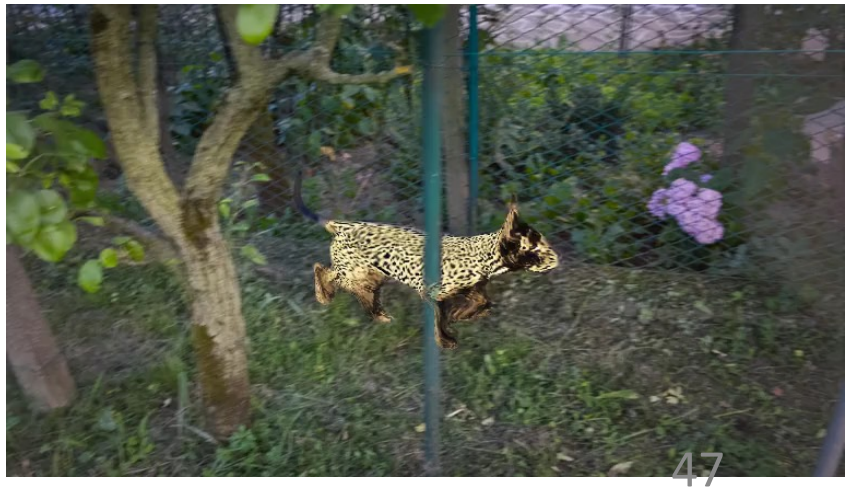


Original Video

Input Video

"dalmatian dog"

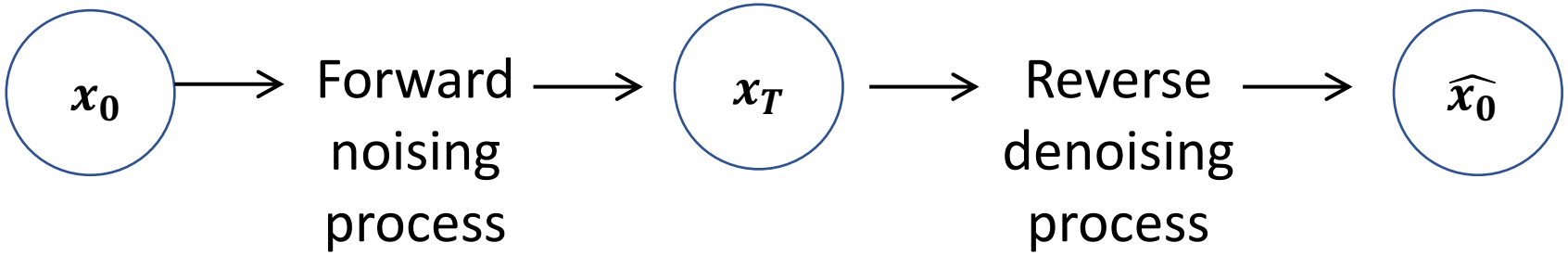
"dog with leopard texture"



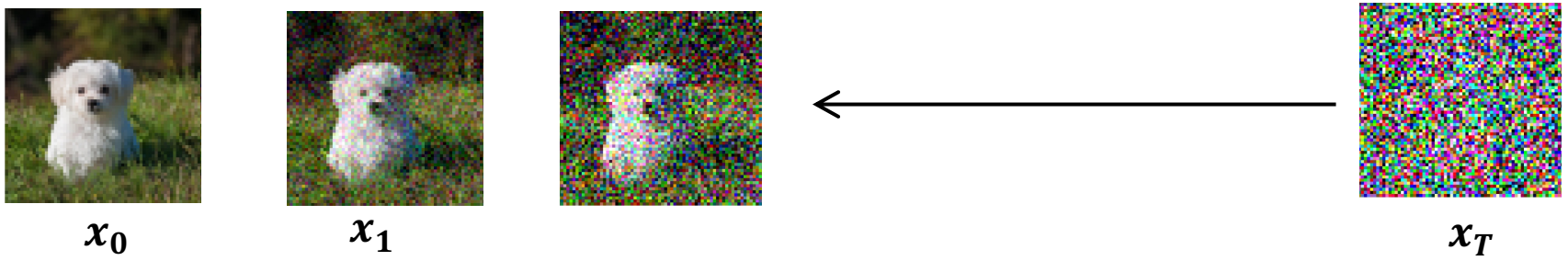
Topics

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



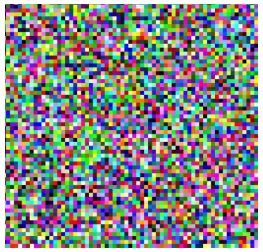
Reverse denoising process



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

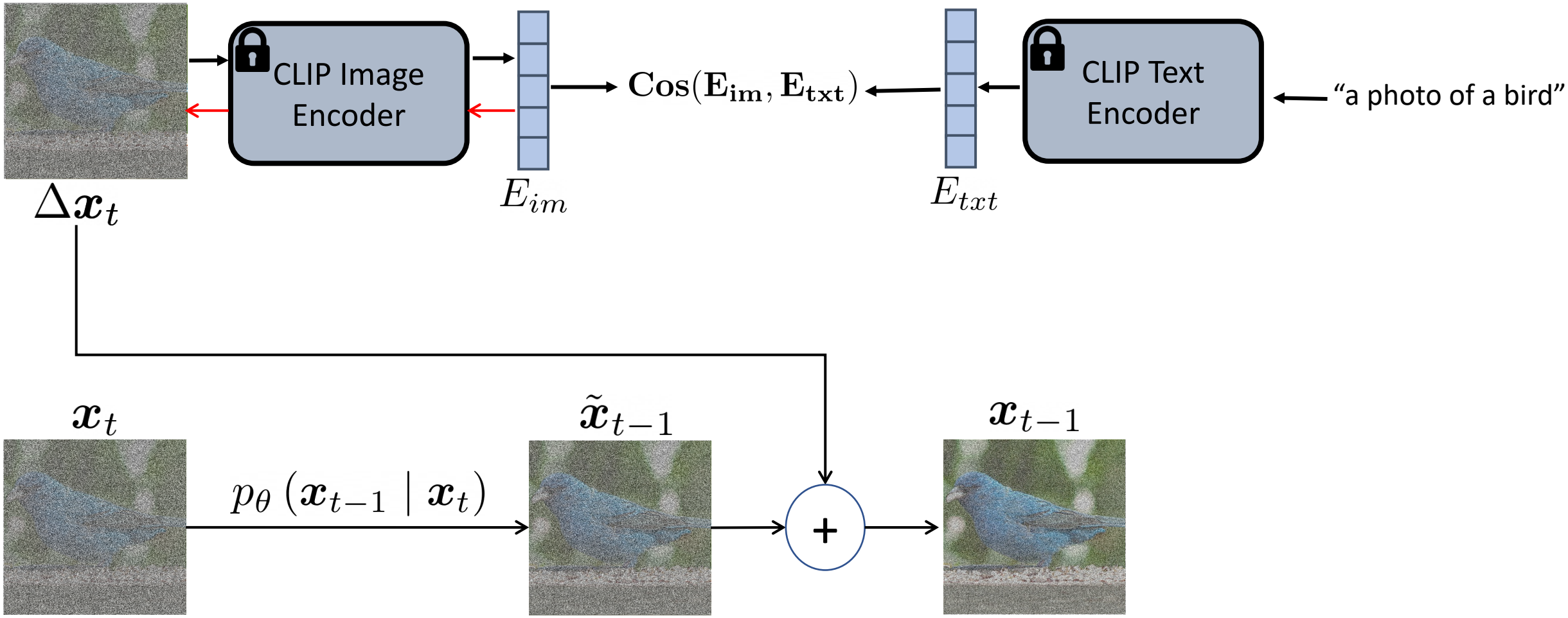
1. Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. For $t = T, \ldots, 1$:
   - Predict mean of the reverse distribution $\mu_\theta(x_t, t)$
   - Sample from reverse distribution:

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t^2 \mathbf{z}$$

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

3. Return x_0

Conditional image generation **with CLIP**



Conditional sampling **with CLIP model**

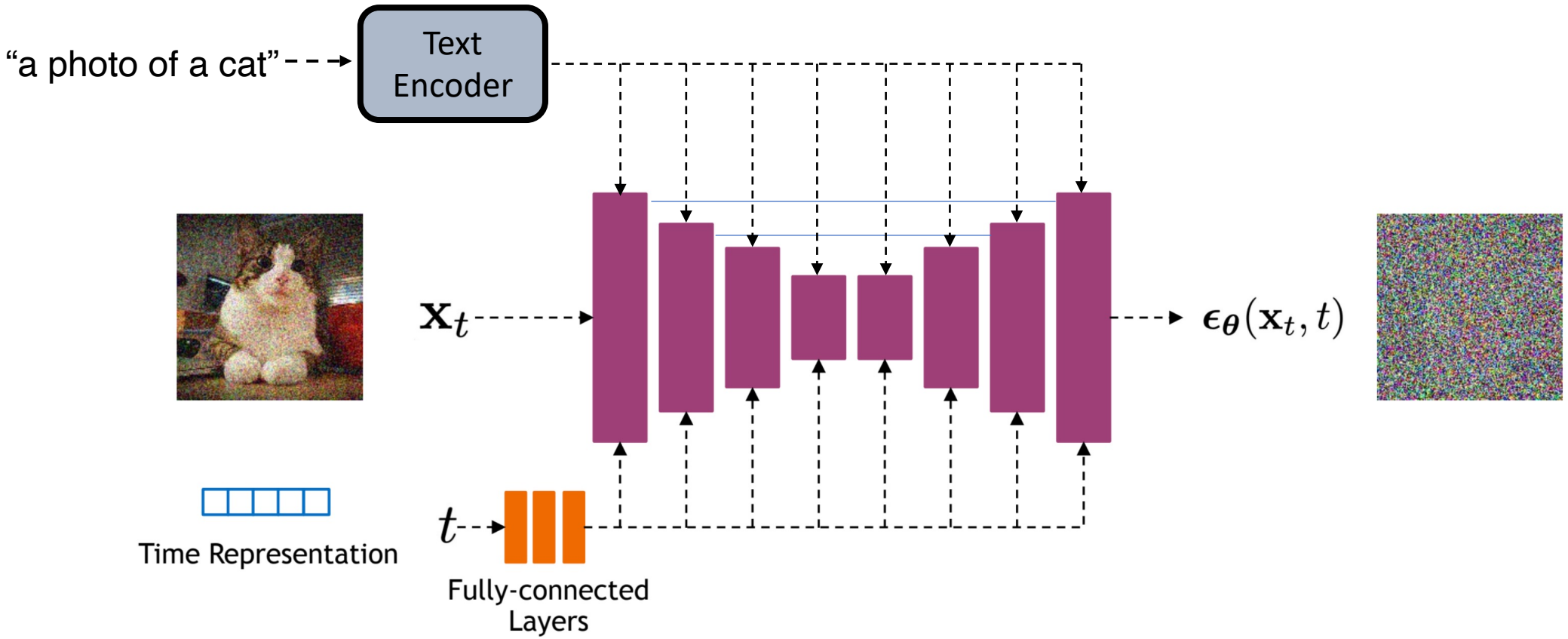
1. Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, text prompt c, text encoder g, image encoder f
2. For $t = T, \ldots, 1$:
   - Predict mean of the reverse distribution $\mu_\theta(x_t, t)$
   - Sample from reverse distribution:

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t^2 \mathbf{z} + s \nabla_{x_t} \left(f\left(x_t\right) \cdot g(c) \right)$$

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

3. Return x_0

Text conditioning implementation



Cross-attention

Noisy image x_t

"A photo of a cat"

Prompt P

Text tokens $\psi(P)$

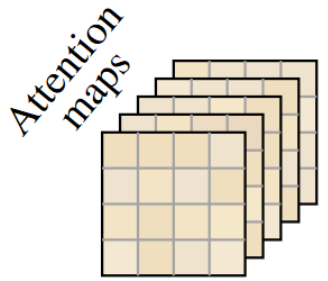
$l_K(\psi(P))$

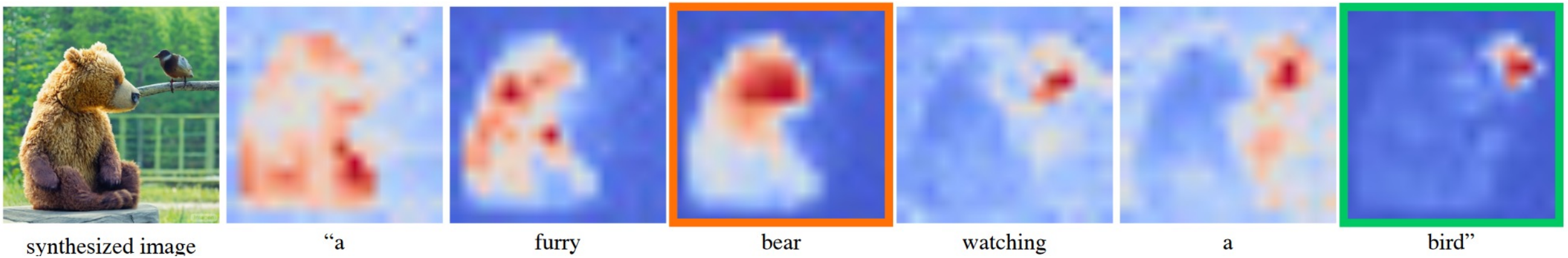
$l_V(\psi(P))$

$\phi(x_t)$

$\hat{\phi}(x_t)$

Text conditioning in Diffusion Models

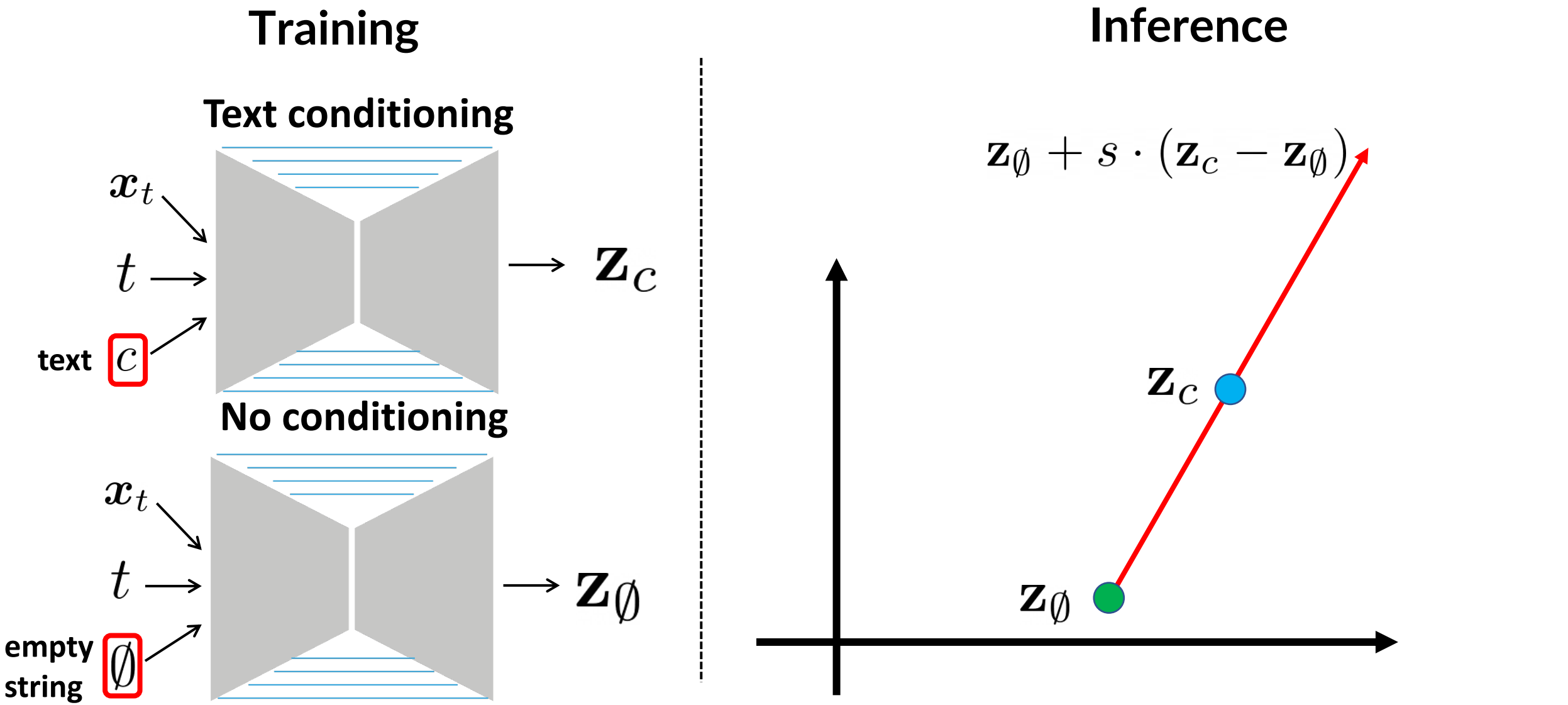




Average attention maps across all timestamps

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

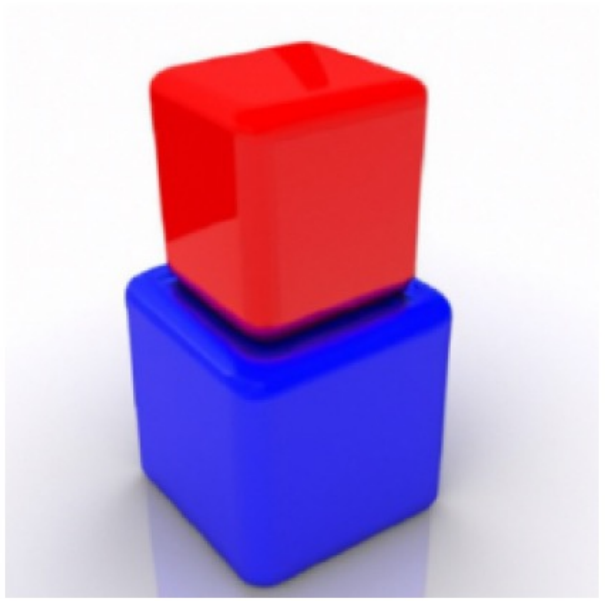
Noised CLIP guidance

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

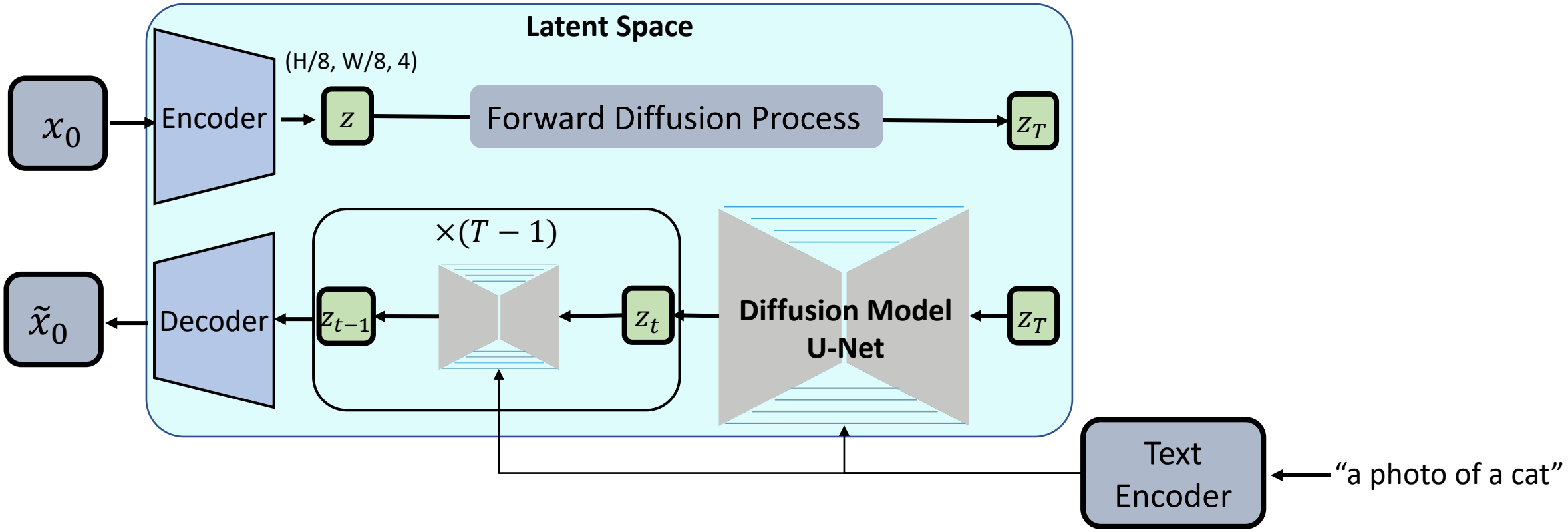
512 x 512

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

High-Resolution Image Synthesis with Latent Diffusion Models, Rombach and Blattmann et al., CVPR 2022

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?