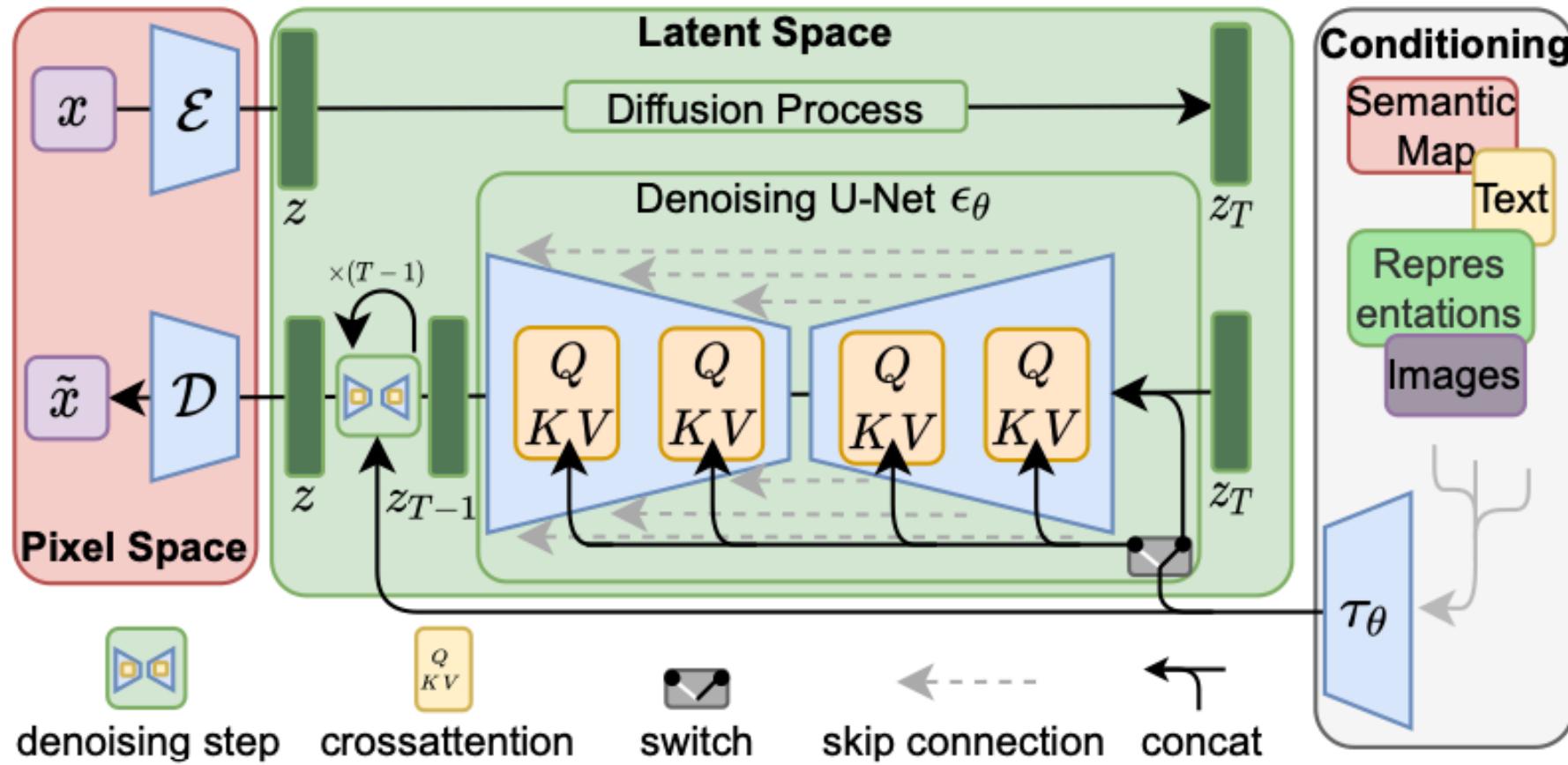
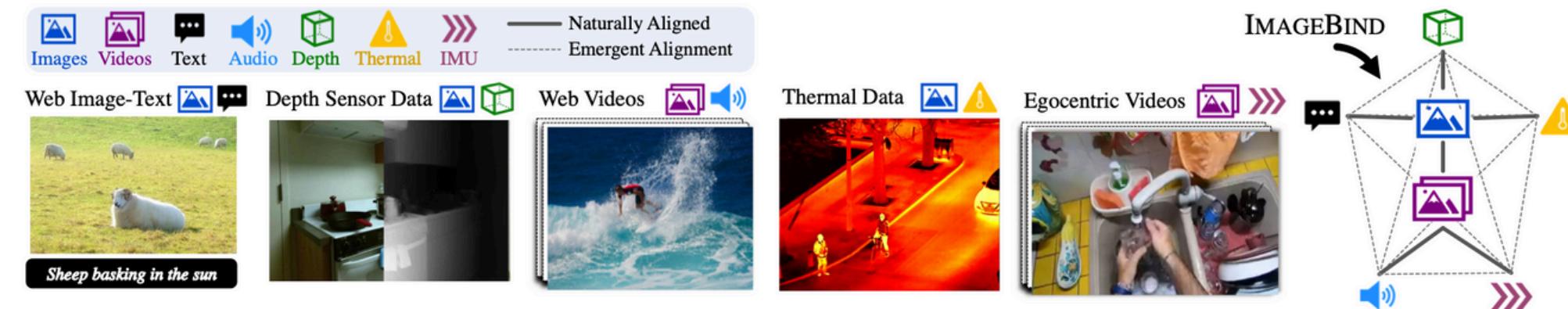


DiffusionCraft



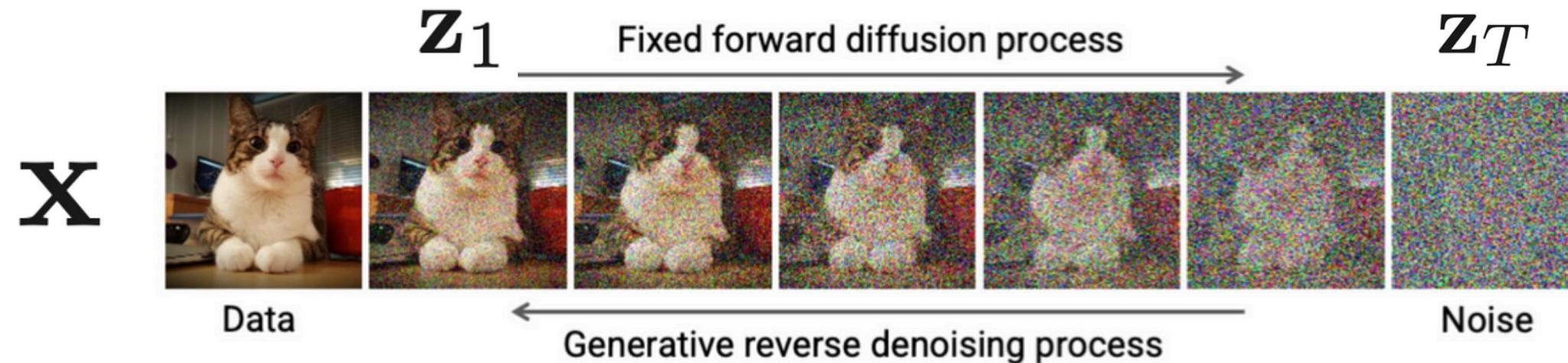
Stable Diffusion

ImageBind



Diffusion Models: an overview

$$\begin{aligned}
 \mathbf{z}_1 &= \sqrt{1 - \beta_1} \cdot \mathbf{x} + \sqrt{\beta_1} \cdot \boldsymbol{\epsilon}_1 \\
 \mathbf{z}_2 &= \sqrt{1 - \beta_2} \cdot \mathbf{z}_1 + \sqrt{\beta_2} \cdot \boldsymbol{\epsilon}_2 \\
 \vdots
 \end{aligned}
 \xrightarrow{\text{In one step}}
 \begin{aligned}
 \mathbf{z}_t &= \sqrt{\alpha_t} \cdot \mathbf{x} + \sqrt{1 - \alpha_t} \cdot \boldsymbol{\epsilon} \\
 \alpha_t &= \prod_{s=1}^t 1 - \beta_s
 \end{aligned}$$



To be learned

Loss function, obtained minimising -ELBO

$$Pr(\mathbf{z}_{t-1}|\mathbf{z}_t, \boldsymbol{\phi}_t) = \text{Norm}_{\mathbf{z}_{t-1}}[\mathbf{f}_t[\mathbf{z}_t, \boldsymbol{\phi}_t], \sigma_t^2 \mathbf{I}]$$

$$Pr(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\phi}_1) = \text{Norm}_{\mathbf{x}}[\mathbf{f}_1[\mathbf{z}_1, \boldsymbol{\phi}_1], \sigma_1^2 \mathbf{I}]$$

$$\begin{aligned}
 L[\boldsymbol{\phi}_{1\dots T}] &= \sum_{i=1}^I \underbrace{\left(-\log \left[\text{Norm}_{\mathbf{x}_i} [\mathbf{f}_1[\mathbf{z}_{i1}, \boldsymbol{\phi}_1], \sigma_1^2 \mathbf{I}] \right] \right)}_{\text{reconstruction term}} \\
 &\quad + \sum_{t=2}^T \frac{1}{2\sigma_t^2} \underbrace{\left\| \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \sqrt{1 - \beta_t} \mathbf{z}_{it} + \frac{\sqrt{\alpha_{t-1}} \beta_t}{1 - \alpha_t} \mathbf{x}_i - \mathbf{f}_t[\mathbf{z}_{it}, \boldsymbol{\phi}_t] \right\|^2}_{\text{target, mean of } q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})} \\
 &\quad \underbrace{-}_{\text{predicted } \mathbf{z}_{t-1}}
 \end{aligned}$$

Diffusion Models: a few adjustments

Reparametrize $\hat{\mathbf{z}}_{t-1} = \mathbf{f}_t[\mathbf{z}_t, \phi_t]$ \longrightarrow $\mathbf{f}_t[\mathbf{z}_t, \phi_t] = \frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}}\mathbf{g}_t[\mathbf{z}_t, \phi_t]$

1/50 to 1/100 number of steps

Make it faster with DDIM:
Skip some denoising steps

$$\hat{\epsilon} = \mathbf{g}_t[\mathbf{z}_t, \phi_t]$$

$$\mathbf{z}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{z}_t - \sqrt{1-\alpha_t} g_t[\mathbf{z}_t, \phi_t]}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} g_t[\mathbf{z}_t, \phi_t] + \sigma_t \epsilon$$

Algorithm 18.1: Diffusion model training

Input: Training data \mathbf{x}
Output: Model parameters ϕ_t

```

repeat
  for  $i \in \mathcal{B}$  do                                // For every training example index in batch
     $t \sim \text{Uniform}[1, \dots T]$            // Sample random timestep
     $\epsilon \sim \text{Norm}[\mathbf{0}, \mathbf{I}]$           // Sample noise
     $\ell_i = \|\mathbf{g}_t[\sqrt{\alpha_t} \mathbf{x}_i + \sqrt{1-\alpha_t} \epsilon, \phi_t] - \epsilon\|^2$  // Compute individual loss
    Accumulate losses for batch and take gradient step
  until converged

```

Algorithm 18.2: Sampling

Input: Model, $\mathbf{g}_t[\bullet, \phi_t]$
Output: Sample, \mathbf{x}

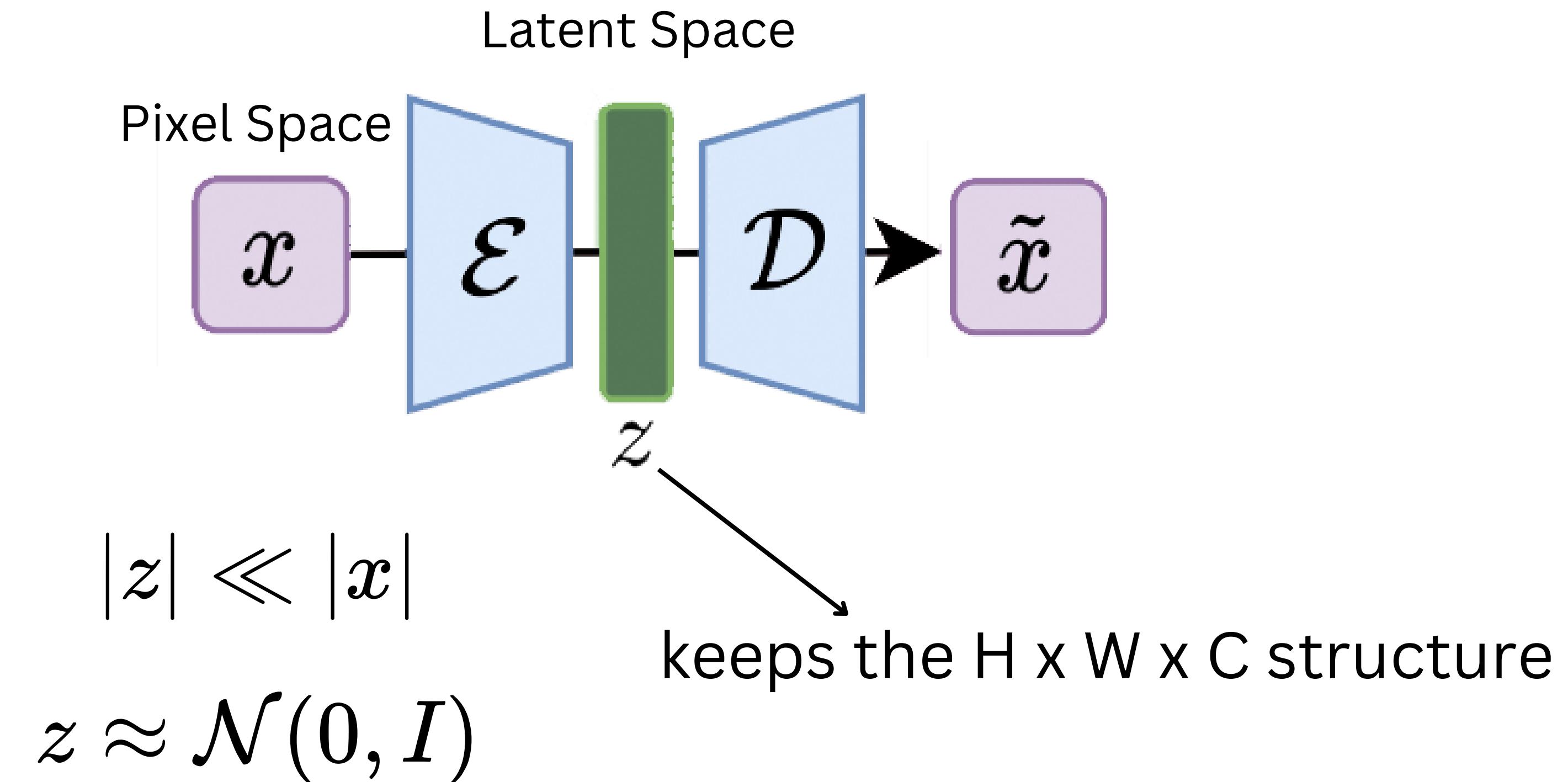
```

 $\mathbf{z}_T \sim \text{Norm}_{\mathbf{z}}[\mathbf{0}, \mathbf{I}]$                                 // Sample last latent variable
for  $t = T \dots 2$  do
   $\hat{\mathbf{z}}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}} \mathbf{g}_t[\mathbf{z}_t, \phi_t]$  // Predict previous latent variable
   $\epsilon \sim \text{Norm}_{\epsilon}[\mathbf{0}, \mathbf{I}]$                                      // Draw new noise vector
   $\mathbf{z}_{t-1} = \hat{\mathbf{z}}_{t-1} + \sigma_t \epsilon$                                     // Add noise to previous latent variable
   $\mathbf{x} = \frac{1}{\sqrt{1-\beta_1}} \mathbf{z}_1 - \frac{\beta_1}{\sqrt{1-\alpha_1}\sqrt{1-\beta_1}} \mathbf{g}_1[\mathbf{z}_1, \phi_1]$  // Generate sample from  $\mathbf{z}_1$  without noise

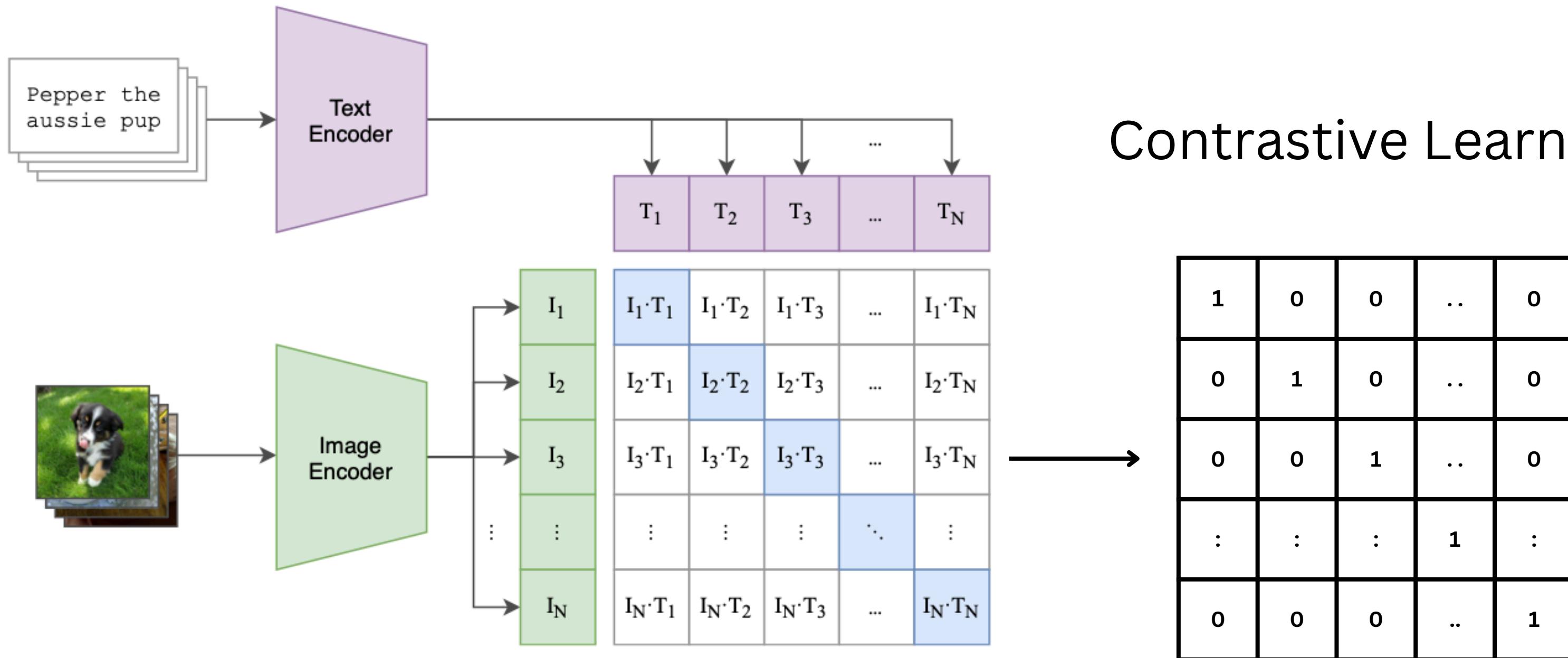
```

Latent Diffusion Models

Intermediate latent representation by a VAE



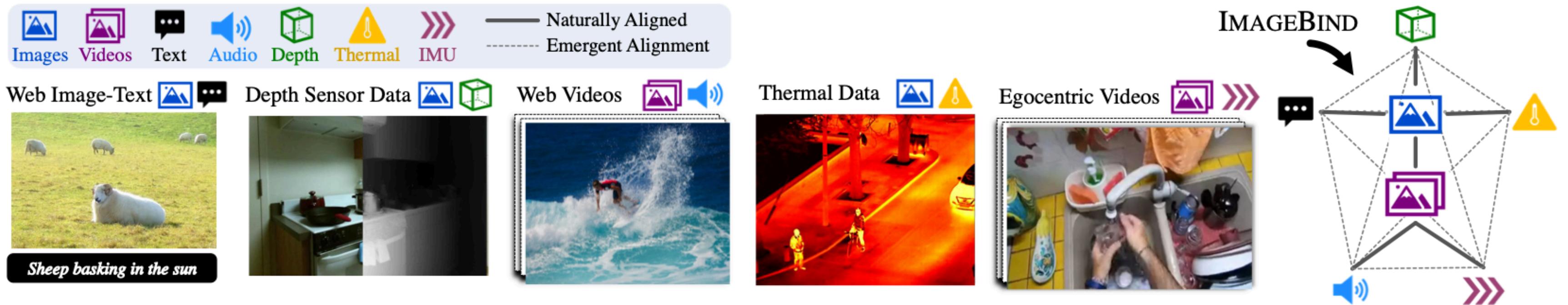
CLIP: aligned text - image embeddings



Trained with Information
Noise-Contrastive Estimation loss:

$$-\log \frac{\exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_k)/\tau)} - \log \frac{\exp(f(\mathbf{x}_i) \cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_k) \cdot g(\mathbf{c}_j)/\tau)}$$

ImageBind: emerging aligned embeddings of different modalities



Designed to work with CLIP text embeddings

Trained with InfoCNE loss:

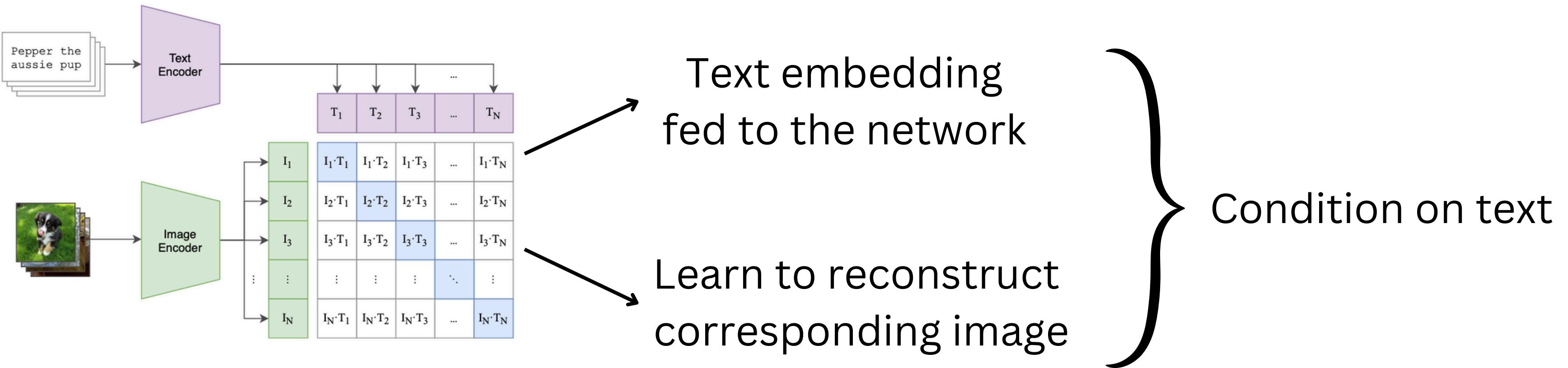
$$L_{\mathcal{I}, \mathcal{M}} + L_{\mathcal{M}, \mathcal{I}}$$

$$L_{\mathcal{I}, \mathcal{M}} = -\log \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_i / \tau)}{\exp(\mathbf{q}_i^\top \mathbf{k}_i / \tau) + \sum_{j \neq i} \exp(\mathbf{q}_i^\top \mathbf{k}_j / \tau)}$$

↓
Image encoder ↓
Modality 2 encoder

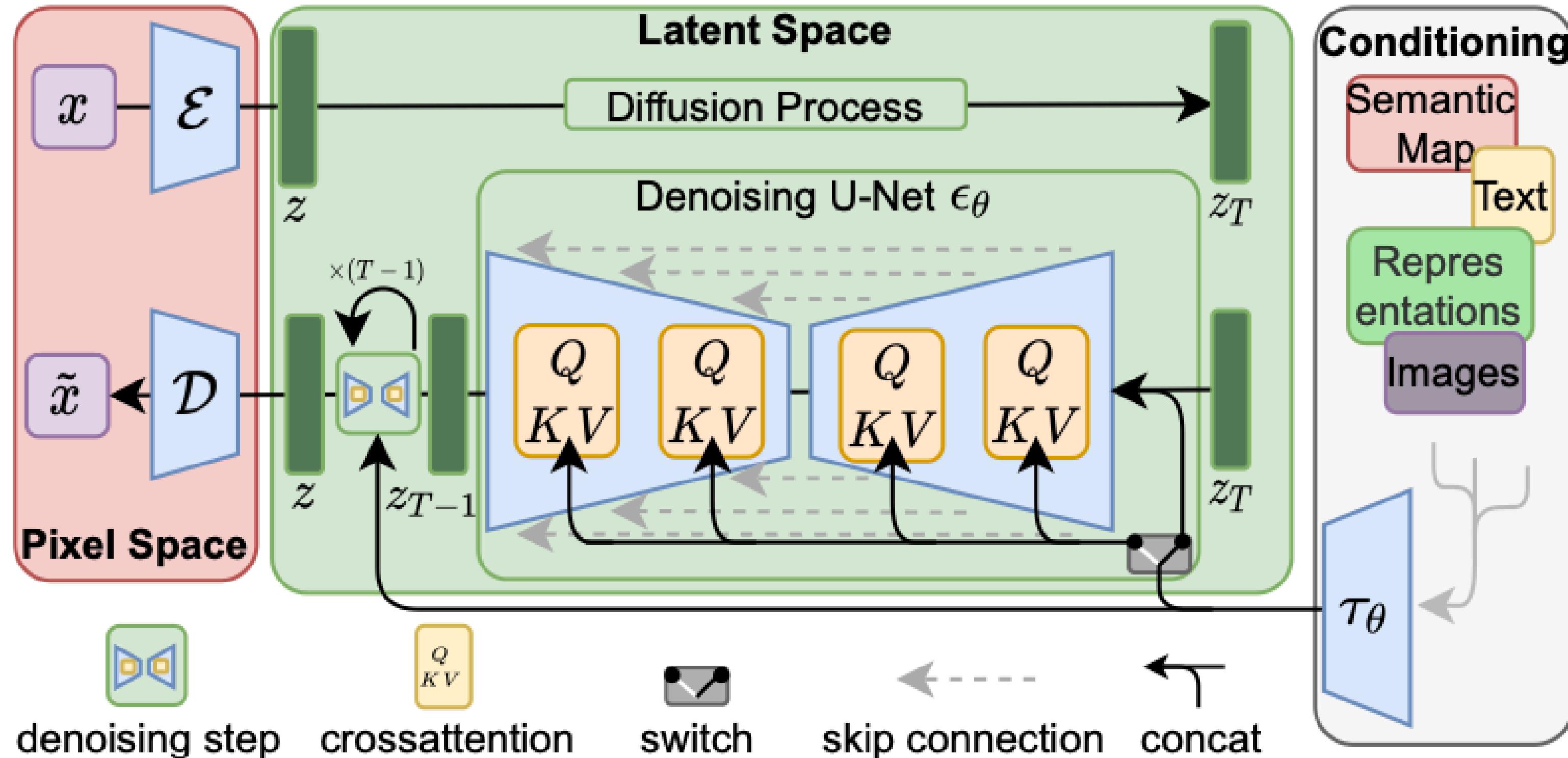
Diffusion Models: training for conditioning

(1) Contrastive pre-training

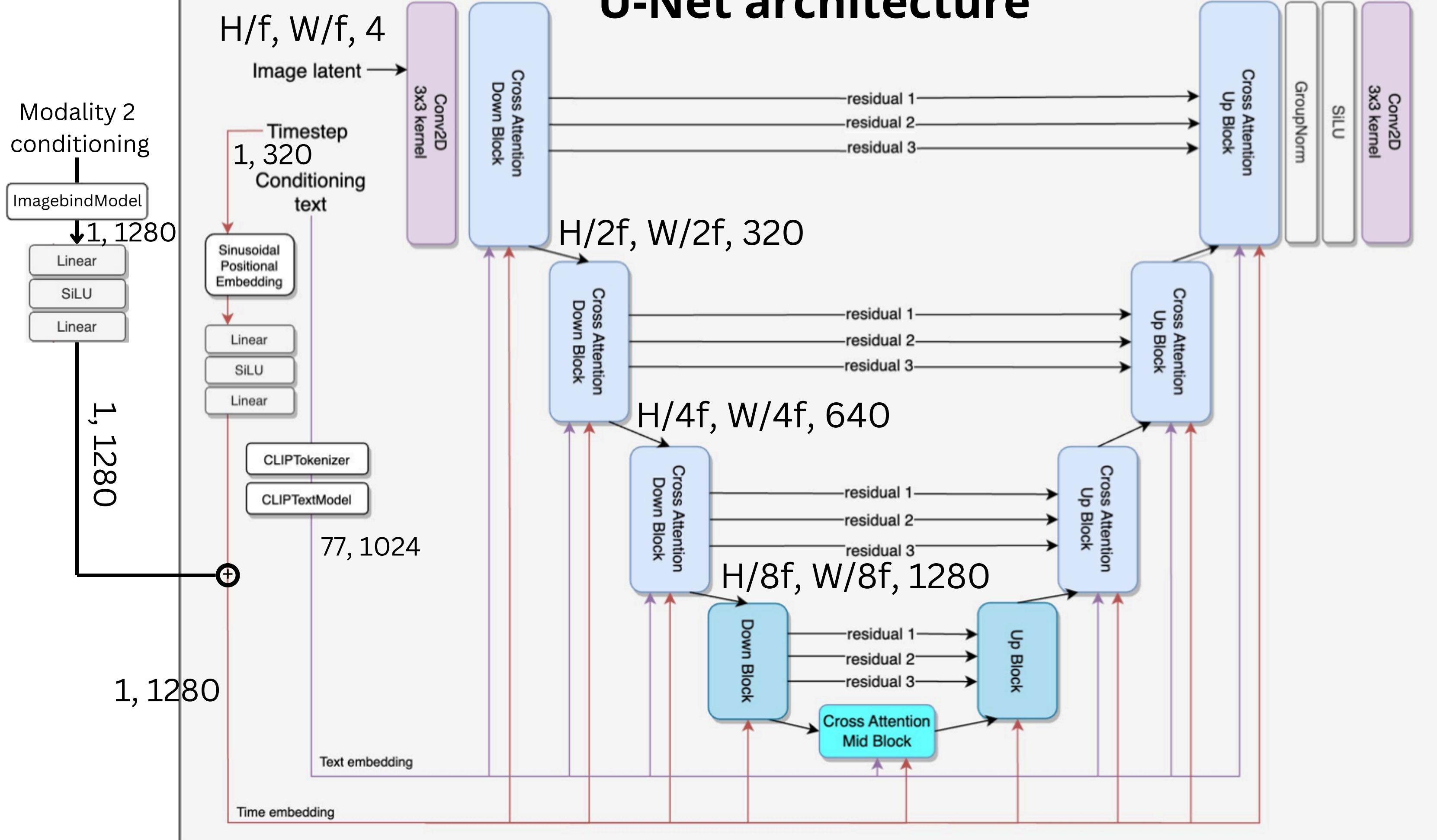


Fine tuned to generate images conditioned on CLIP image embeddings

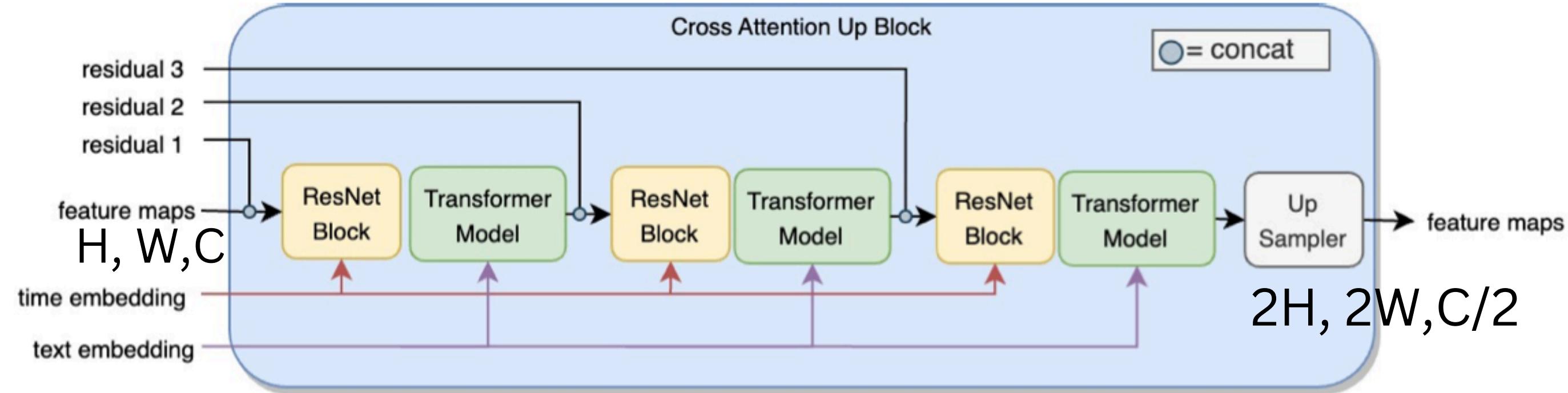
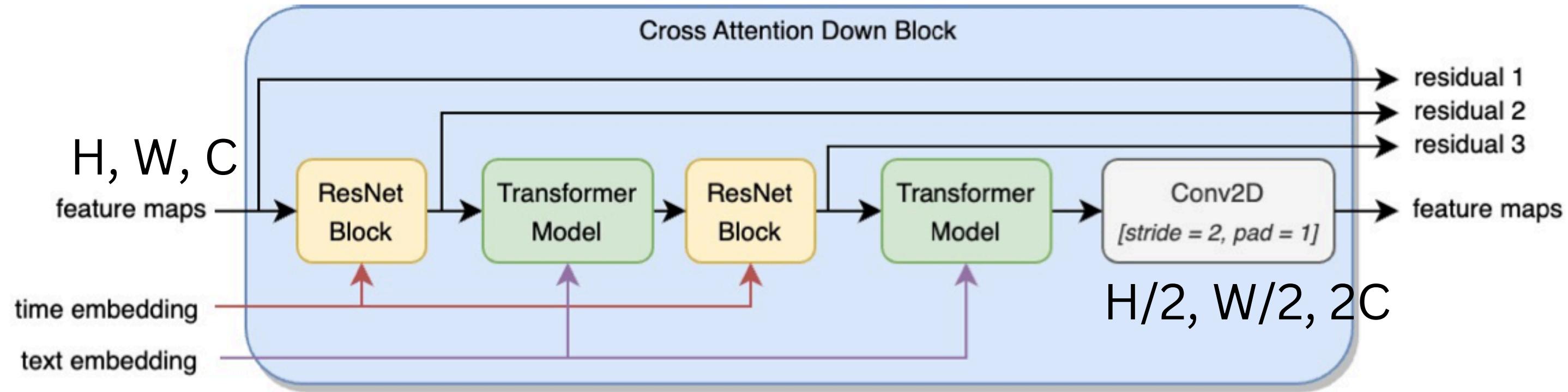
Latent Diffusion: the big picture



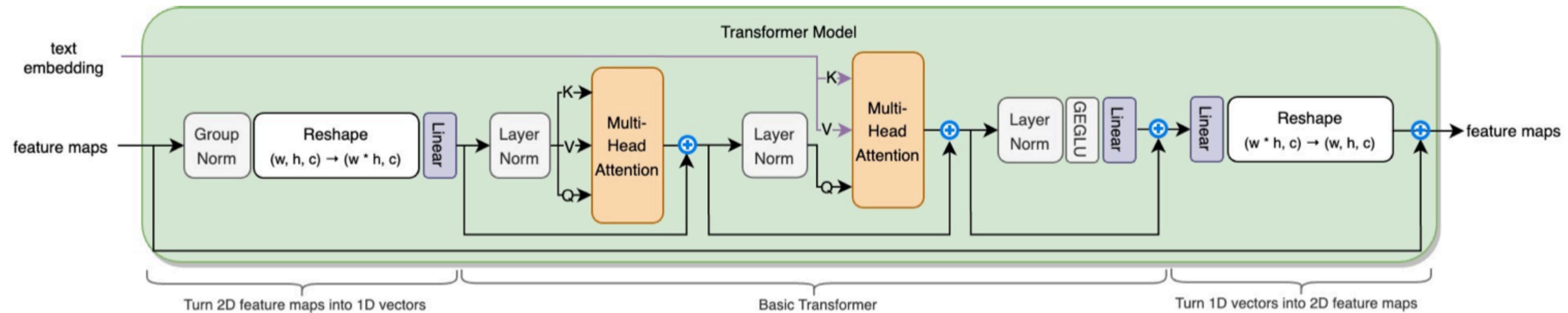
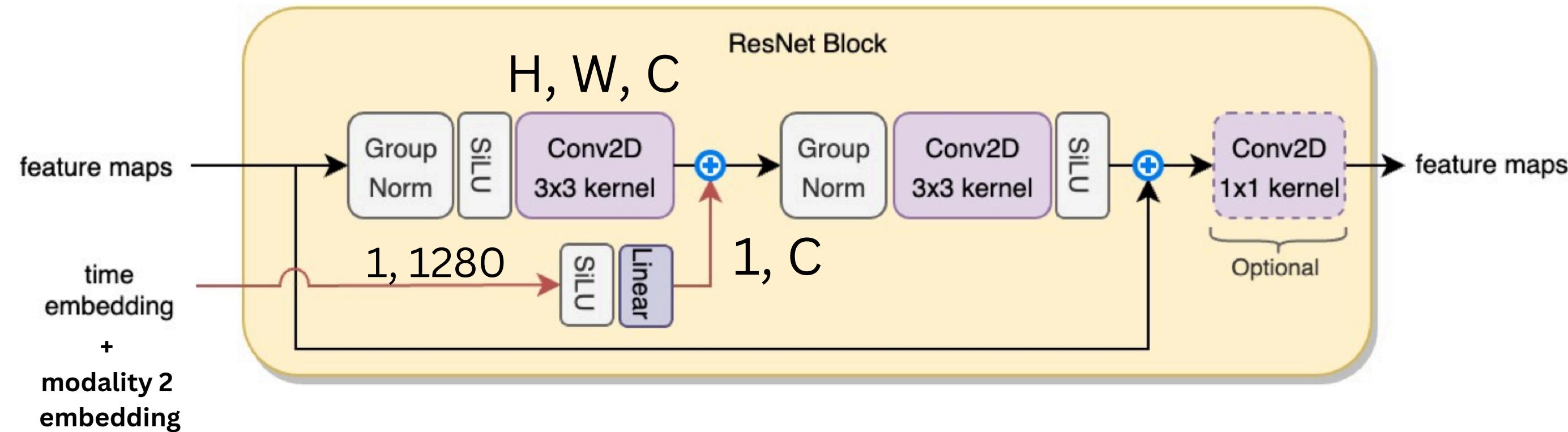
U-Net architecture



Cross - Attention blocks



Residual blocks and Transformer blocks



Parameters of Image Generation

- Unconditional guidance: c_{unc} $\epsilon_\theta = \epsilon_\theta(x, \emptyset) + c_{unc} * (\epsilon_\theta(x, c) - \epsilon_\theta(x, \emptyset))$
Often a negative prompt
- Conditioning strength: c_{str} $\mathbf{c} = \mathbf{c}_{str} \cdot \mathbf{c}$
- Image destruction (image variation problem)
$$\left\{ \begin{array}{l} 0 = \text{original image} \\ 1 = \text{complete destruction} \end{array} \right.$$
- Weight of conditionings: α $\mathbf{c} = \alpha \cdot \mathbf{c}_1 + (1 - \alpha) \cdot \mathbf{c}_2$
- Number of denoising steps: set by default to 75

Experiment 1: varying C_{unc}

1



4



7



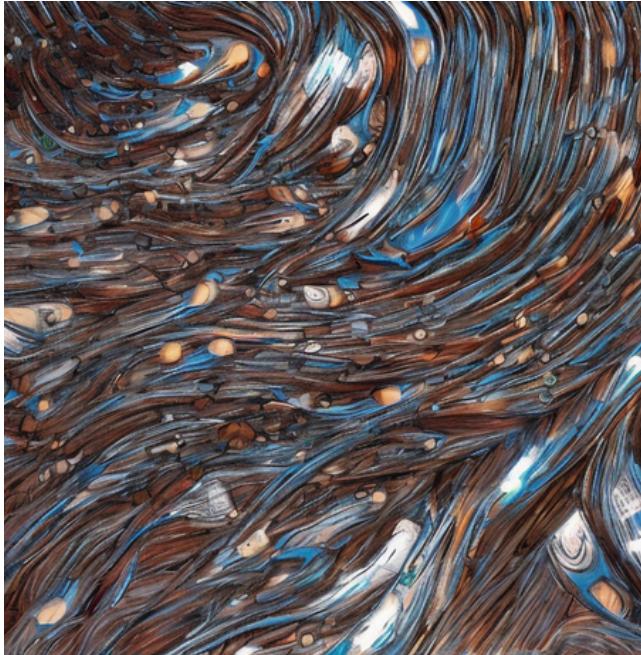
10



13



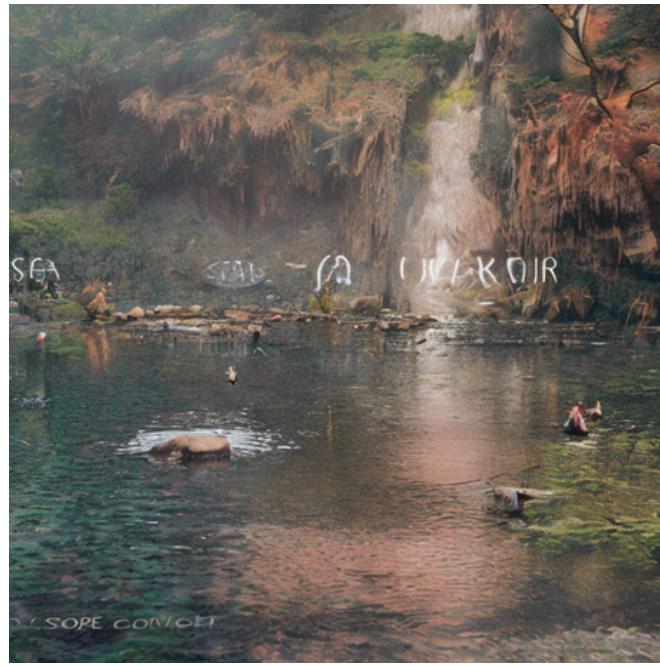
“Japanese girl, 18-21 years old, brown hair [...]”



“A girl with blue hair with eyes closed [...]”

Experiment 1: varying C_{unc}

1



4



7



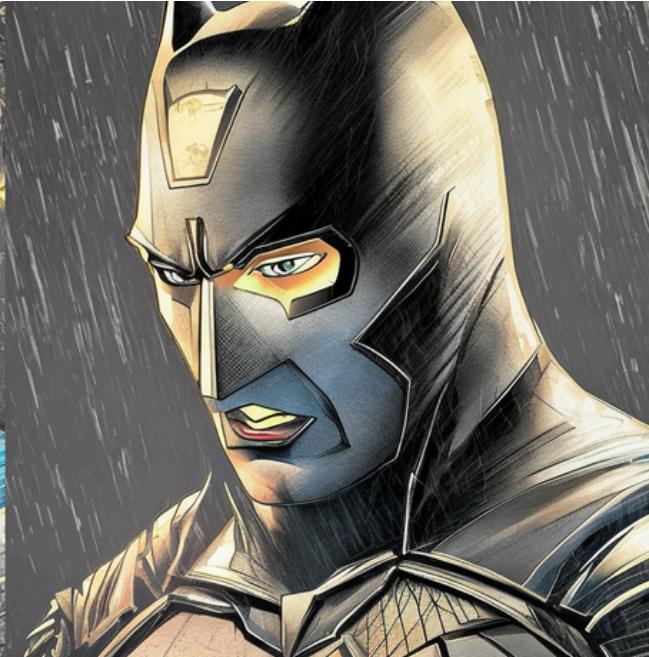
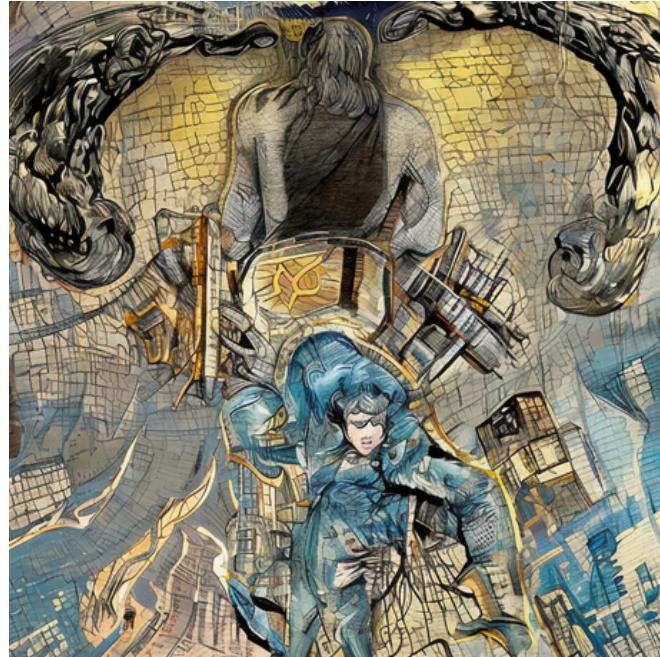
10



13



“Photo of a mountain lake landscape, clear water [...]”



“Digital wallpaper portrait of batman, dark knight [...]”

Experiment 2: varying C_{str}



+ “Autumn landscape”



0

0.25

0.5

0.75

1.0

1.25

1.5

Generated with 75 Steps of diffusion

Experiment 2: varying C_{str}



+

“Red dragon in the
waves art print”



0

0.25

0.5

0.75

1.0

1.25

1.5

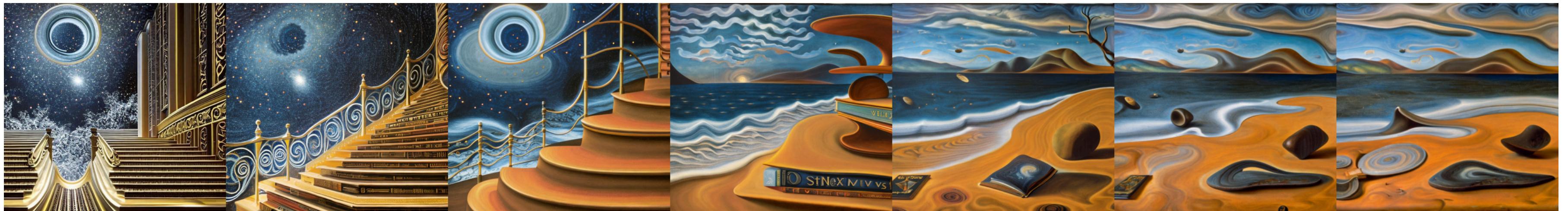
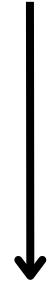
Generated with 75 Steps of diffusion

Experiment 2: varying C_{str}



+

“A dreamy illustration of an infinite staircase leading to the universe”



0

0.25

0.5

0.75

1.0

1.25

1.5

Generated with 75 Steps of diffusion

Experiment 2: varying C_{str}



thunder.wav

+

“[...] mountain, peach trees,
dreamlike atmosphere”



0

0.25

0.5

0.75

1.0

1.25

1.5

Generated with 75 Steps of diffusion

Experiment 3: varying image destruction

Initial Image



Conditioning



+
↓

+ “A painting”



0.1

0.2

0.3

0.4

0.5

0.6

0.7

0.8

0.9

Generated with 75 Steps of diffusion

Experiment 3: varying image destruction

Initial Image



Conditioning

“Robocop angry robot”

+



0.1

0.2

0.3

0.4

0.5

0.6

0.7

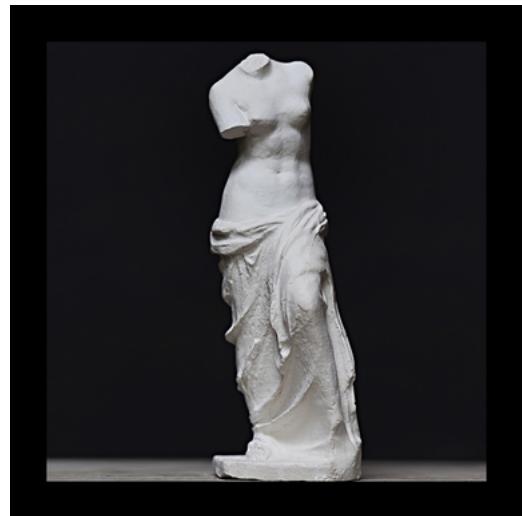
0.8

0.9

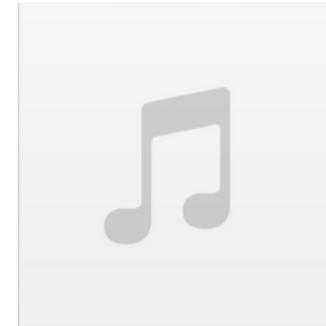
Generated with 75 Steps of diffusion

Experiment 3: varying image destruction

Initial Image



Conditioning



+ “A statue”

firewood.wav

+



0.3

0.4

0.5

0.6

0.7

0.8

0.9

Generated with 75 Steps of diffusion

Experiment 3: varying image distribution

Initial Image



Conditioning



+ “A photo”



0.3

0.4

0.5

0.6

0.7

0.8

0.9

Generated with 75 Steps of diffusion

Experiment 4: varying α

α



$1 - \alpha$



+

↓



0 0.1 0.2 0.3 0.4 0.5 0.6 0.7

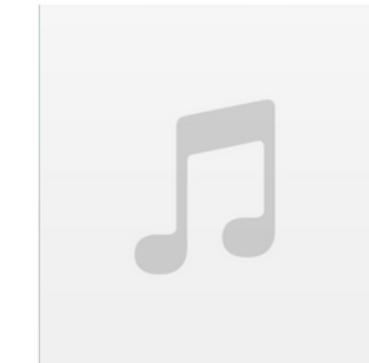
Generated with 75 Steps of diffusion

Experiment 4: varying α

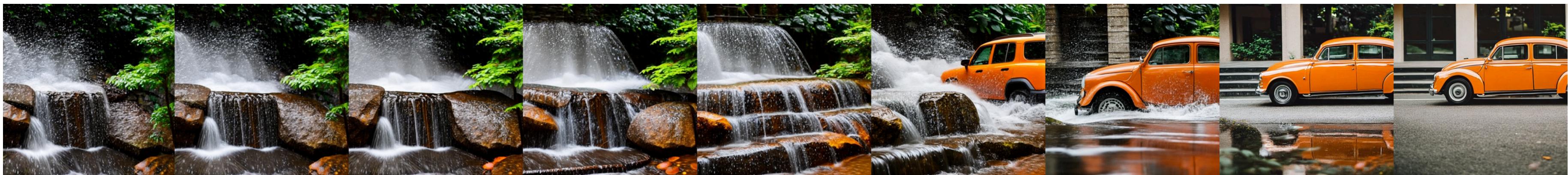
α



$1 - \alpha$



rain.wav



0.1

0.2

0.3

0.4

0.5

0.6

0.7

0.8

0.9

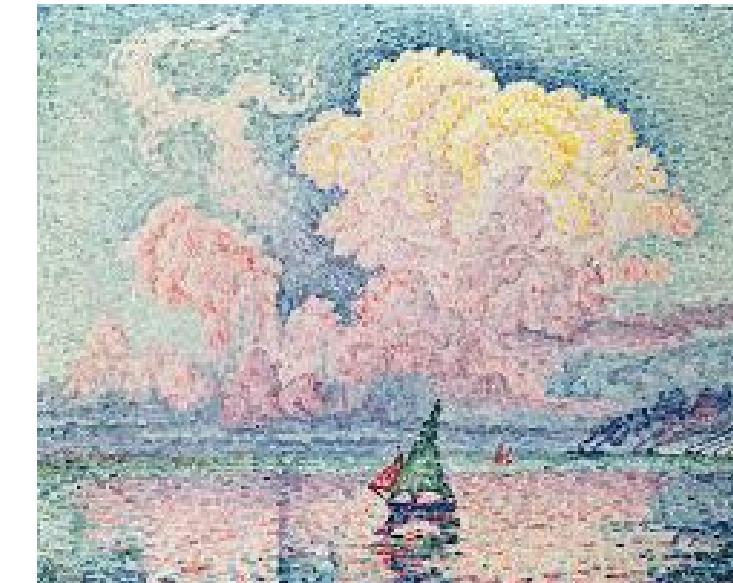
Generated with 75 Steps of diffusion

Experiment 4: varying α

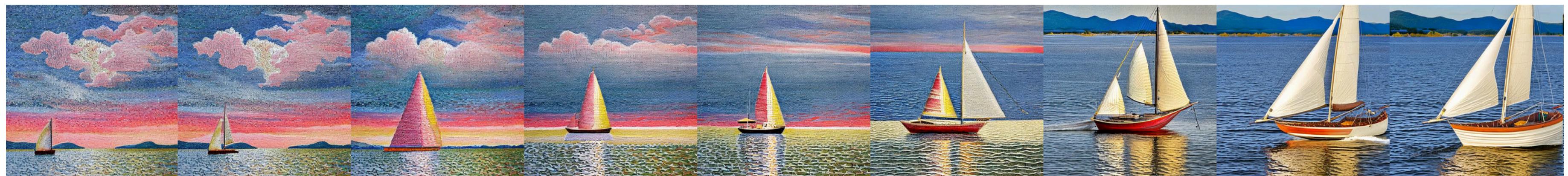
α



$1 - \alpha$



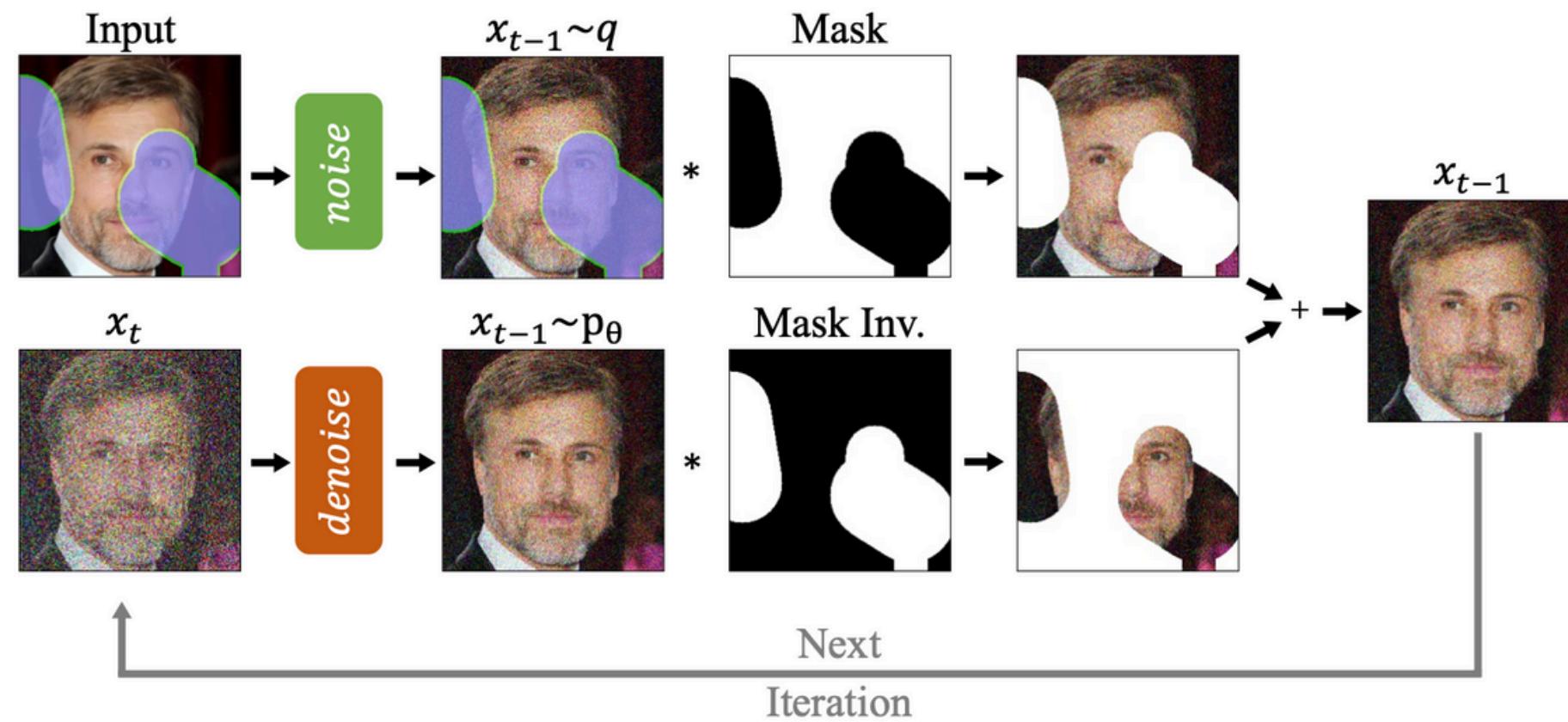
+
↓



0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8

Generated with 75 Steps of diffusion

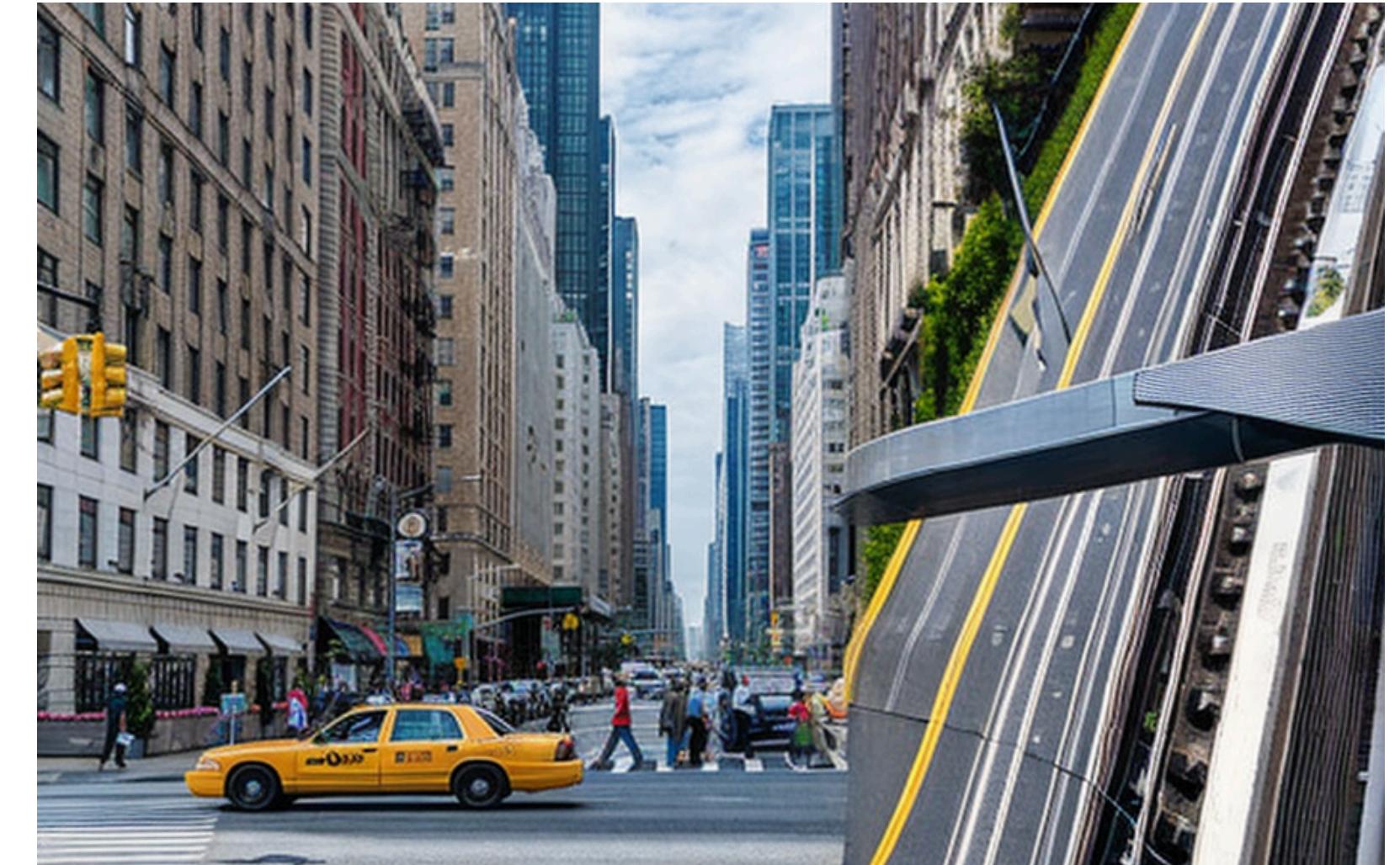
Experiment 5: inpainting



+ “A dog on a bench”

Limitations

- Very inconsistent inpainting results



Limitations

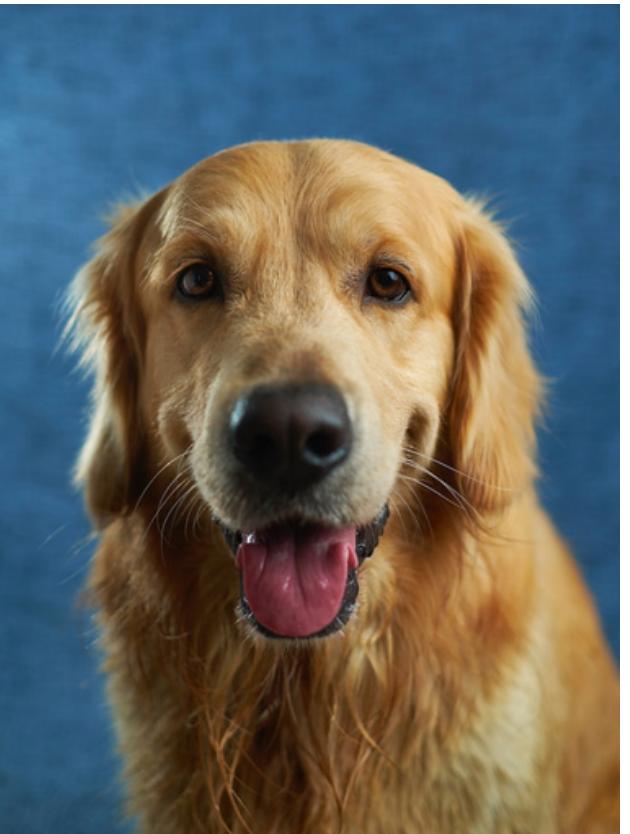
- Not able to do image removal consistently



Limitations

- Very sensitive to hyperparameters

Conditioning
Image



$c_{str} = 1.0$



$c_{str} = 0.5$



Limitations

- Audio embedding unpredictable visual meaning

