Deep Generative Models: Image Editing with Diffusion Models

Fall Semester 2024

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

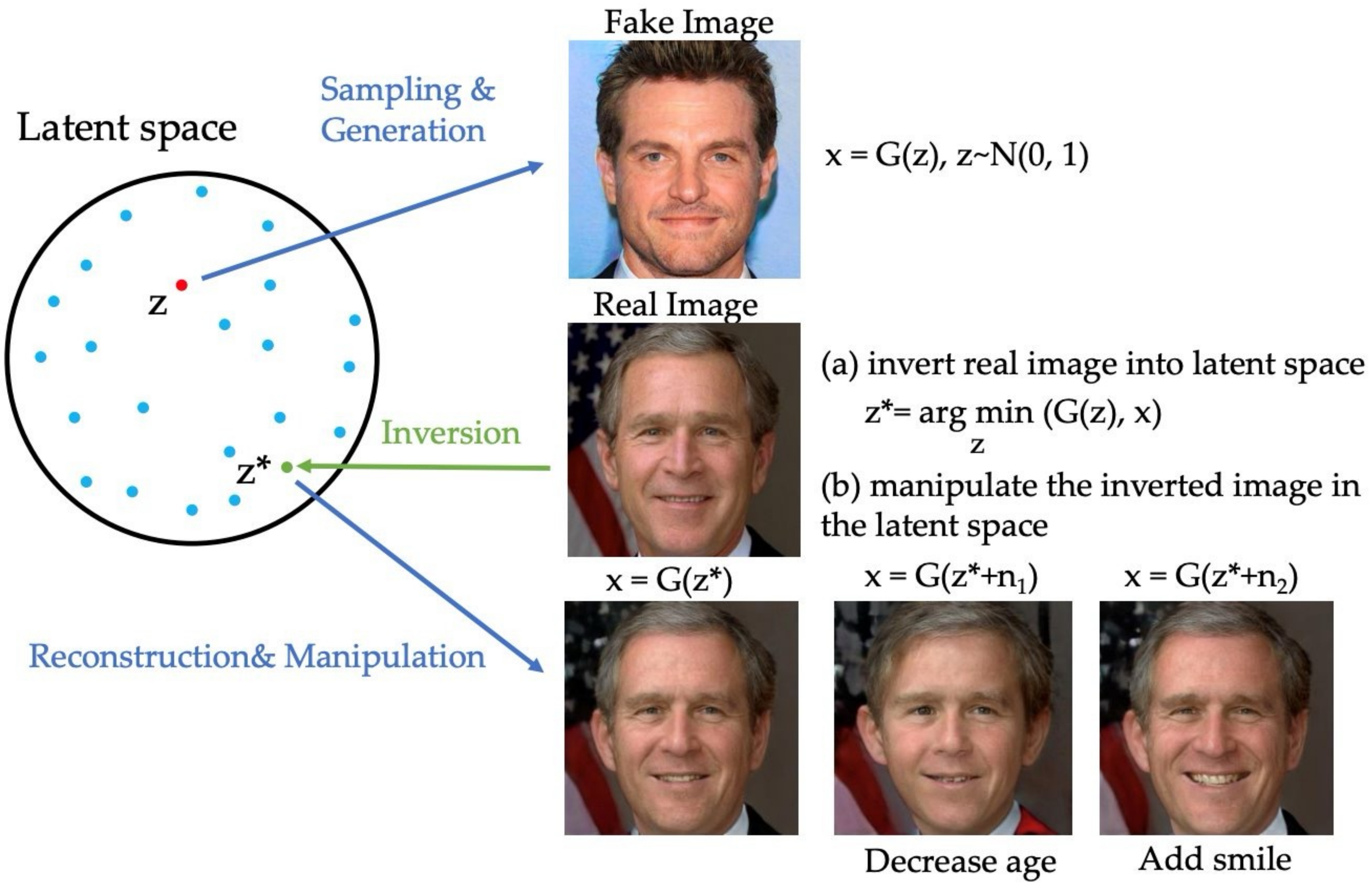
Penn
UNIVERSITY of PENNSYLVANIA

Diffusion Models

- **Derivation of Diffusion Models + Stable Diffusion/Control net (Last Lecture)**
  - Markov Hierarchical Variational Auto Encoders (MHVAE)
  - Diffusion Models are VAEs with Linear Gaussian Autoregressive latent space
  - ELBO for Diffusion Models is a particular case of ELBO for VAEs with extra structure
  - Implementation Details
  - Latent Diffusion Models (Stable Diffusion) + Controllable generation

- **Image Editing with Diffusion Models (Today's Lecture)**
  - DDIM, P2P, Overview of other baselines from project

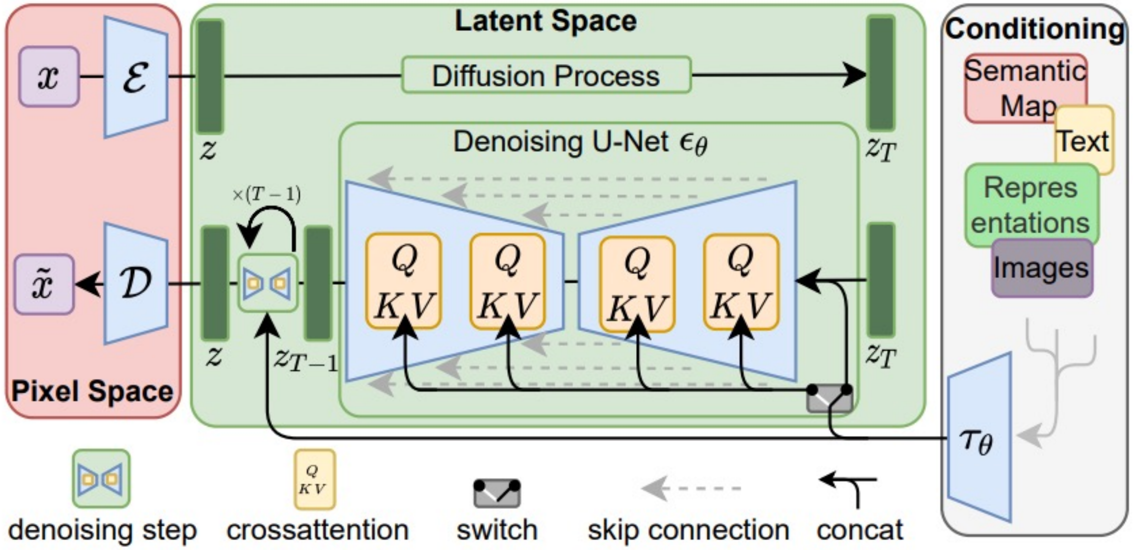
Latent Space Image Editing: Inversion + Manipulation



We learned that diffusion models are hierarchical VAEs so can we use their “latent space” to do editing?

Text-to-Image Diffusion Models

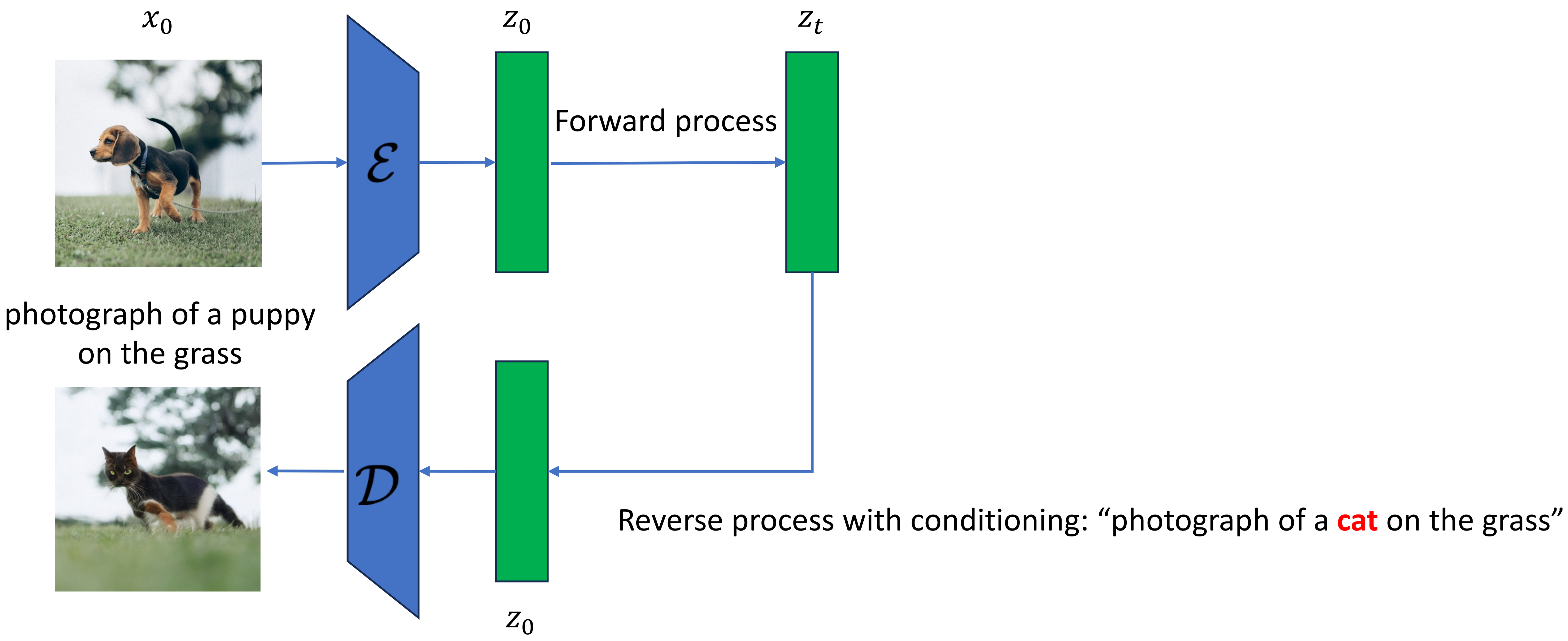
- Last class, we learned that stable diffusion can perform conditional generation using a text prompt



Text prompt: “photograph of a puppy on the grass”

Naïve Image Editing Idea

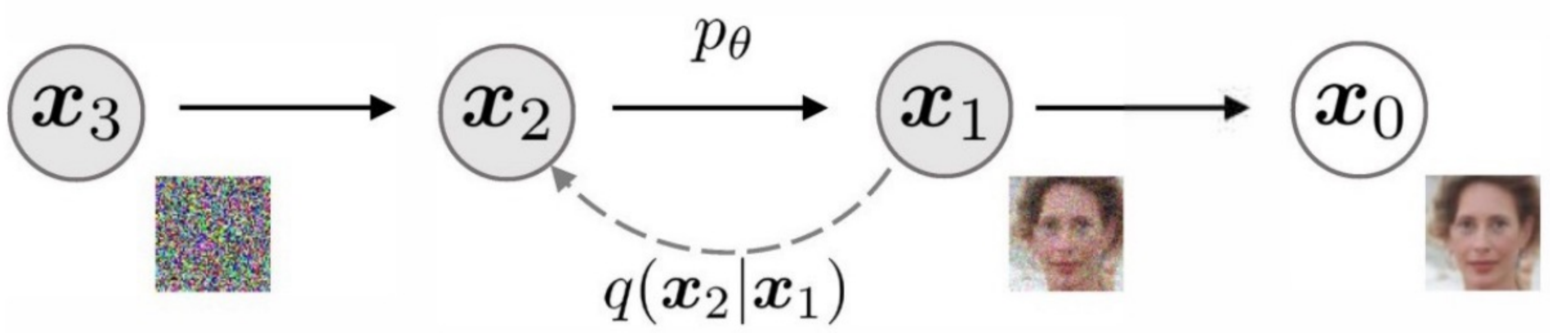
- Instead of starting from pure noise, let us perform naïve inversion using the forward process and a fixed image



Depending on how much noise we add, we can change a lot of features in the image or not enough features

Better Inversion?

- Problem: There is a lot of randomness in the diffusion model



- What if we had a different sampling mechanism?
  - In the next couple slides, we will derive a different sampling mechanism for pixel-space diffusion models (DDIM) that will allow us to achieve better inversion as a result

Denoising Diffusion Implicit Models (DDIM)

- Recall our ELBO derivation

$$\log p(x)$$

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

- Previously: Compute $q_\phi(x_{t-1}|x_t, x_0)$ by Bayes rule + forward process $q(x_t|x_0) = N(\sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I)$
- New idea: Define inference distribution as

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

- Marginal $q(x_t|x_0)$ gives same forward process as DDPM
- Note that when $\sigma_t = 0$ for all t, the process is deterministic!
  - Hint: Inversion will be easier!

Learning Objective

- Recall KL divergence for Gaussians

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

- Choose variance of p to match exactly variance of q

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

$$D_{\mathrm{KL}}(q(x_{t-1}\mid x_t,x_0)|p_\theta(x_{t-1}\mid x_t))$$
$$=D_{\mathrm{KL}}\left(\mathcal{N}\left(x_{t-1};\mu_q,\Sigma_q(t)\right)|\mathcal{N}\left(x_{t-1};\mu_\theta,\Sigma_q(t)\right)\right)$$
$$=\frac{1}{2\sigma_q^2(t)}\left[\|\mu_\theta-\mu_q\|_2^2\right]$$

This is going to be same as DDPM!

- Choose mean of p to match form of mean of q

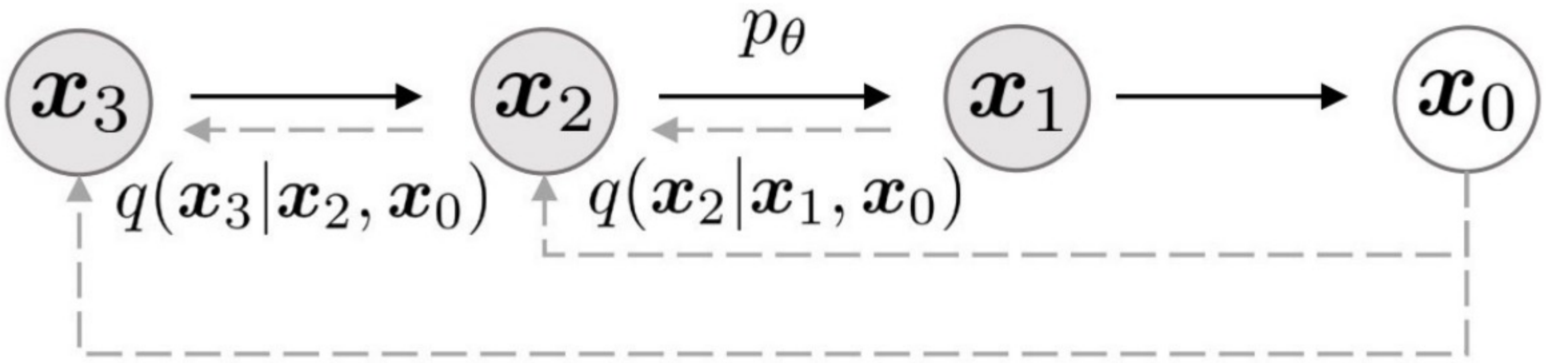
$$\mu_q(x_t,x_0)=\sqrt{\overline{\alpha}_{t-1}}x_0+\sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\,\frac{x_t-\sqrt{\overline{\alpha}_t}x_0}{\sqrt{1-\overline{\alpha}_t}}$$

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

What have we done?

- We created a new inference distribution such that the training objective is same as DDPM
    - This should make sense because the marginal $q(x_t|x_0)$ was same as DDPM forward process and that is all the training objective depended on
- But we introduced this parameter σ_t !
    - One application: Much faster sampling

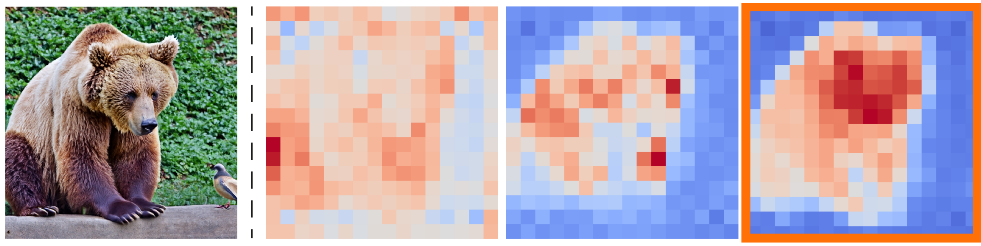
$$x_{t-1} = \underbrace{\sqrt{\overline{\alpha_{t-1}}}\,\widehat{x_\theta}(x_t, t)}_{\text{Predicted } x_0} + \underbrace{\sqrt{1 - \overline{\alpha_{t-1}} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha_t}}\,\widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha_t}}}}_{\text{Direction pointing to } x_t} + \underbrace{\sigma_t \epsilon}_{\text{Random noise}}$$



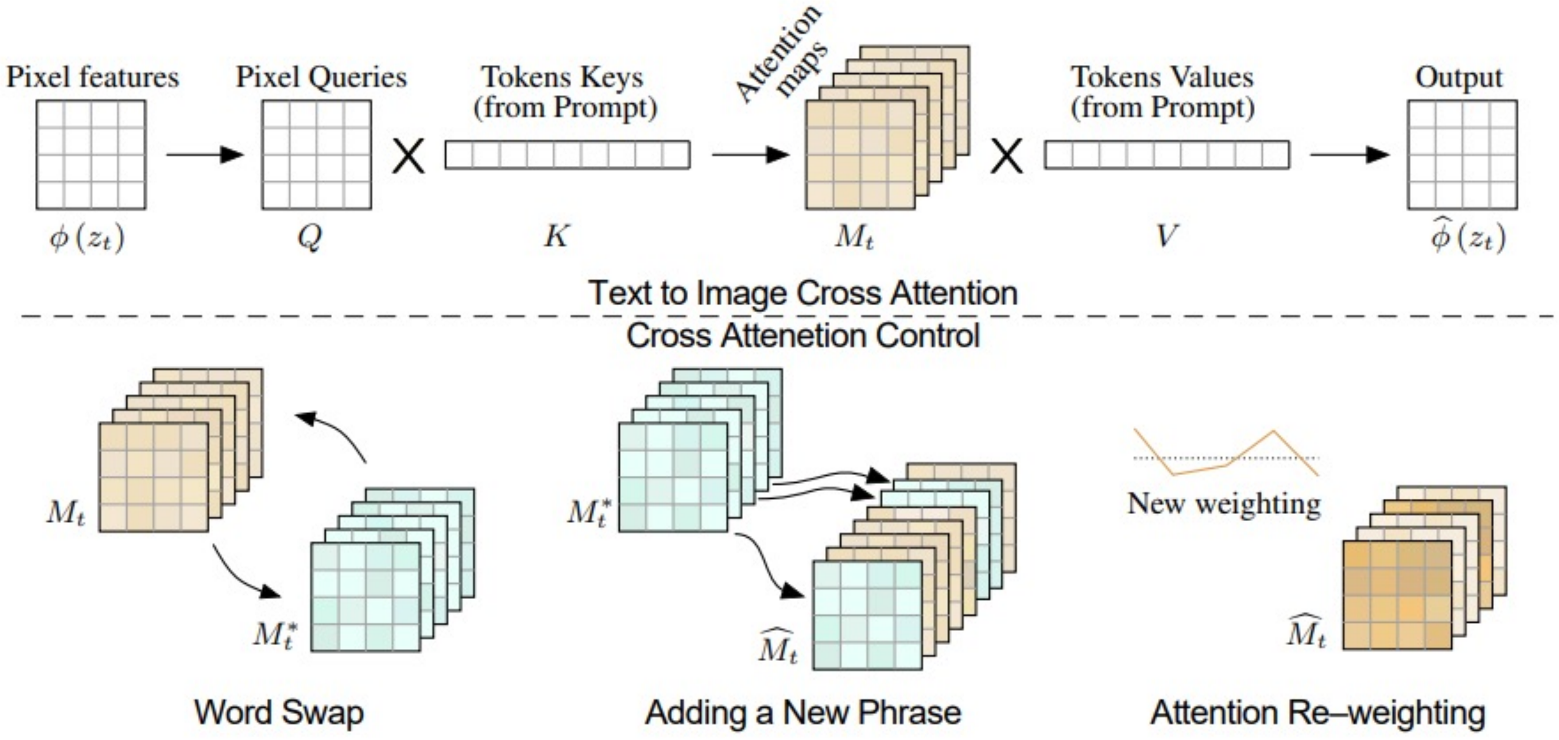
DDIM Inversion

- Finally, we can come back to what we started off with: image editing for which we wanted ”inversion” of diffusion model
- DDIM with $\sigma_t = 0$ gives us deterministic sampling (i.e. given x_T, DDIM sampling is fixed)
- This is useful for inversion
  - Take x_0 and compute the forward process using $\sigma_t = 0$ and some sample of x_T. This computed x_t is the ”inversion” of x_0 into the latent space of the diffusion model
- Next, we will see how to perform edits in this space
  - One example: Prompt2Prompt (P2P)

Prompt2Prompt



- Attention Control: DDIM Inversion has no symbolic (rigid) control for structural consistency! Authors proposed to save the cross-attention maps during DDIM Forward and re-use (inject) them during reverse process.



“photo of a cat riding on a bicycle.”

bicycle -> motorcycle

bicycle -> car

bicycle -> airplane

bicycle -> train

Project Overview

- Some baseline methods in your project improve upon inversion or editing
  - *DDIM Inversion, better editing:* Direct Inversion, Null Text Inversion, Pix2Pix Zero
  - *DDPM Inversion*: Edit-Friendly P2P
  - *Naïve Inversion, latent space editing*: Blended Latent Diffusion, MasaCtrl
- Other methods just train conditional diffusion models on large datasets to perform editing
  - Instruct Pix2Pix, InstructDiffusion, StyleDiffusion