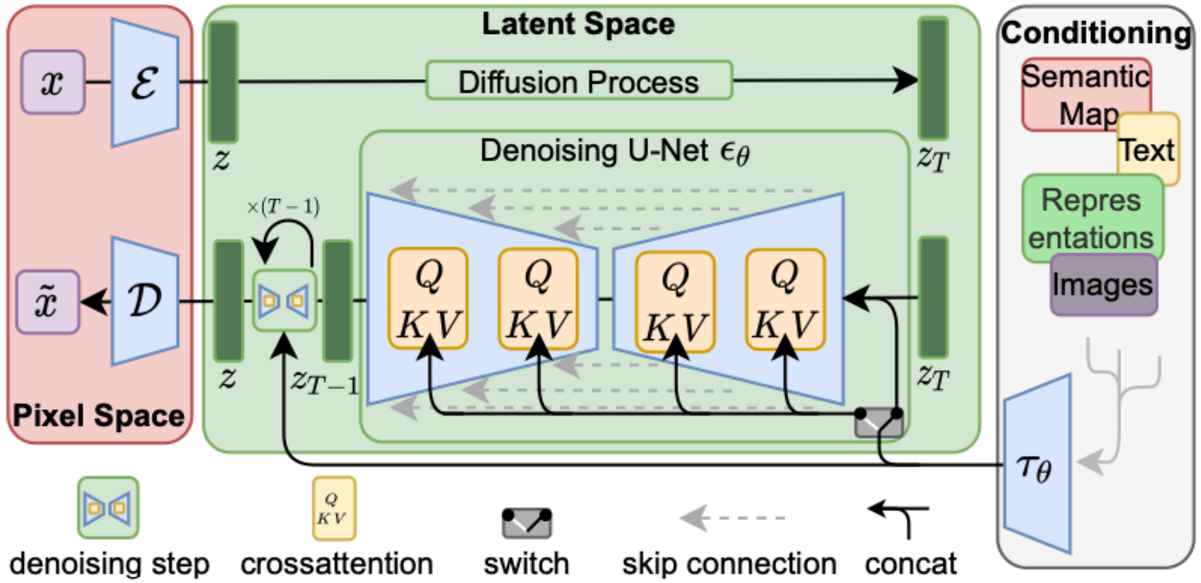
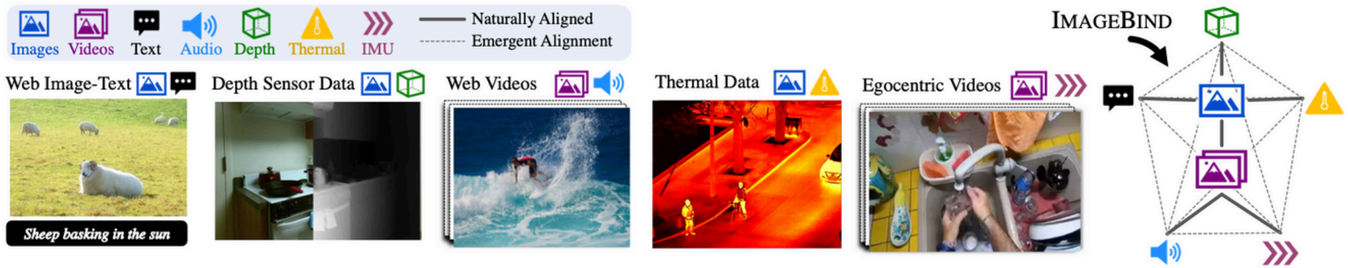
DiffusionCraft



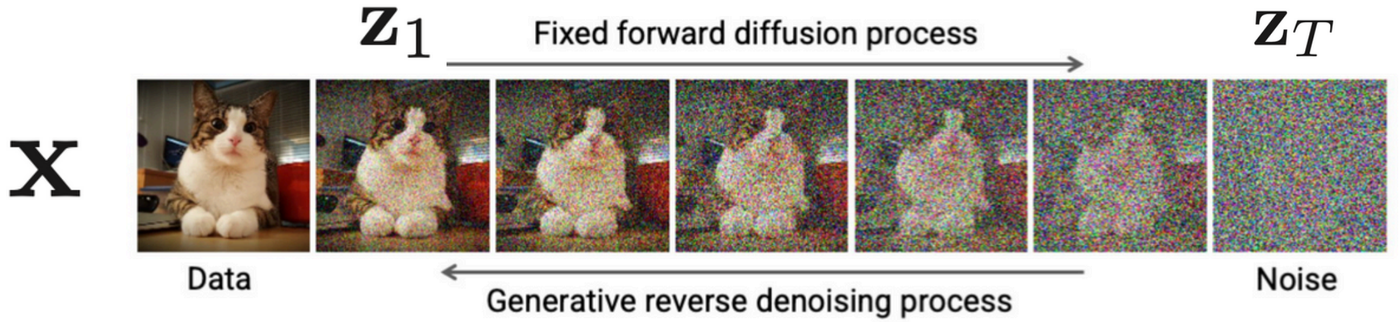
Stable Diffusion

ImageBind



Diffusion Models: an overview

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



To be learned

$$
\begin{aligned}
Pr(\mathbf{z}_{t-1}|\mathbf{z}_t,\boldsymbol{\phi}_t) &= \text{Norm}_{\mathbf{z}_{t-1}}\Big[\mathbf{f}_t[\mathbf{z}_t,\boldsymbol{\phi}_t],\sigma_t^2\mathbf{I}\Big] \\
Pr(\mathbf{x}|\mathbf{z}_1,\boldsymbol{\phi}_1) &= \text{Norm}_{\mathbf{x}}\Big[\mathbf{f}_1[\mathbf{z}_1,\boldsymbol{\phi}_1],\sigma_1^2\mathbf{I}\Big]
\end{aligned}
$$

Loss function, obtained minimising -ELBO

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

Diffusion Models: a few adjustments

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

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

1/50 to 1/100 number of steps

Make it faster with DDIM:
Skip some denoising steps

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

Algorithm 18.1: Diffusion model training

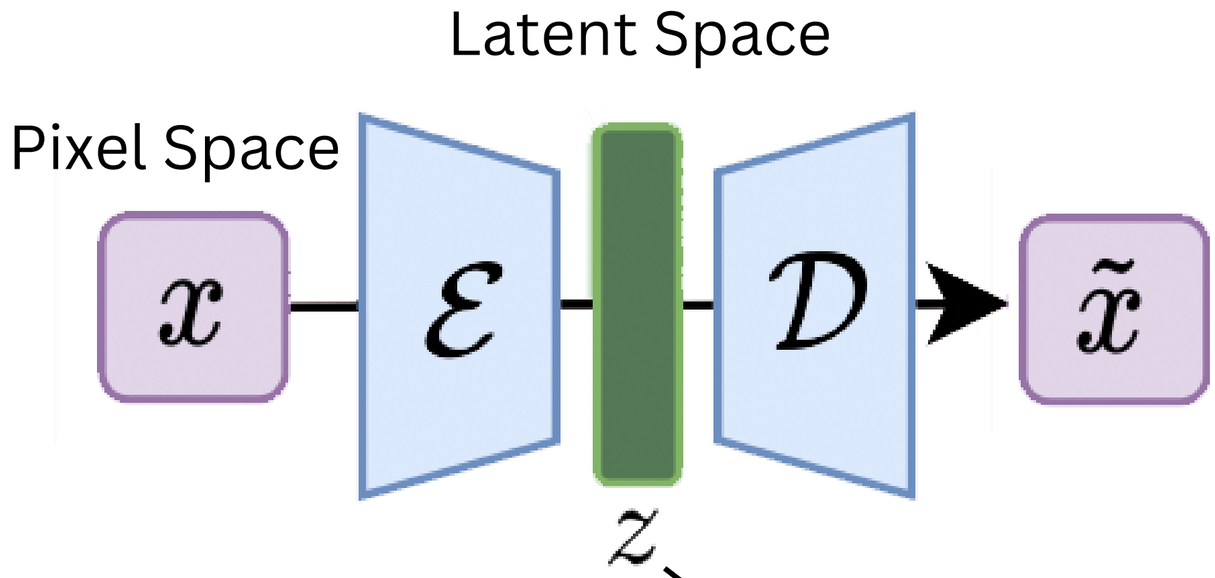
Input: Training data $\mathbf{x}$
Output: Model parameters $\boldsymbol{\phi}_t$
repeat
  for $i \in \mathcal{B}$ **do** // For every training example index in batch
    $t \sim \text{Uniform}[1, \ldots T]$ // Sample random timestep
    $\boldsymbol{\epsilon} \sim \text{Norm}[\mathbf{0}, \mathbf{I}]$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

Algorithm 18.2: Sampling

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

Latent Diffusion Models

Intermediate latent representation by a VAE

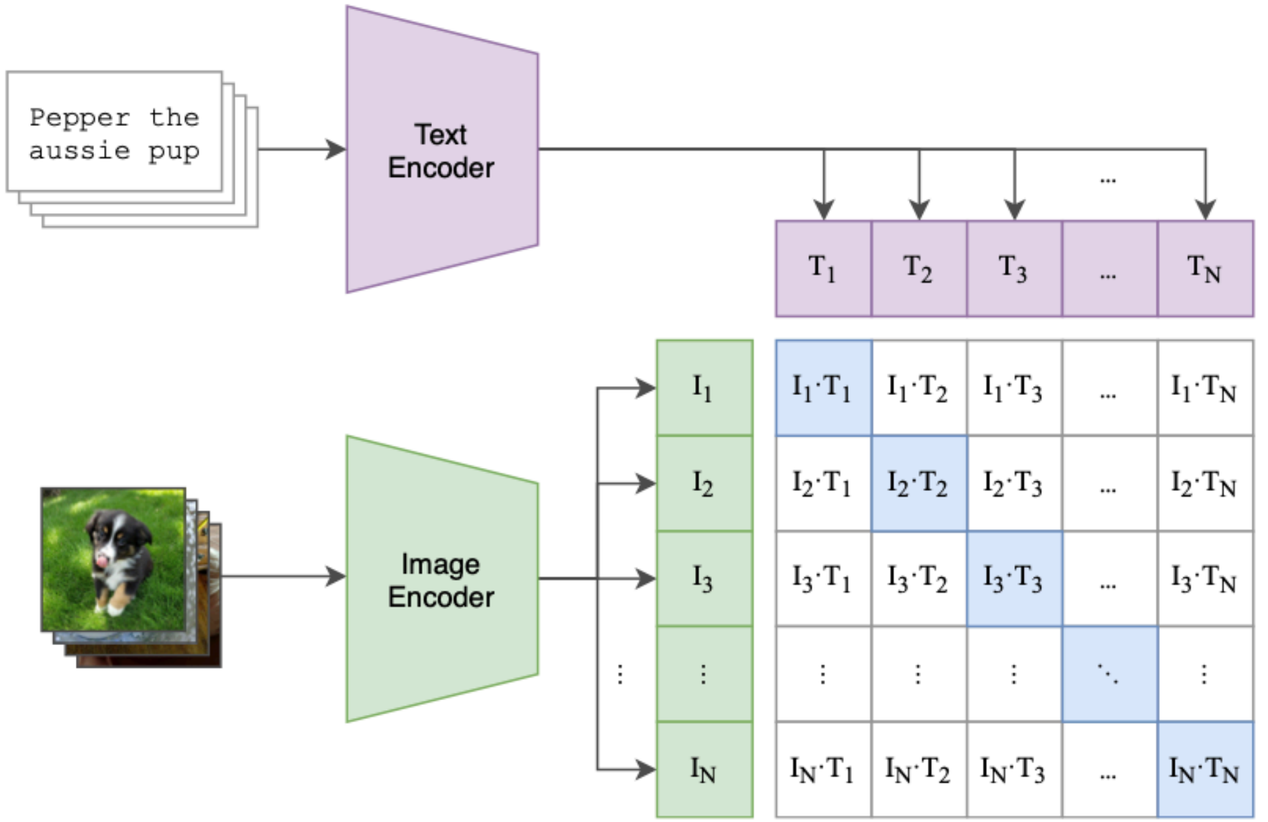


$$|z| \ll |x|$$

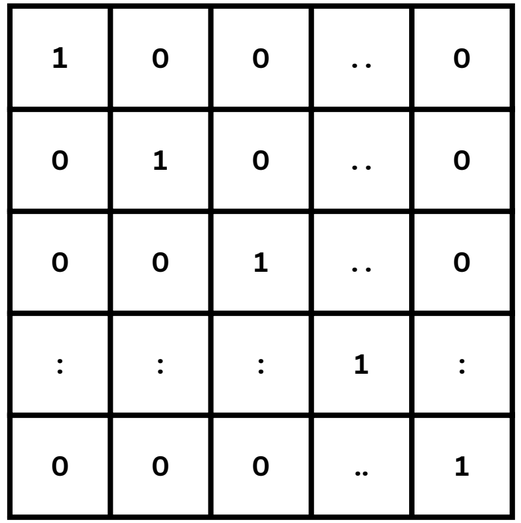
keeps the H x W x C structure

$$z \approx \mathcal{N}(0, I)$$

CLIP: aligned text - image embeddings



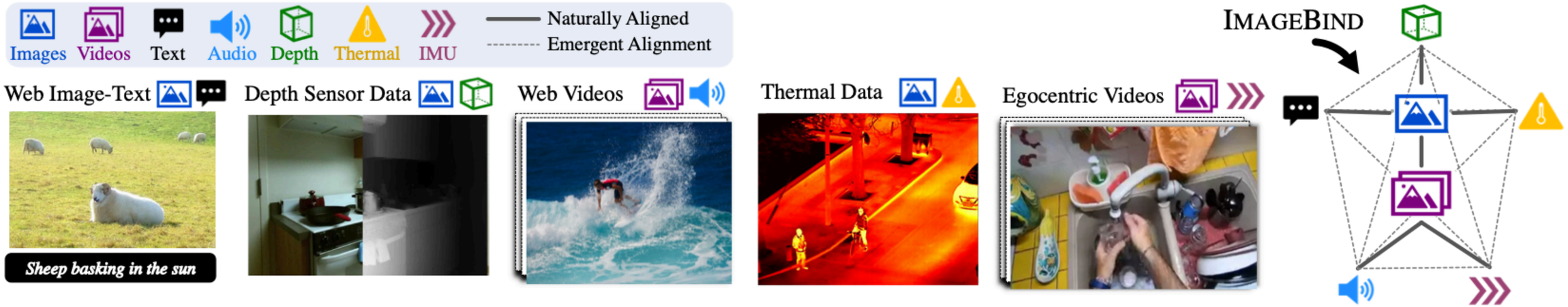
Contrastive Learning



Trained with Symmetric Cross Entropy 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

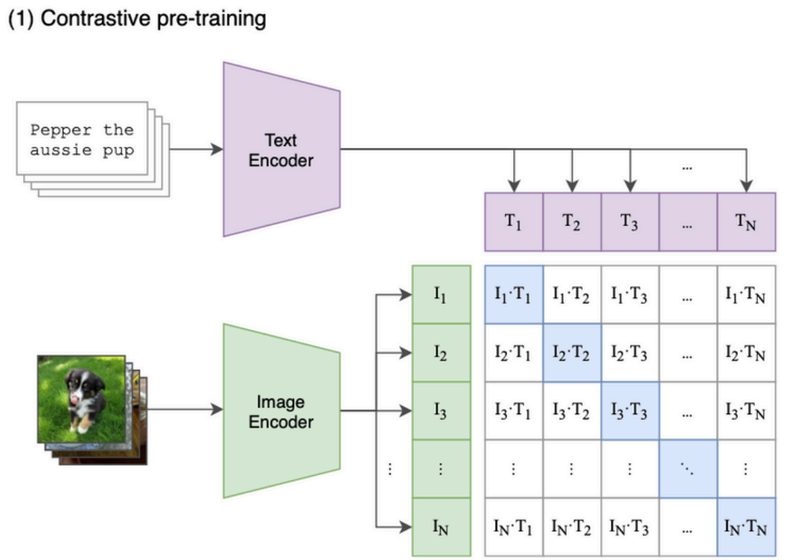
Trained with InfoCNE loss:

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

Image encoder

Modality 2 encoder

Diffusion Models: training for conditioning



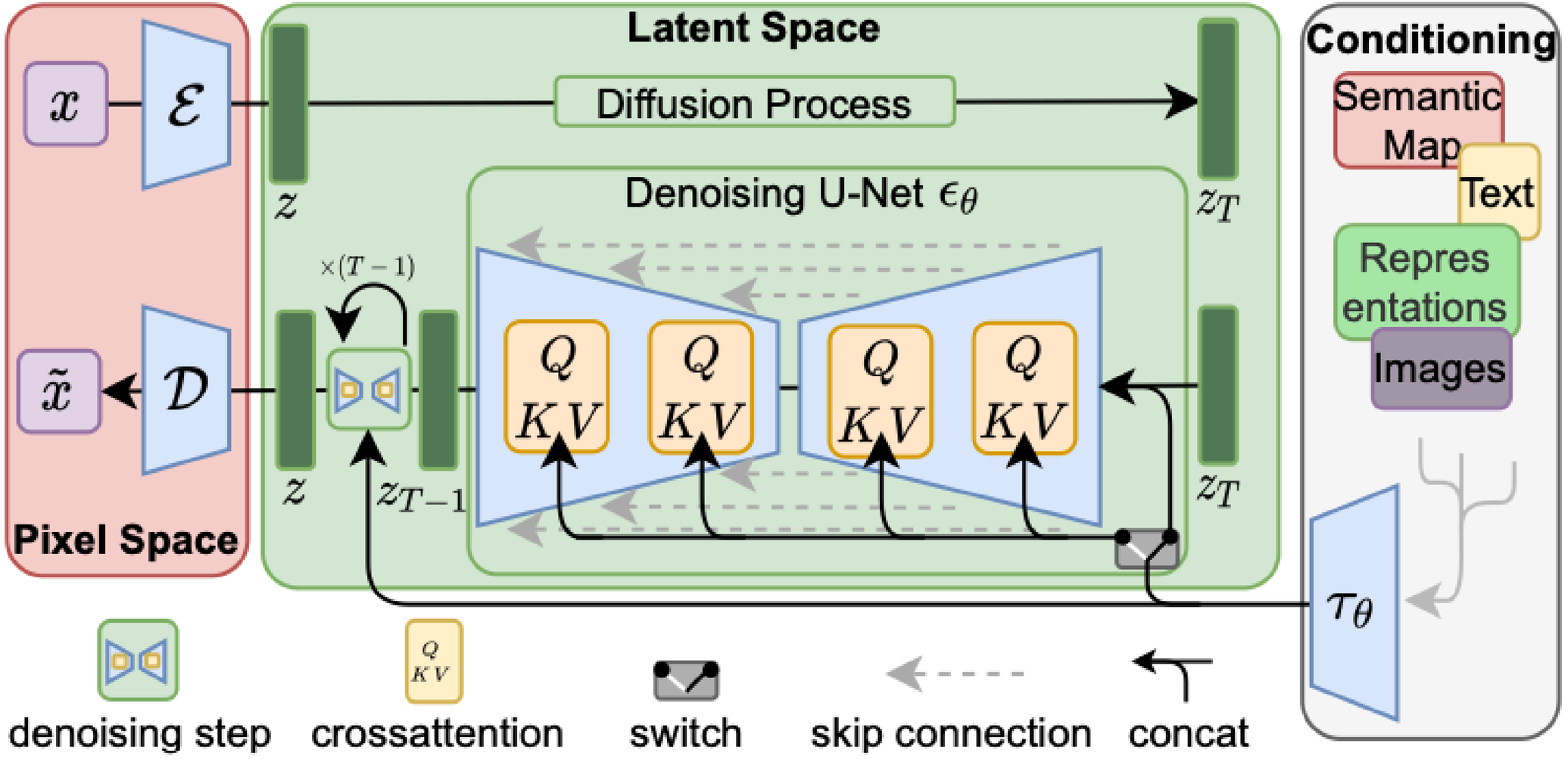
Text embedding fed to the network

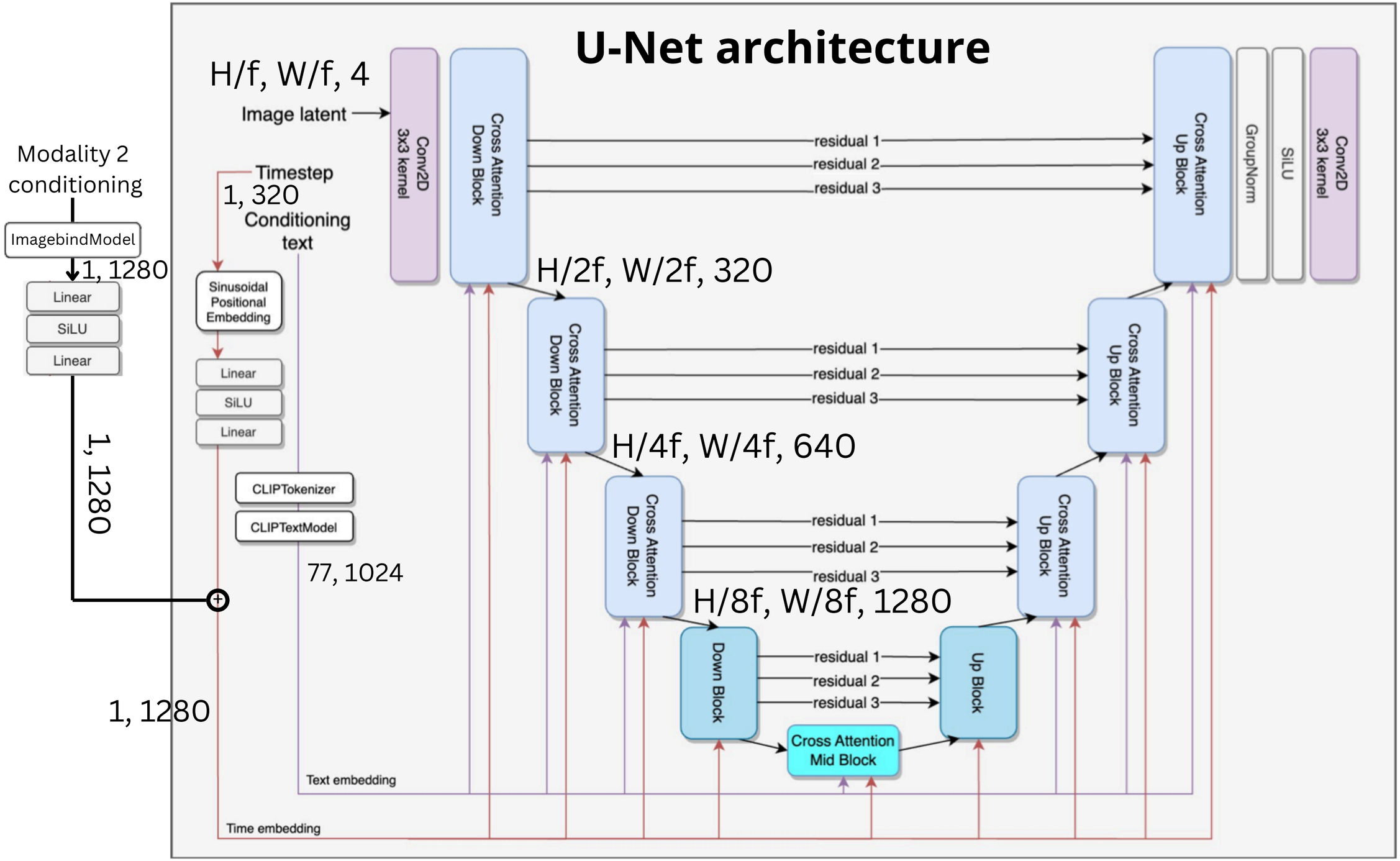
Learn to reconstruct corresponding image

Condition on text

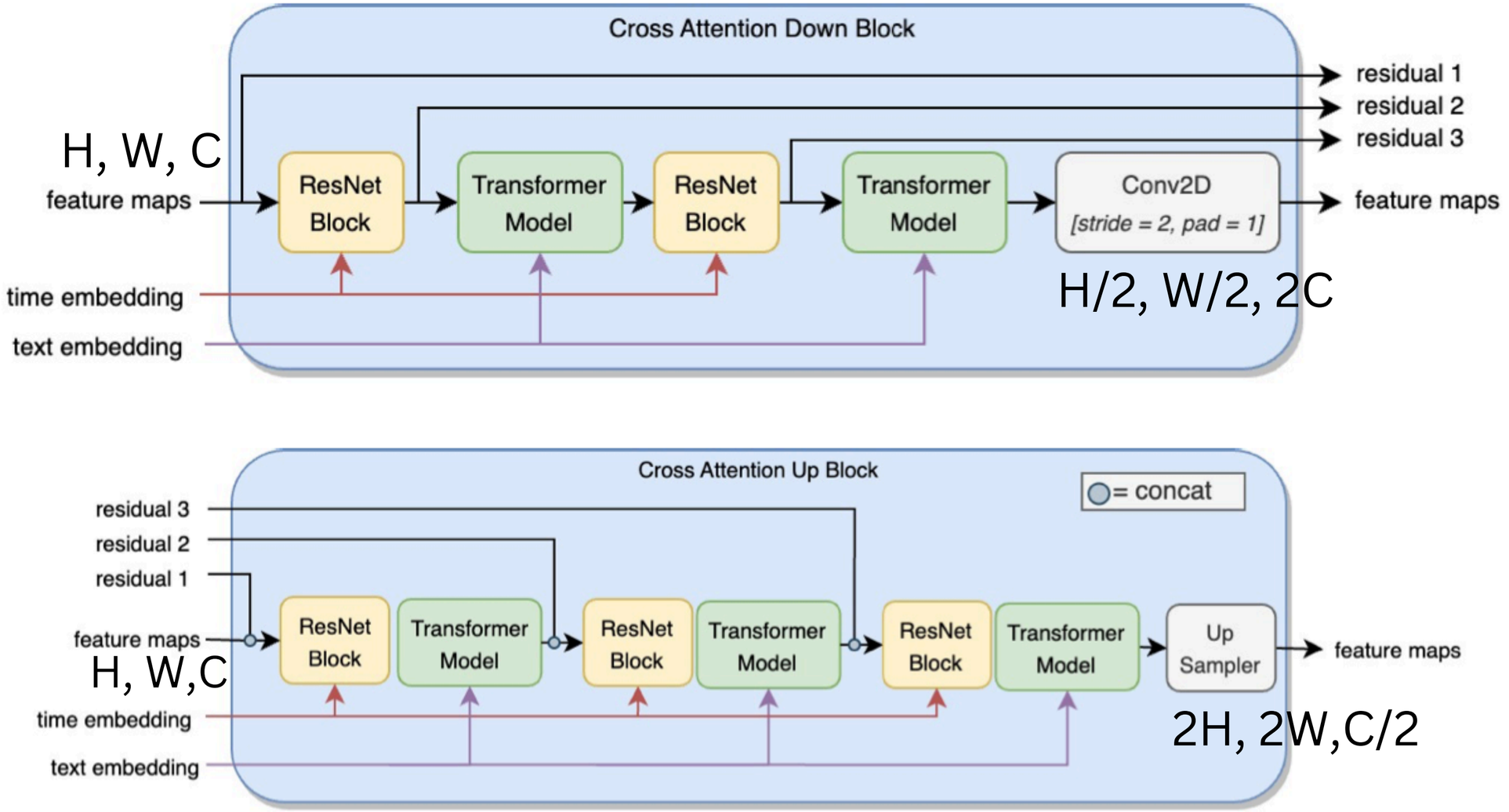
Fine tuned to generate images conditioned on CLIP image embeddings

Latent Diffusion: the big picture

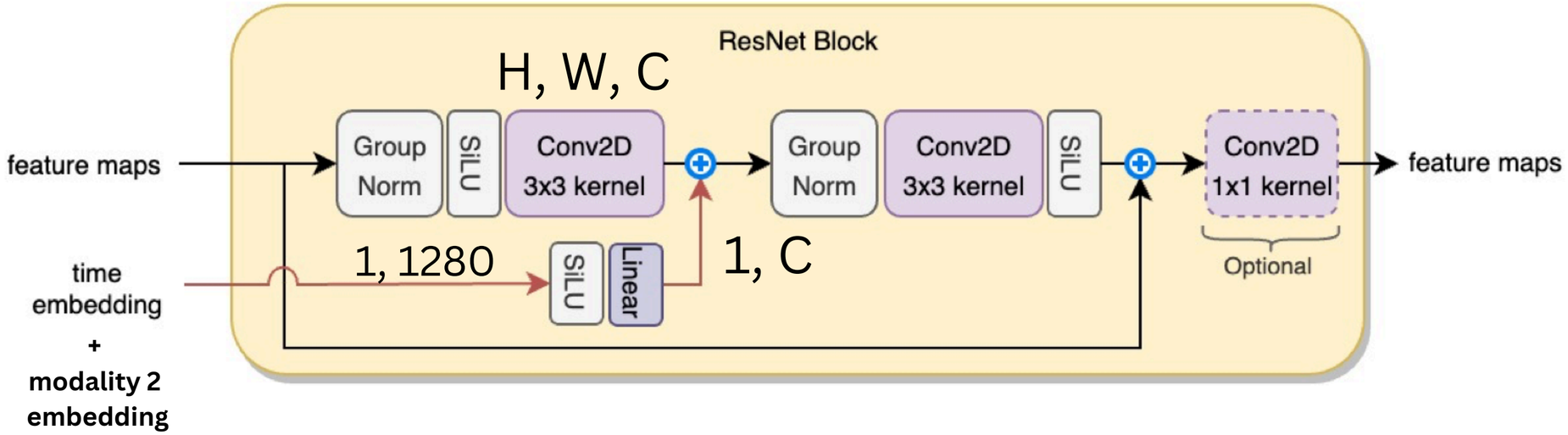


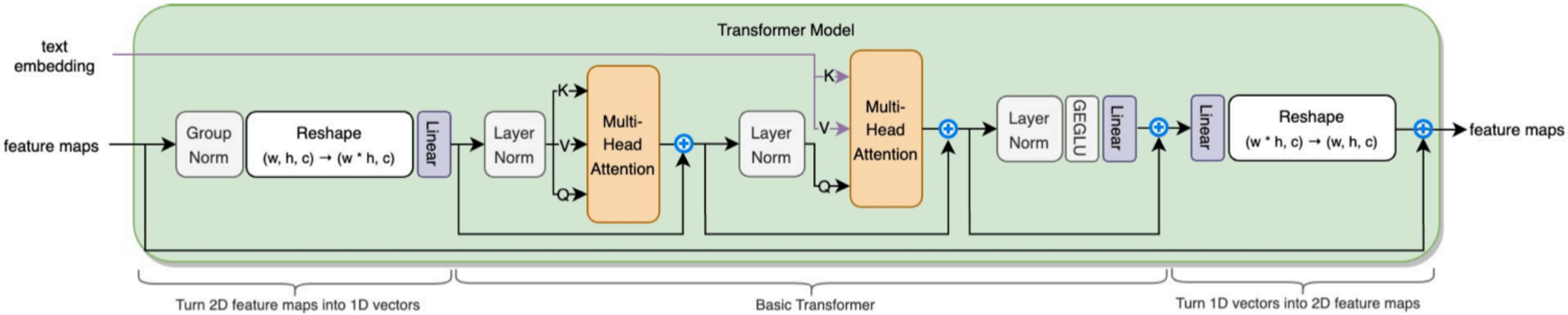

U-Net architecture
H/f, W/f, 4
Image latent
Conv2D 3x3 kernel
Cross Attention Down Block
residual 1
residual 2
residual 3
Cross Attention Up Block
GroupNorm
SiLU
Conv2D 3x3 kernel
Modality 2 conditioning
ImagebindModel
1, 1280
Linear
SiLU
Linear
1, 1280
Timestep
1, 320
Sinusoidal Positional Embedding
Linear
SiLU
Linear
Conditioning text
CLIPTokenizer
CLIPTextModel
77, 1024
H/2f, W/2f, 320
H/4f, W/4f, 640
H/8f, W/8f, 1280
Down Block
Up Block
Cross Attention Mid Block
1, 1280
Text embedding
Time embedding

Cross - Attention blocks



Residual blocks and Transformer blocks





Parameters of Image Generation

- Unconditional guidance: c_{unc} $\epsilon_\theta = \epsilon_\theta(x, \emptyset) + c_{unc} * (\epsilon_\theta(x, c) - \epsilon_\theta(x, \emptyset))$

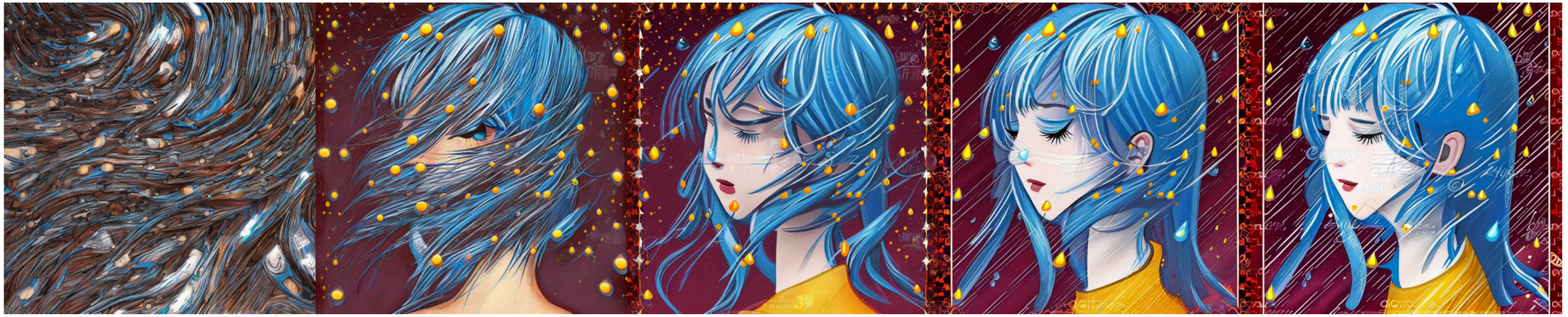
Often a negative prompt

- Conditioning strength: $\mathbf{c}_{str}$ $\mathbf{c} = \mathbf{c}_{\mathrm{str}} \cdot \mathbf{c}$
- Image destruction (image variation problem)
  - 0 = original image
  - 1 = complete destruction
- Weight of conditionings: α $\mathbf{c} = \boldsymbol{\alpha} \cdot \mathbf{c_1} + (1 - \boldsymbol{\alpha}) \cdot \mathbf{c_2}$
- 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 [...]”

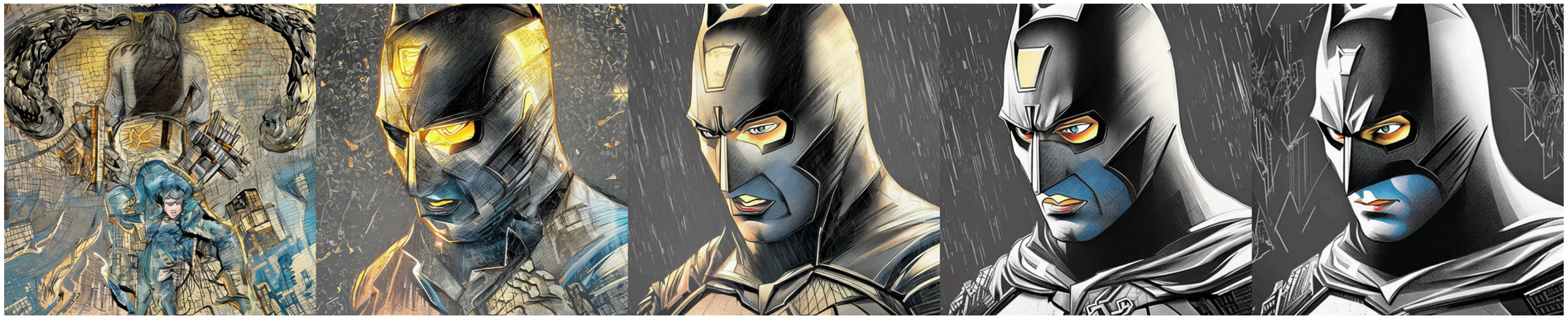


“A girl with blue hair with eyes closed [...]”

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 $\mathbf{c}_{str}$

+ “Autumn landscape”

0 0.25 0.5 0.75 1.0 1.25 1.5

Generated with 75 Steps of diffusion

Experiment 2: varying $\mathbf{c}_{str}$

+

"Red dragon in the waves art print"

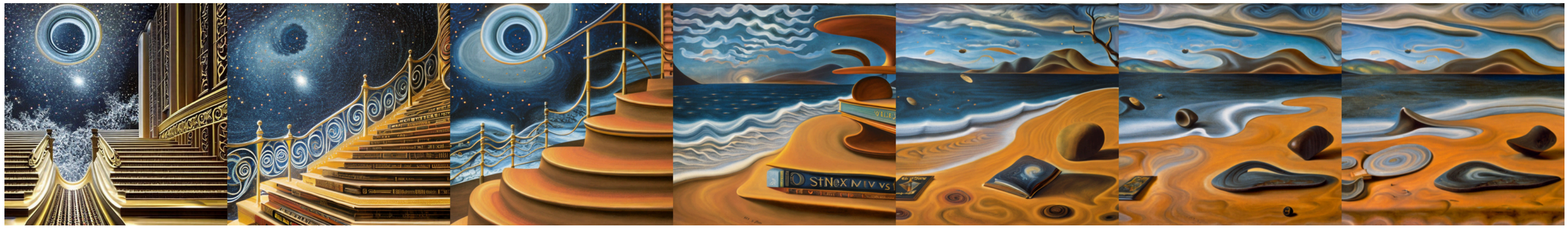
0 0.25 0.5 0.75 1.0 1.25 1.5

Generated with 75 Steps of diffusion

Experiment 2: varying $\mathbf{c}_{str}$

+

"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 $\mathbf{C}_{str}$

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 distruction

Initial Image

+

Conditioning

+ “A painting”

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

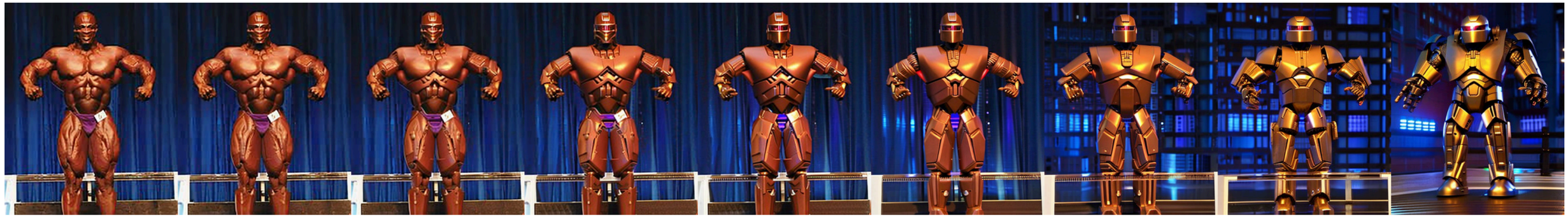
Generated with 75 Steps of diffusion

Experiment 3: varying image distruction

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 distruction

Initial Image

+

Conditioning

firewood.wav

+ “A statue”

0.3 0.4 0.5 0.6 0.7 0.8 0.9

Generated with 75 Steps of diffusion

Experiment 3: varying image distruction

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

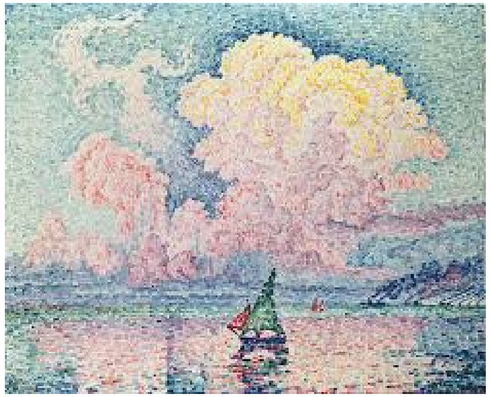
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

Generated with 75 Steps of diffusion

Experiment 4: varying α

α $1 - \alpha$

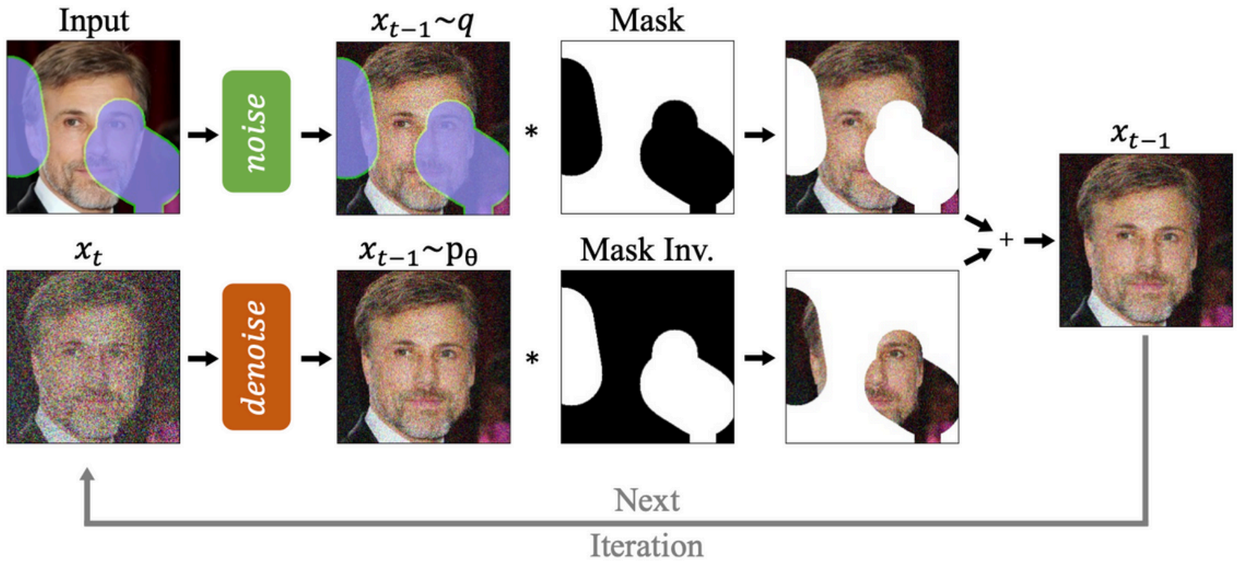
+



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

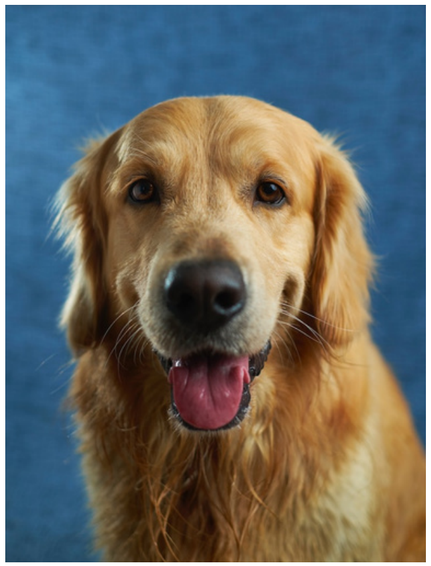
Limitations

- 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

thunder.wav

+

“a landscape“