Deep Generative Models: Image Editing with Diffusion Models

Fall Semester 2026

René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS)
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

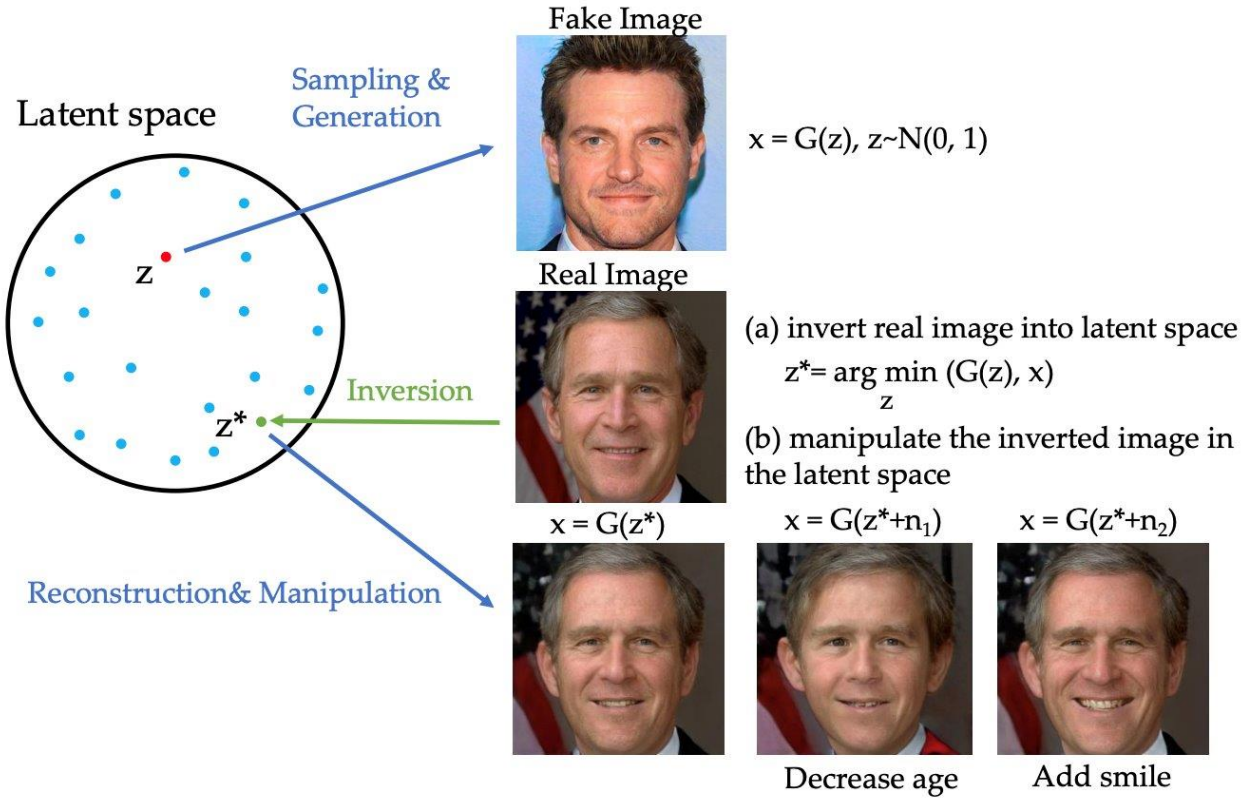
Penn
University of Pennsylvania

Outline

- Markov Hierarchical Variational Auto Encoders (MHVAEs)
  - Autoregressive Encoder and Autoregressive Decoder of an MHVAE
  - Derivation of the ELBO of an MHVAE
- Diffusion Models: MHVAEs with a Linear Gaussian Autoregressive Latent Space
  - Forward Diffusion Process
  - Reverse Diffusion Process
  - ELBO for Diffusion Models as a particular case of the ELBO for MHVAEs
- Implementation Details: UNet architecture, Training and Sampling Strategies
- Application of Diffusion Models
  - Text-Conditioned Diffusion Model: Stable Diffusion
  - Multimodal Control for Consistent Synthesis: ControlNet
  - **Image Editing: DDIM, P2P**

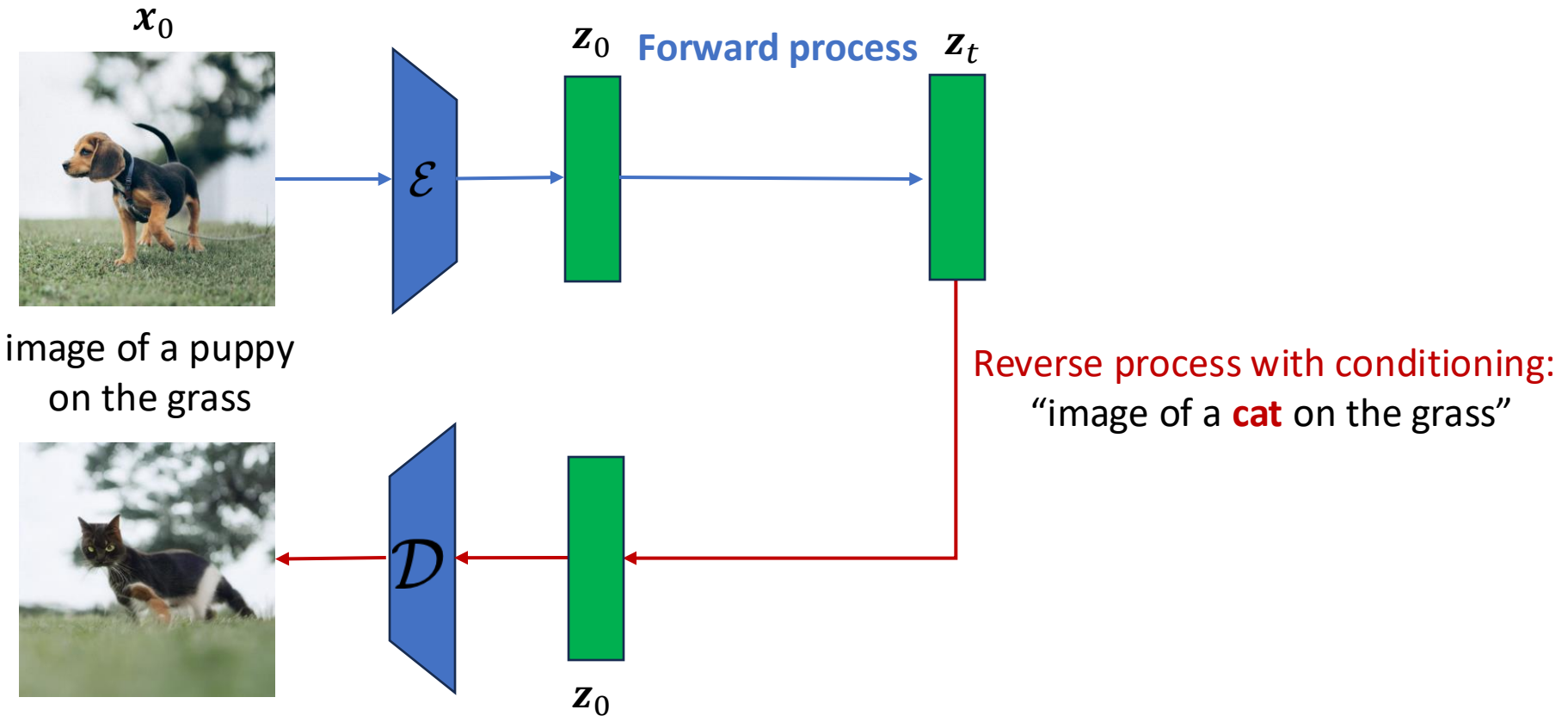
Latent Space Image Editing: Inversion + Manipulation

- Diffusion models so far can be used for image generation.
- Stable Diffusion performs text-to-image conditioning in a rich latent space.
- Can we use the latent space of diffusion models to perform image editing?



Naïve Image Editing Idea

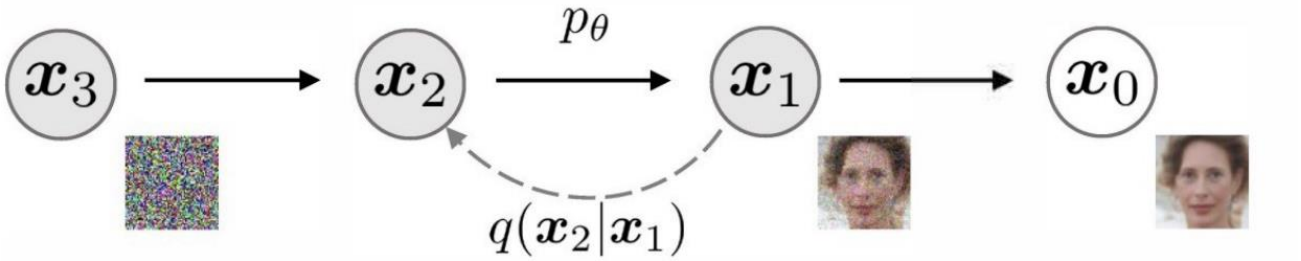
- Instead of starting from pure noise, let us perform naïve inversion using the forward process and a fixed image.



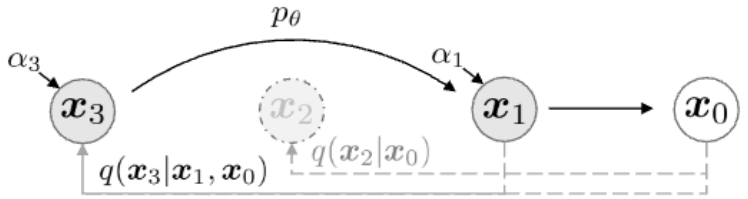
- Depending on how much noise is added, we can change a lot of features in the image or not enough features.

How to improve the inversion?

- Problems
  1. Randomness in the model: if we encode $\boldsymbol{x}_0$ to $\boldsymbol{x}_t$ using the forward process and then run the reverse process, we will not get $\boldsymbol{x}_0$.
  2. The reverse process requires T sequential steps, which can be slow.



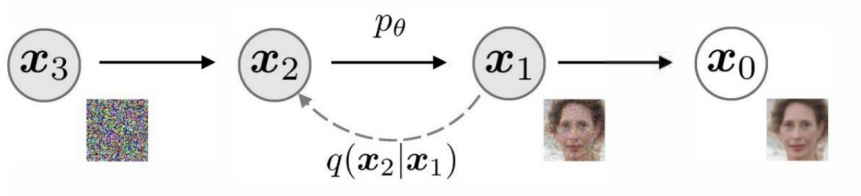
- What if we had a different sampling mechanism?

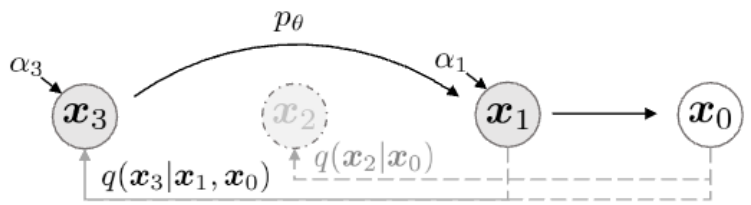


- We will introduce a sampling process that allows for better inversion and image editing.

Designing a Faster Processes

- Diffusion models are designed to approximate the reverse process of the forward process.
- Intuition: If we had a forward process with few steps, the reverse process would also require a small number of steps to sample a new image.
- How can we design sampling processes with fewer steps?





- We will generalize the Markovian forward process of DDPM to non-Markovian processes to obtain a large family of models.
- Then, we can select a diffusion process that can be simulated in few steps to achieve fast sampling!

Recall the Forward Process of DDPM

$$\overline{\alpha_t} = \prod_{i=1}^{t} \alpha_i$$

- Recall that the forward process of DDPM is Markovian and given by

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0, (1-\overline{\alpha_t})\boldsymbol{I}\right)$$

- Recall also that reversing q gives

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\overline{\alpha_{t-1}})\boldsymbol{x}_t + \sqrt{\overline{\alpha_{t-1}}}(1-\alpha_t)\boldsymbol{x}_0}{1-\overline{\alpha_t}}}_{\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\overline{\alpha_{t-1}})}{1-\overline{\alpha_t}}\boldsymbol{I}}_{\boldsymbol{\Sigma}_q(t)}\Big)$$

- Recall also that DDPM sampling process is

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)\right) + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_t$$

- Because noise is added at each step of sampling, **we cannot jump over steps**!

Recall the Loss of DDPM

- Recall also that the (unweighted) learning objective used in DDPM is

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_0),t}\left[\|\bar{\boldsymbol{\epsilon}}_t - \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)\|_2^2\right]$$

- where $\boldsymbol{x}_t = \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_t}}\bar{\boldsymbol{\epsilon}}_t$, $\bar{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ are i.i.d. and independent of $\boldsymbol{x}_t$.

- Note that $\boldsymbol{x}_{t-1}$ does **not** appear in the DDPM training loss: the neural network $\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)$ simply takes $\boldsymbol{x}_t$ as input and learns to predict the noise relative to $\boldsymbol{x}_0$.

- If the neural network only ever learns from $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$, why are we restricted to using a Markov process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ during generation?

- Could we define (1) a non-Markovian process that (2) yields the same $q(\boldsymbol{x}_t|x_0)$ so that training is the same, yet sampling is faster?

Denoising Diffusion Implicit Model (DDIM)

- Define the generalized posterior distribution

$$q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha_t}}}, \sigma_t^2 \boldsymbol{I}\right),$$

  where $\sigma_t^2 = \eta \frac{(1-\alpha_t)(1-\overline{\alpha_{t-1}})}{1-\overline{\alpha_t}} \geq 0$ is a variance parameter.

- The generalized posterior q_σ is designed such that it maintains the same forward distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ as in DDPM.

- Different choices of $\sigma_t \geq 0$ result in different generative models.
  - For $\eta = 1$, we obtain DDPM and the process is stochastic (slow)
  - For $\eta = 0, \forall t \geq 1$, the process is deterministic (fast)

- We will see that setting $\sigma_t = 0, \forall t \geq 1$, will allow for deterministic denoising and faster sampling!

DDIM Marginals $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ match DDPM's

$$q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_T; \sqrt{\overline{\alpha_T}}\boldsymbol{x}_0, (1-\overline{\alpha_T})\boldsymbol{I}\right)$$

$$q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha_t}}}, \sigma_t^2\boldsymbol{I}\right)$$

- Assume $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0, (1-\overline{\alpha_t})\boldsymbol{I}\right)$ (case $t = T$ holds by definition).
- By properties of Gaussians,

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0) = \int q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) d\boldsymbol{x}_t = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$$

$$\boldsymbol{\mu}_{t-1} = \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \frac{\sqrt{(1-\overline{\alpha_{t-1}}-\sigma_t^2)}}{\sqrt{1-\overline{\alpha_t}}}\left(\sqrt{\overline{\alpha_t}}\boldsymbol{x}_0 - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0\right) = \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0$$

$$\boldsymbol{\Sigma}_{t-1} = \left(\sigma_t^2 + \frac{\sqrt{(1-\overline{\alpha_{t-1}}-\sigma_t^2)}}{\sqrt{1-\overline{\alpha_t}}}(1-\overline{\alpha_t})\frac{\sqrt{(1-\overline{\alpha_{t-1}}-\sigma_t^2)}}{\sqrt{1-\overline{\alpha_t}}}\right)\boldsymbol{I}$$

$$= (\sigma_t^2 + 1 - \overline{\alpha_{t-1}} - \sigma_t^2)\boldsymbol{I} = (1-\overline{\alpha_{t-1}})\boldsymbol{I}$$

Therefore, $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0, (1-\overline{\alpha_{t-1}})\boldsymbol{I}\right)$.

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}_x)$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{\Sigma}_{y|x}\right)$$
$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{\Sigma}_{y|x} + \boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^\top\right)$$

DDIM Non-Markovian Process

$$q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha_t}}}, \sigma_t^2 \boldsymbol{I}\right),$$

- We can verify the chain $(\boldsymbol{x}_t)_{t=2}^T$ is Non-Markovian via Bayes' rule:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}$$

$$= \frac{\mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha_t}}}, \sigma_t^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0, (1-\overline{\alpha_t})\boldsymbol{I}\right)}{\mathcal{N}\left(\boldsymbol{x}_{t-1}; \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0, (1-\overline{\alpha_{t-1}})\boldsymbol{I}\right)}$$

$$= \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0 + \frac{\sqrt{(1-\overline{\alpha_t})(1-\overline{\alpha_{t-1}}-\sigma_t^2)}}{(1-\overline{\alpha_{t-1}})}\left(\boldsymbol{x}_{t-1} - \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0\right), \frac{(1-\overline{\alpha_t})}{(1-\overline{\alpha_{t-1}})}\sigma_t^2 \boldsymbol{I}\right)$$

- This would be Markovian if $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$. However, here we cannot get rid of $\boldsymbol{x}_0$, except when $\sigma_t^2 = \frac{(1-\alpha_t)(1-\overline{\alpha_{t-1}})}{1-\overline{\alpha_t}}$, which reduces to DDPM.

ELBO for the Generalized Process

- Recall that the score matching term of the ELBO reduces to:

$$D_{\mathrm{KL}}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right) = D_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0), \boldsymbol{\Sigma}_q) || \, \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_q)\right)$$

$$= \frac{1}{2\sigma_q^2(t)}\left[|| \, \boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q \, ||_2^2\right]$$

- Choosing mean of $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ to match form of mean of $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$
  - In DDPM we had:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha_{t-1}})\boldsymbol{x}_t + \sqrt{\overline{\alpha_{t-1}}}(1-\alpha_t)\boldsymbol{x}_0}{1-\overline{\alpha_t}} \quad \Rightarrow \quad \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha_{t-1}})\boldsymbol{x}_t + \sqrt{\overline{\alpha_{t-1}}}(1-\alpha_t)\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha_t}}$$

  - For the generalized process we have:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \sqrt{\overline{\alpha_{t-1}}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2}\,\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha_t}}} \quad \Rightarrow \quad \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \sqrt{\overline{\alpha_{t-1}}}\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t) + \sqrt{1-\overline{\alpha_{t-1}}-\sigma_t^2}\,\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha_t}}\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\overline{\alpha_t}}}$$

Reparameterization as an Alternative Form for ELBO

- Plugging $\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\bar{\boldsymbol{\epsilon}}_t}{\sqrt{\overline{\alpha}_t}}$ into the denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ gives:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \sqrt{\overline{\alpha}_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\,\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\overline{\alpha}_t}}$$

$$= \sqrt{\overline{\alpha}_{t-1}}\,\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\bar{\boldsymbol{\epsilon}}_t}{\sqrt{\overline{\alpha}_t}} + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\,\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\,\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\bar{\boldsymbol{\epsilon}}_t}{\sqrt{\overline{\alpha}_t}}}{\sqrt{1-\overline{\alpha}_t}}$$

$$= \sqrt{\overline{\alpha}_{t-1}}\,\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\bar{\boldsymbol{\epsilon}}_t}{\sqrt{\overline{\alpha}_t}} + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\,\bar{\boldsymbol{\epsilon}}_t$$

- Choosing the mean and variance of p to match the mean and variance of q:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \sqrt{\overline{\alpha}_{t-1}}\,\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\overline{\alpha}_t}} + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t), \qquad \boldsymbol{\Sigma}_\theta(t) = \sigma_t^2\boldsymbol{I}$$

What have we achieved so far?

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \sqrt{\overline{\alpha}_{t-1}} \frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\overline{\alpha}_t}} + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2} \cdot \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)$$

- We created a new generalized inference distribution with the same training objective as in DDPM.
- The generalized process captures a rich family of generative processes depending on the selection of the parameter σ_t.
- We can select σ_t to achieve much faster sampling!
- Recall that $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \sigma_t^2 \boldsymbol{I})$ and thus

$$\boldsymbol{x}_{t-1} = \sqrt{\overline{\alpha}_{t-1}} \underbrace{\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\overline{\alpha}_t}}}_{\text{predicted } \boldsymbol{x}_0} + \underbrace{\sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2} \cdot \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}_{\text{direction pointing to } \boldsymbol{x}_t} + \underbrace{\sigma_t \boldsymbol{\epsilon}_t}_{\text{Gaussian noise}} \quad , \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Denoising Diffusion Implicit Models (DDIM)

- DDIM uses $\sigma_t = 0, \forall t \geq 1$ in the generalized process.
- We can sample using the following equation, which gives **deterministic sampling**

$$\boldsymbol{x}_{t-1} = \sqrt{\overline{\alpha_{t-1}}} \underbrace{\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha_t}}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\overline{\alpha_t}}}}_{\text{predicted } x_0} + \underbrace{\sqrt{1-\overline{\alpha_{t-1}}} \cdot \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}_{\text{direction pointing to } x_t}$$

- Faster sampling: Since the reverse trajectory is deterministic, we can skip steps!
- Consider the full reverse trajectory $\boldsymbol{x}_T \rightarrow \boldsymbol{x}_{T-1} \rightarrow \cdots \rightarrow \boldsymbol{x}_0$.
- DDIM uses a decreasing sequence $\{\tau_1, \ldots, \tau_{S-2}\}$ to subsample a trajectory of length S

$$\boldsymbol{x}_T \rightarrow \boldsymbol{x}_{\tau_1} \rightarrow \cdots \rightarrow \boldsymbol{x}_{\tau_{S-2}} \rightarrow \boldsymbol{x}_0$$

- In practice, $S \ll T$ and in this way we can obtain faster sampling!

Denoising Diffusion Implicit Models (DDIM)

- DDIM uses $\sigma_t = 0, \forall t \geq 1$ in the generalized process.
- We can sample using the following equation, which gives **deterministic sampling**

$$\boldsymbol{x}_{t-1} = \sqrt{\overline{\alpha}_{t-1}} \underbrace{\frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\overline{\alpha}_t}}}_{\text{predicted } x_0} + \underbrace{\sqrt{1-\overline{\alpha}_{t-1}} \cdot \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}_{\text{direction pointing to } x_t}$$

- Faster sampling: Since the reverse trajectory is deterministic, we can skip steps!
  - **Predict the clean image $\hat{\boldsymbol{x}}_0$:** Given current noisy state $\boldsymbol{x}_t$ and the network's predicted noise $\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)$, project all the way back to estimate the original clean data point:

$$\hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

  - **Directionally re-corrupt to $\boldsymbol{x}_{t-1}$:** Now that you have a guess for $\hat{x}_0$ and the direction pointing toward it, step forward to any arbitrary target step $t-k$ along a trajectory:

$$\boldsymbol{x}_{t-k} = \sqrt{\bar{\alpha}_{t-k}}\hat{\boldsymbol{x}}_0 + \sqrt{1-\bar{\alpha}_{t-k}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)$$

Sample Efficiency of DDIM

- DDIM with only $S = 10$ steps of reverse process achieves better FID score than DDPM with 1000 steps in the reverse process.

η: noise added at each step of the reverse process

		CIFAR10 (32 × 32)					CelebA (64 × 64)				
	S	10	20	50	100	1000	10	20	50	100	1000
DDIM	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
η	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
DDPM	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98

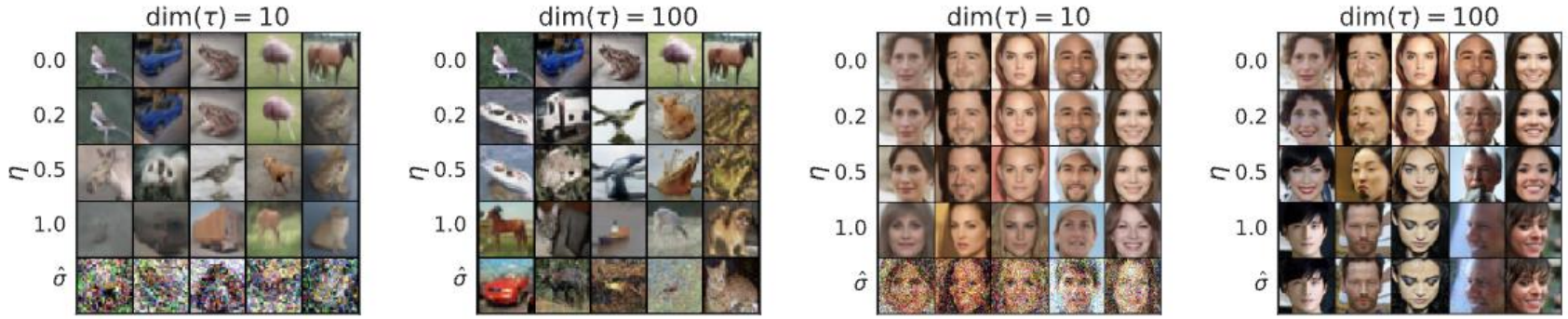


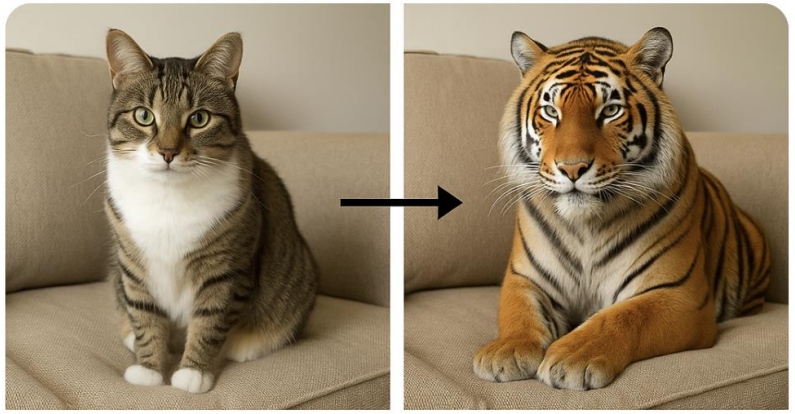
Figure 3: CIFAR10 and CelebA samples with $\dim(\tau) = 10$ and $\dim(\tau) = 100$.

What have we achieved so far?

- For image editing, we required exact inversion of the diffusion model and fast sampling.
- DDIM with $\sigma_t = 0$ provides deterministic sampling in a few steps.

- To perform image editing with DDIM:
  1. **Encode**: Run the forward process to get $\boldsymbol{x}_t$ for some intermediate t (partial noising of $\boldsymbol{x}_0$).
  2. **Edit**: Modify the conditioning input.
  3. **Decode**: Run the reverse DDIM process using the new conditioning to get the modified image.

Input: "A photo of a **cat** on a couch"

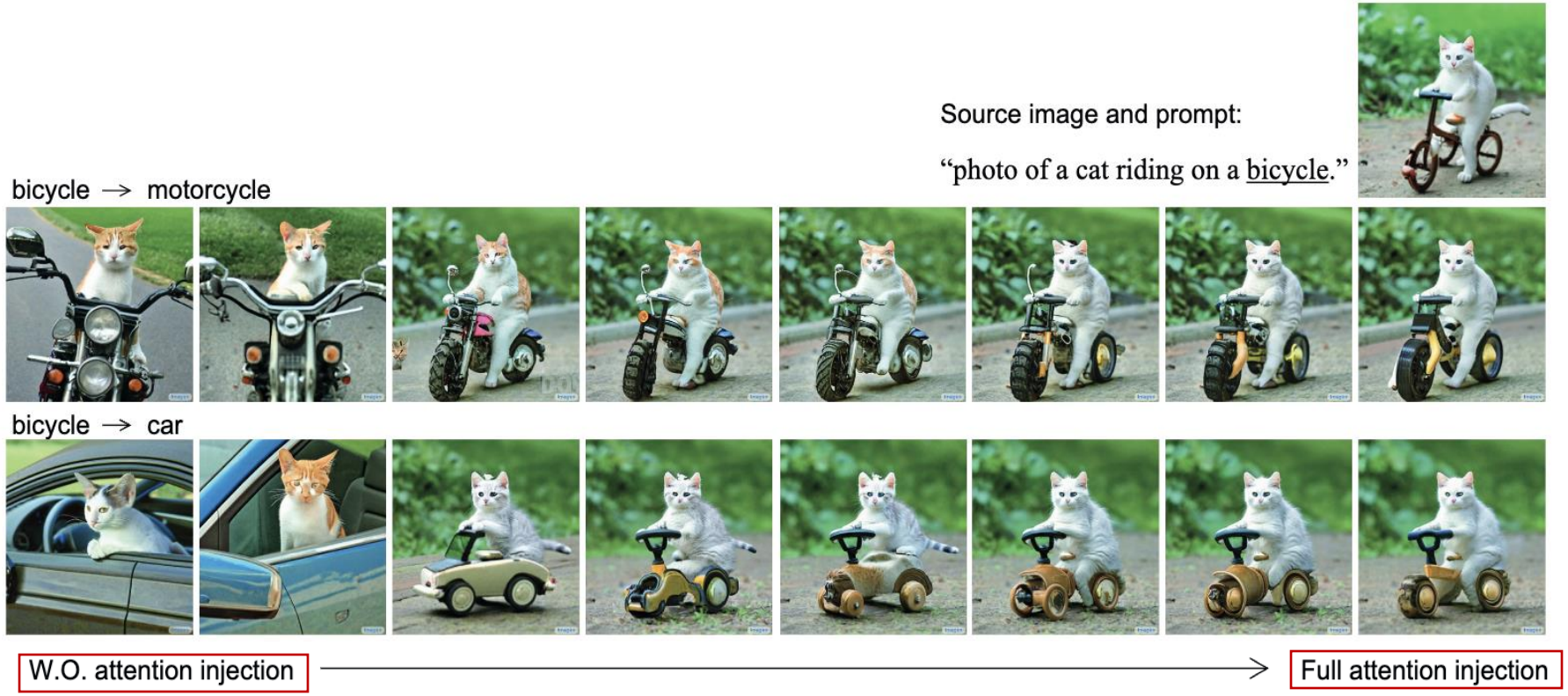


Edit: "A photo of a **tiger** on a couch"

- Next, we will see how to perform even more advanced edits in this space.
  - One example: Prompt2Prompt (P2P)

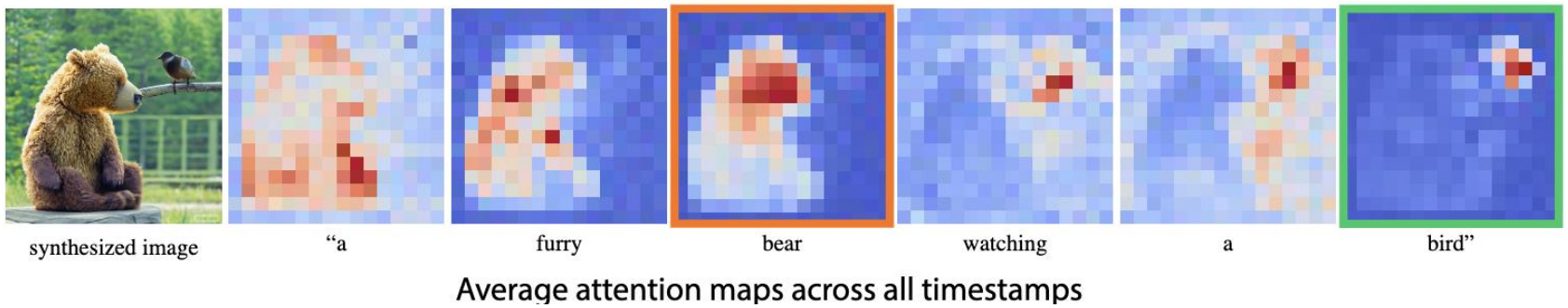
Prompt2Prompt

- DDIM Inversion has no symbolic (rigid) control for structural consistency between the initial and edited images.
- Prompt2Prompt (P2P): when generating an image based on a prompt, save the cross-attention maps during the forward process, change a word of the prompt but use the attention maps of the original word during the reverse process.

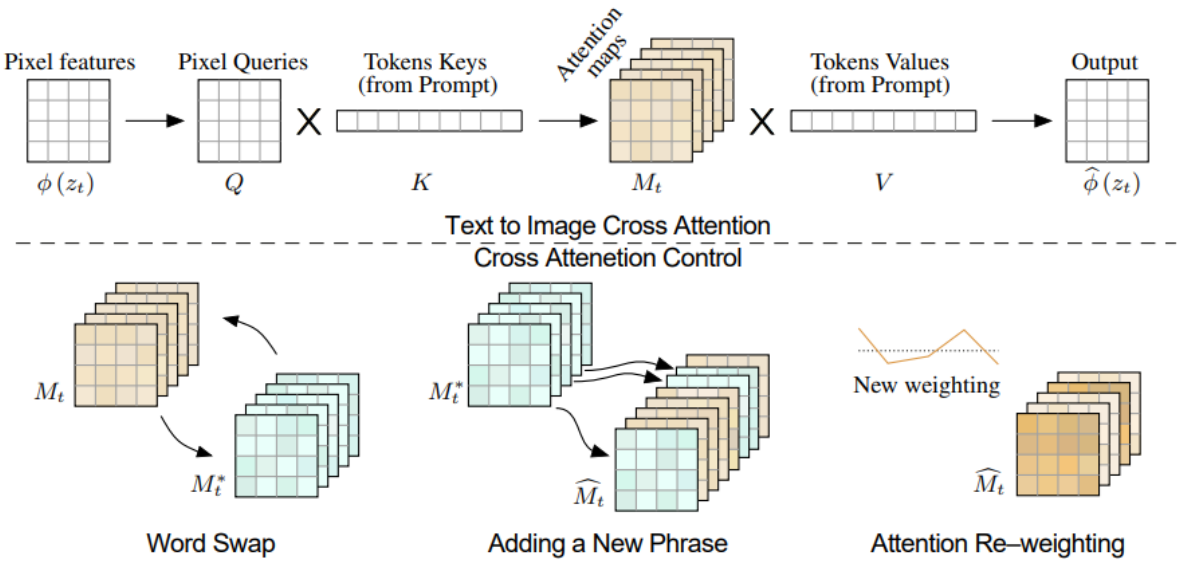


Prompt2Prompt

- The spatial layout of the generated image depends on the cross-attention maps, each indicating how each spatial location in the image attends to each word in the text.



- To perform image editing with P2P:
  1. **Save** the initial attention maps of the forward DDIM process.
  2. **Edit:** Compute the attention maps corresponding to the edit prompt.
  3. **Decode:** Inject the edited attention maps and run the reverse process.



How to perform cross-attention replacement?

- At each denoising step t and cross-attention layer ℓ, the UNet produces an attention map $\boldsymbol{M}^{(\ell,t)} \in \mathbb{R}^{(H \times W) \times K}$, where K is the number of tokens.

Example

- Source Prompt P = "a **cat** on a couch" → Edit Prompt P' = "a **tiger** on a couch"
- **Lock/shared tokens**: keep their spatial maps from the source
- **Swap tokens**: copy the old token's map into the new token's column.
- For $i_{\text{cat}} \in P, j_{\text{tiger}} \in P'$ and S being the set of shared tokens, the attention map $\boldsymbol{M'}^{(\ell,t)}$ corresponding to the edit prompt P' is given

$$\boldsymbol{M'}^{(\ell,t)}[:, j] \leftarrow \boldsymbol{M}^{(\ell,t)}[:, i], \text{ for } (i, j) \in S \text{ (copy)}$$

$$\boldsymbol{M'}^{(\ell,t)}[:, j_{\text{tiger}}] \leftarrow \boldsymbol{M}^{(\ell,t)}[:, i_{\text{cat}}] \text{ (swap)}$$

Samples of Edited Images

"Photo of a cat riding on a bicycle."



source image



cat → dog

"A photo of a butterfly on a flower."



source image



"...on a spikey flower."

"Photo of a house with a flag on a mountain."



source image



house → hotel



house → tent



house → car



house → tree

Conclusion on Image Editing

- The latent space of diffusion models can be used for image editing.
- Image editing using DDPM faces the problems of inversion and slow sampling.
- To speed up the sampling process, we considered a generalized non-Markovian forward process.
- DDIM provides deterministic reverse process and fast sampling.
- Prompt2Prompt allows for edits in the image, while maintaining the structural properties of the initial image.

