Deep Generative Models: Image Editing with Diffusion Models

Fall Semester 2025

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

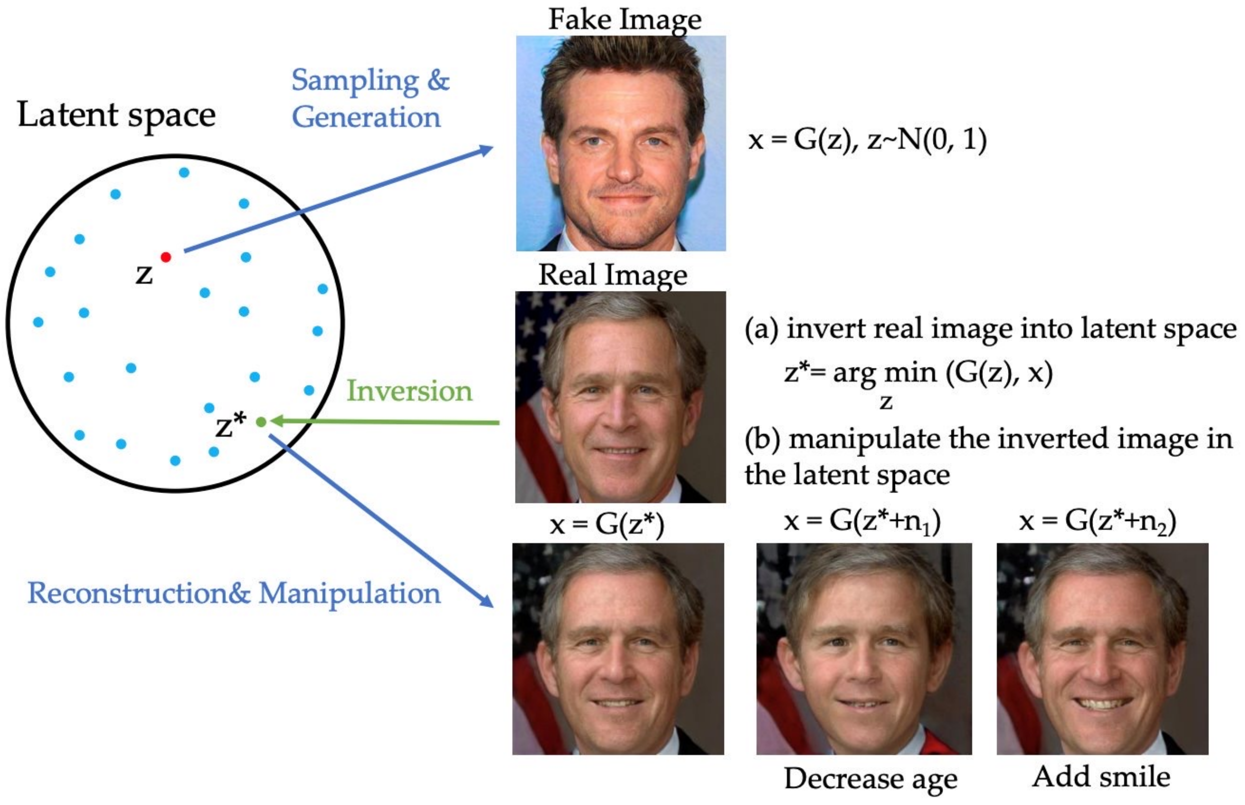
- Markov Hierarchical Variational Auto Encoders (MHVAEs)
  - Autoregressive Encoder and Autoregressive Decoder of an MHVAE
  - Derivation of the ELBO of an MHVAE

- Diffusion Models as 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

- Applications of Diffusion Models
  - Stable Diffusion: Text-Conditioned Diffusion Model
  - ControlNet: Multimodal Control for Consistent Synthesis
  - **Image Editing: DDIM, P2P**

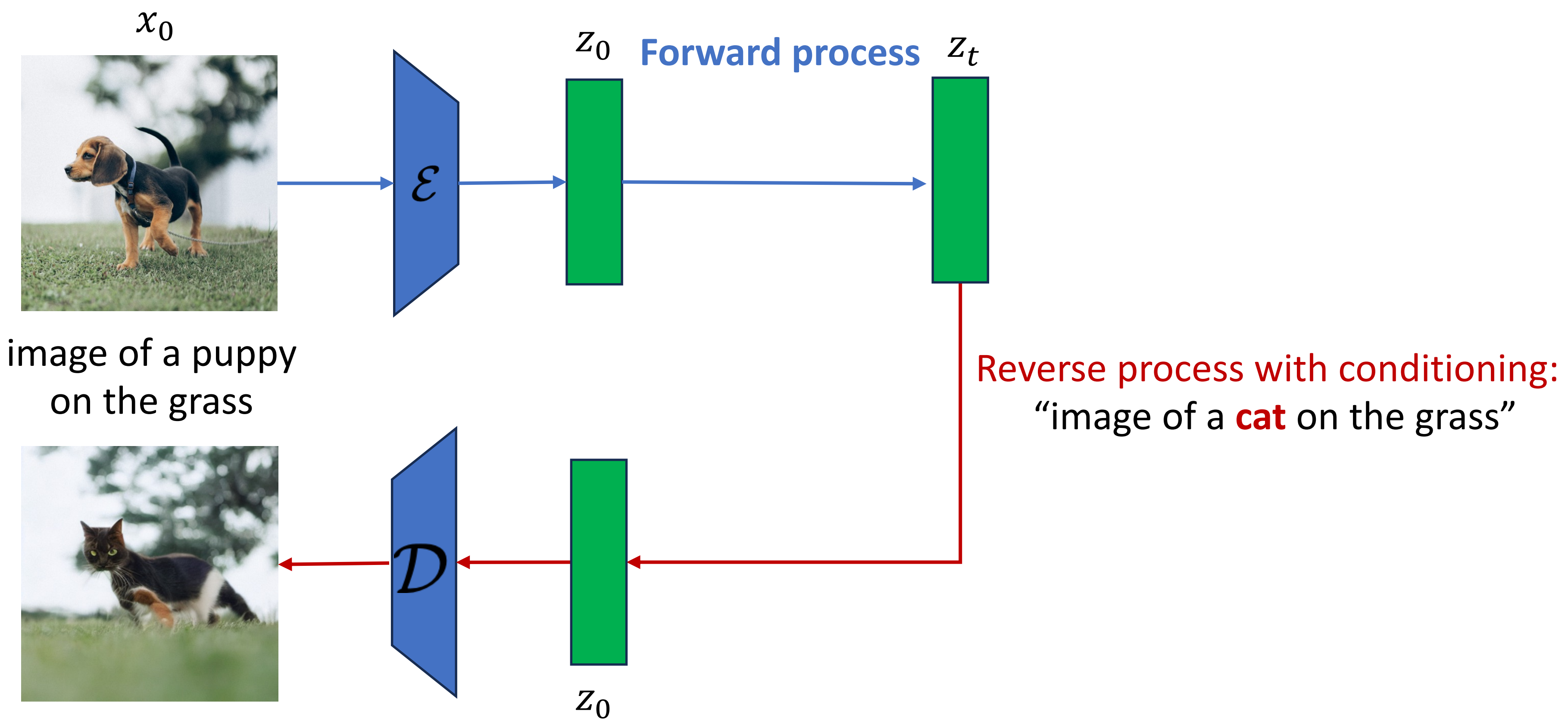
Latent Space Image Editing: Inversion + Manipulation

- 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

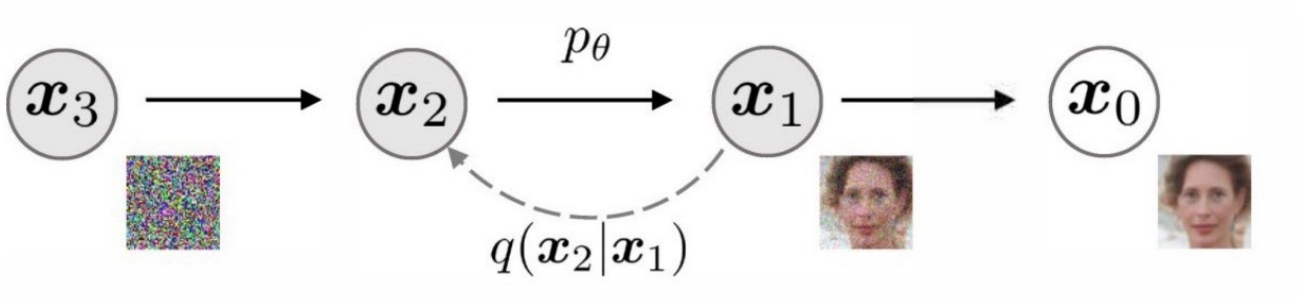
- Instead of starting from pure noise, let us perform naïve inversion using the forward process and a fixed image.



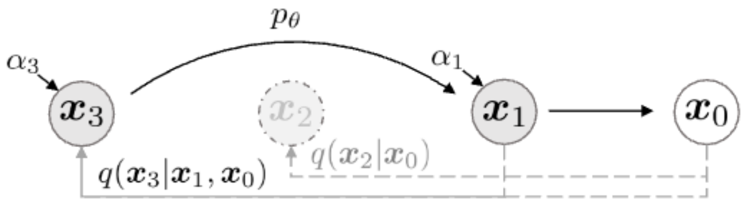
- Depending on how much noise is added, we can change a lot of features in the image or not enough features.

How to improve the inversion?

- Problems
  - Randomness in the model: if we encode x_0 to x_t using the forward process and then run the reverse process, we will not get x_0.
  - The reverse process requires T sequential steps, which can be slow.



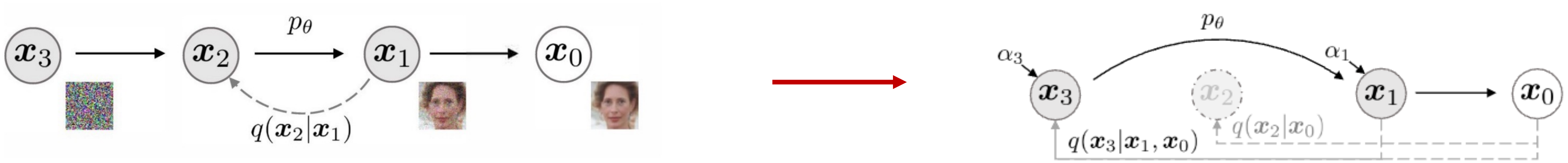
- What if we had a different sampling mechanism?



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

Designing Faster Processes

- In diffusion models, the reverse process is designed to approximate the forward process.
- Intuition: If we had a forward process with few steps, the backward process would also require a small number of steps to sample a new image.
- How can we design sampling processes with less number of 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!

Generalized Non-Markovian Processes

- Define the generalized posterior distribution

$$q_\sigma(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(\sqrt{\overline{\alpha_{t-1}}}x_0 + \sqrt{1 - \overline{\alpha_{t-1}} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha_t}}x_0}{\sqrt{1-\overline{\alpha_t}}}, \sigma_t^2 I\right),$$

  where $\sigma_t \geq 0$ is a variance parameter.
- The generalized posterior q_σ is designed such that it maintains the same forward distribution $q(x_t|x_0)$ as in DDPM.
- Different choices of $\sigma_t \geq 0$ result in different generative models.
  - For $\sigma_t = \sqrt{\beta_t}$, we obtain DDPM.
  - For $\sigma_t = 0, \forall t \geq 0$, the process is deterministic!
- We will see that setting $\sigma_t = 0, \forall t \geq 0$, will allow for deterministic denoising and faster sampling!

$\sigma_q^2(t) = \sigma_t^2$

ELBO for the Generalized Process

- Recall our ELBO derivation

$$\log p(x) \geq \underbrace{\mathbb{E}_{q_\phi(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(x_T \mid x_0) \,||\, p_\theta(x_T)\right)}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q_\phi(x_t|x_0)}\left[D_{\mathrm{KL}}\left(q_\phi(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right)\right]}_{\text{score matching term}}$$

- The KL divergence for Gaussians

$$\mathrm{D_{KL}}\left(\mathcal{N}(x; \mu_x, \Sigma_x) \,||\, \mathcal{N}(y; \mu_y, \Sigma_y)\right) = \frac{1}{2}\left[\cancel{\log \frac{|\Sigma_y|}{|\Sigma_x|}} - \mathrm{d} + \cancel{\mathrm{tr}(\Sigma_y^{-1}\Sigma_x)} + (\mu_y - \mu_x)^{\mathrm{T}} \Sigma_y^{-1} (\mu_y - \mu_x)\right]$$

- Choosing mean of $p_\theta(x_{t-1} \mid x_t)$ to match form of mean of $q(x_{t-1} \mid x_t, x_0)$

$$\mu_q(x_t, x_0) = \sqrt{\overline{\alpha}_{t-1}}\, x_0 + \sqrt{1 - \overline{\alpha}_{t-1} - \sigma_t^2}\, \frac{x_t - \sqrt{\overline{\alpha}_t}\, x_0}{\sqrt{1 - \overline{\alpha}_t}}, \qquad \mu_\theta(x_t, t) = \sqrt{\overline{\alpha}_{t-1}}\, \widehat{x_\theta}(x_t, t) + \sqrt{1 - \overline{\alpha}_{t-1} - \sigma_t^2}\, \frac{x_t - \sqrt{\overline{\alpha}_t}\, \widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha}_t}}$$

- The ELBO reduces to:

$$D_{\mathrm{KL}}\left(q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right) = D_{\mathrm{KL}}\left(\mathcal{N}\left(x_{t-1}; \mu_q(x_t, x_0), \Sigma_q\right) || \,\mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_q\right)\right)$$

$$= \frac{1}{2\sigma_q^2(t)}\left[||\, \mu_\theta - \mu_q \,||_2^2\right]$$

What have we achieved so far?

$$\mu_\theta(x_t, t) = \sqrt{\overline{\alpha_{t-1}}}\widehat{x_\theta}(x_t, t) + \sqrt{1 - \overline{\alpha_{t-1}} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha_t}}\widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha_t}}}$$

- 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 $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$ and thus

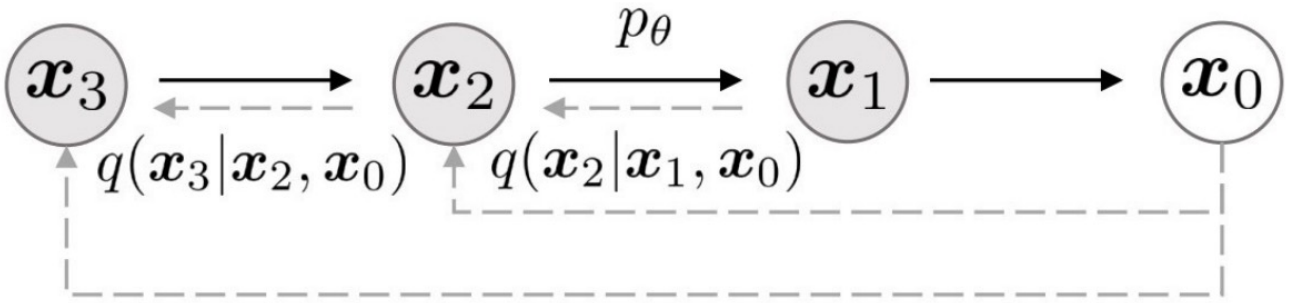
$$x_{t-1} = \underbrace{\sqrt{\overline{\alpha_{t-1}}}\widehat{x_\theta}(x_t, t)}_{\text{predicted } x_0} + \underbrace{\sqrt{1 - \overline{\alpha_{t-1}} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha_t}}\widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha_t}}}}_{\text{direction pointing to } x_t} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \quad , \epsilon_t \sim \mathcal{N}(0, I)$$

Denoising Diffusion Implicit Models (DDIM)

- DDIM uses $\sigma_t = 0, \forall t \geq 0$ in the generalized process.
- We can sample using the equation

$$x_{t-1} = \underbrace{\sqrt{\overline{\alpha}_{t-1}}\widehat{x_\theta}(x_t, t)}_{\text{predicted } x_0} + \underbrace{\sqrt{1 - \overline{\alpha}_{t-1}}\frac{x_t - \sqrt{\overline{\alpha}_t}\widehat{x_\theta}(x_t, t)}{\sqrt{1-\overline{\alpha}_t}}}_{\text{direction pointing to } x_t}$$

- This gives us deterministic sampling.
- Faster sampling: Consider the forward process $x_{1:T}$ of DDPM. DDIM uses a subset $\{\tau_1, \dots, \tau_s\}$ of length S of the whole DDPM process and inverses that process.
- In practice, s $\ll T$ and in this way we can obtain faster sampling!



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

		S	CIFAR10 (32 × 32) 10	20	50	100	1000	CelebA (64 × 64) 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
	η	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
		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

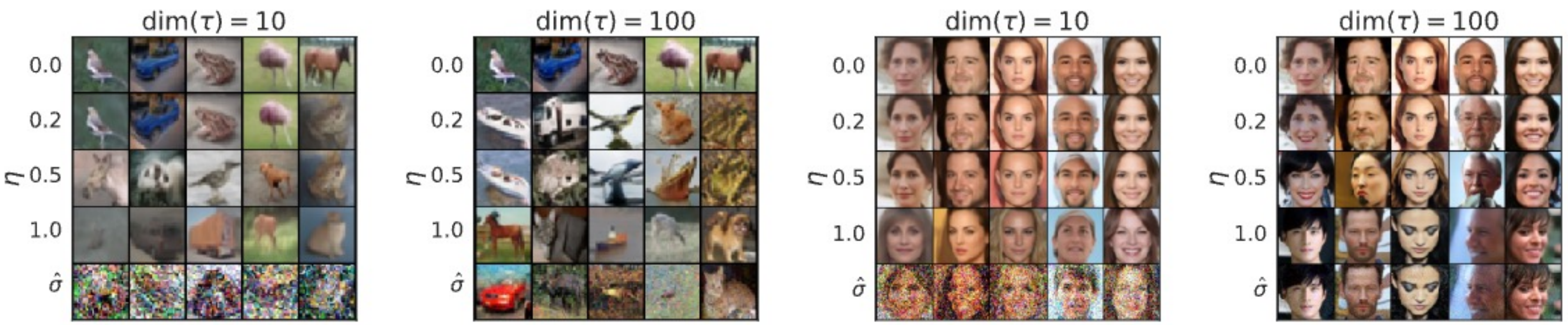
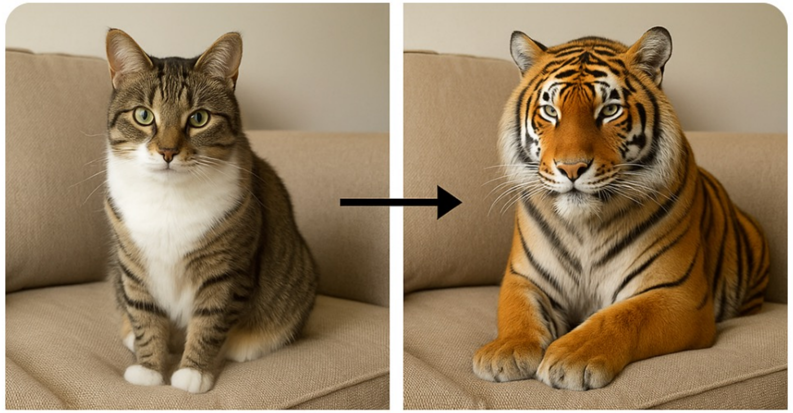


Figure 3: CIFAR10 and CelebA samples with $\dim(\tau) = 10$ and $\dim(\tau) = 100$.

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 x_t for some intermediate t (partial noising of 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"

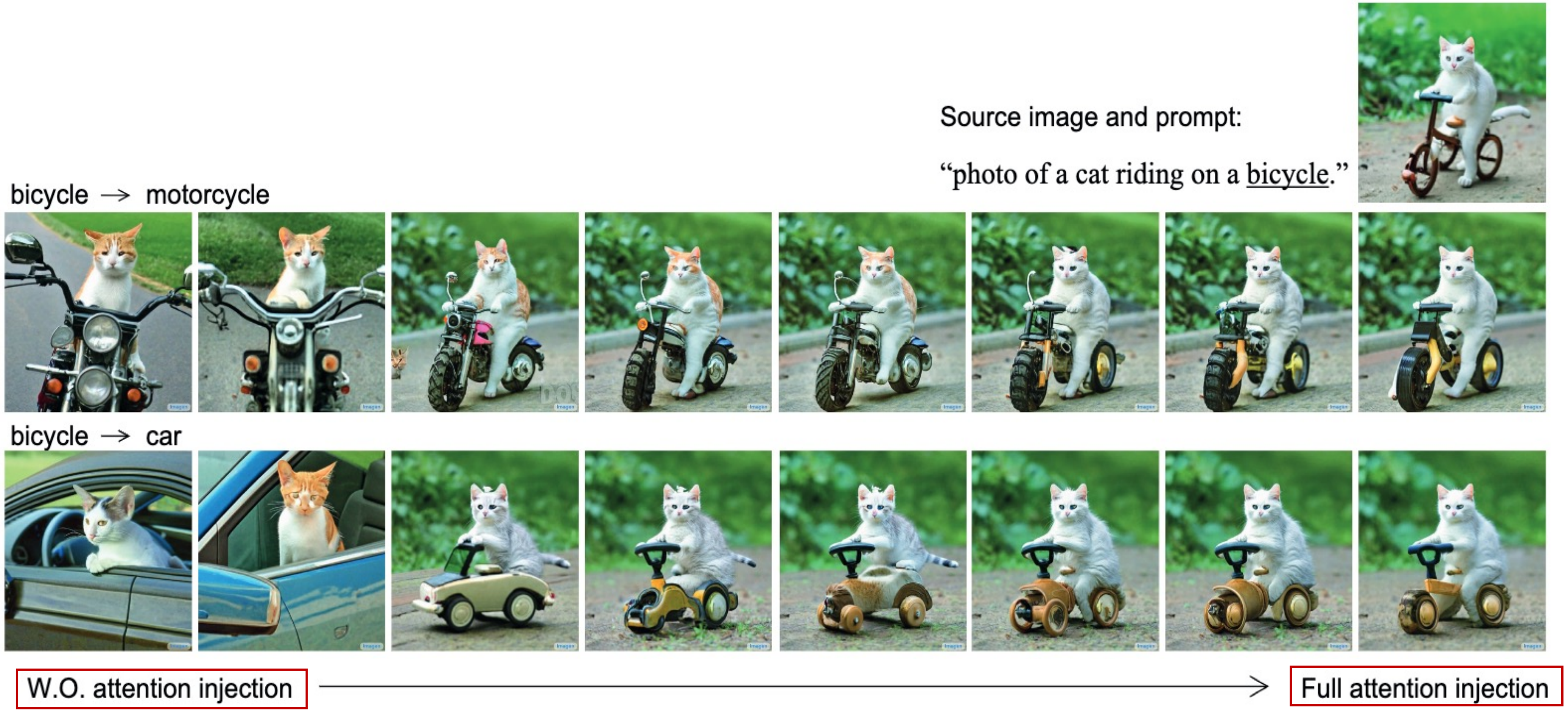


Edit: "A photo of a **tiger** on a couch"

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

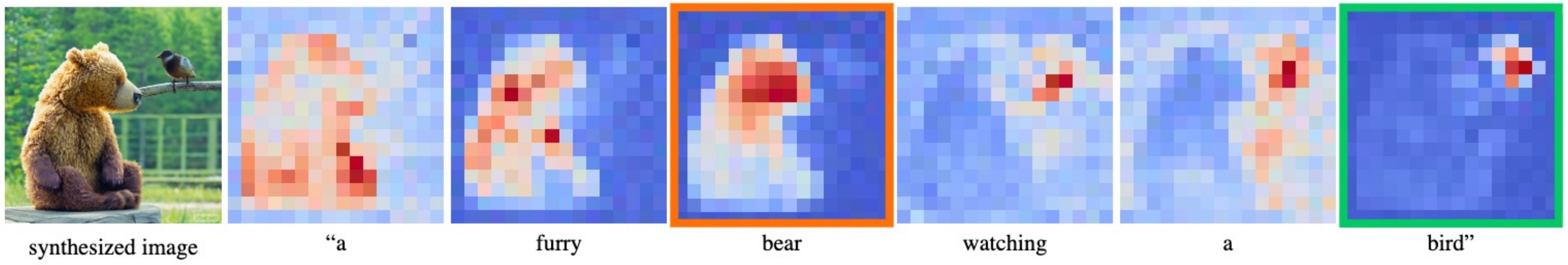
Prompt2Prompt

- DDIM Inversion has no symbolic (rigid) control for structural consistency.
- Prompt2Prompt (P2P) proposes to save the cross-attention maps during the forward process, edit the image and reuse the same attention maps during the reverse process.



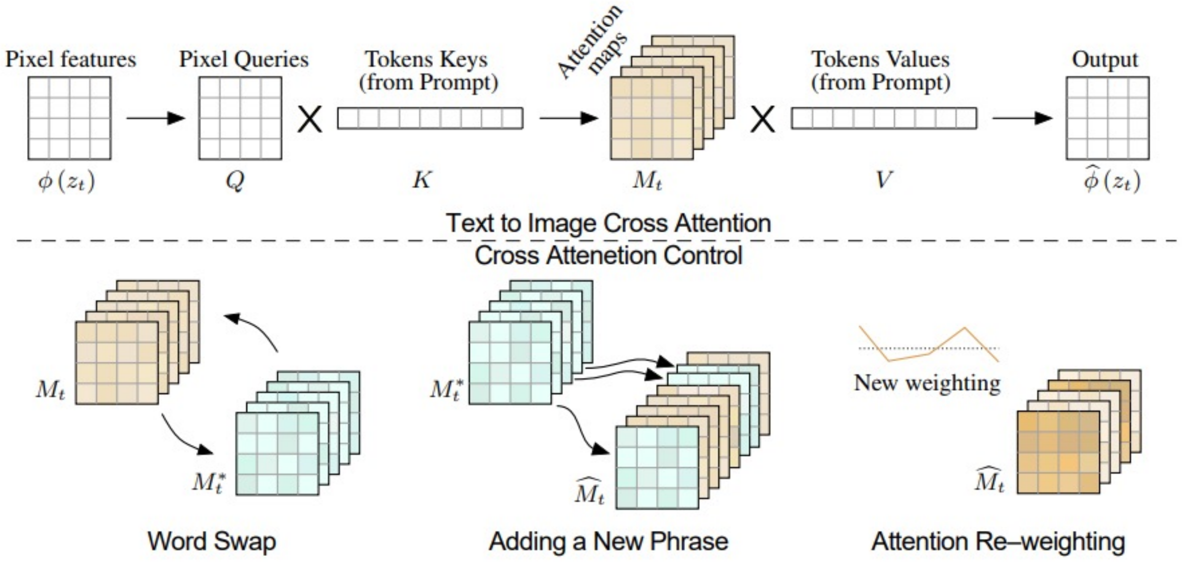
Prompt2Prompt

- The spatial layout and geometry of the generated image depends on the cross-attention maps.



Average attention maps across all timestamps

- To edit an image with P2P:
  1. Run the forward DDIM process to save the attention maps of the initial image.
  2. Edit: Compute the attention maps corresponding to the edit prompt.
  3. Decode: Inject the edited attention maps to the reverse process and get edited image.



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.