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

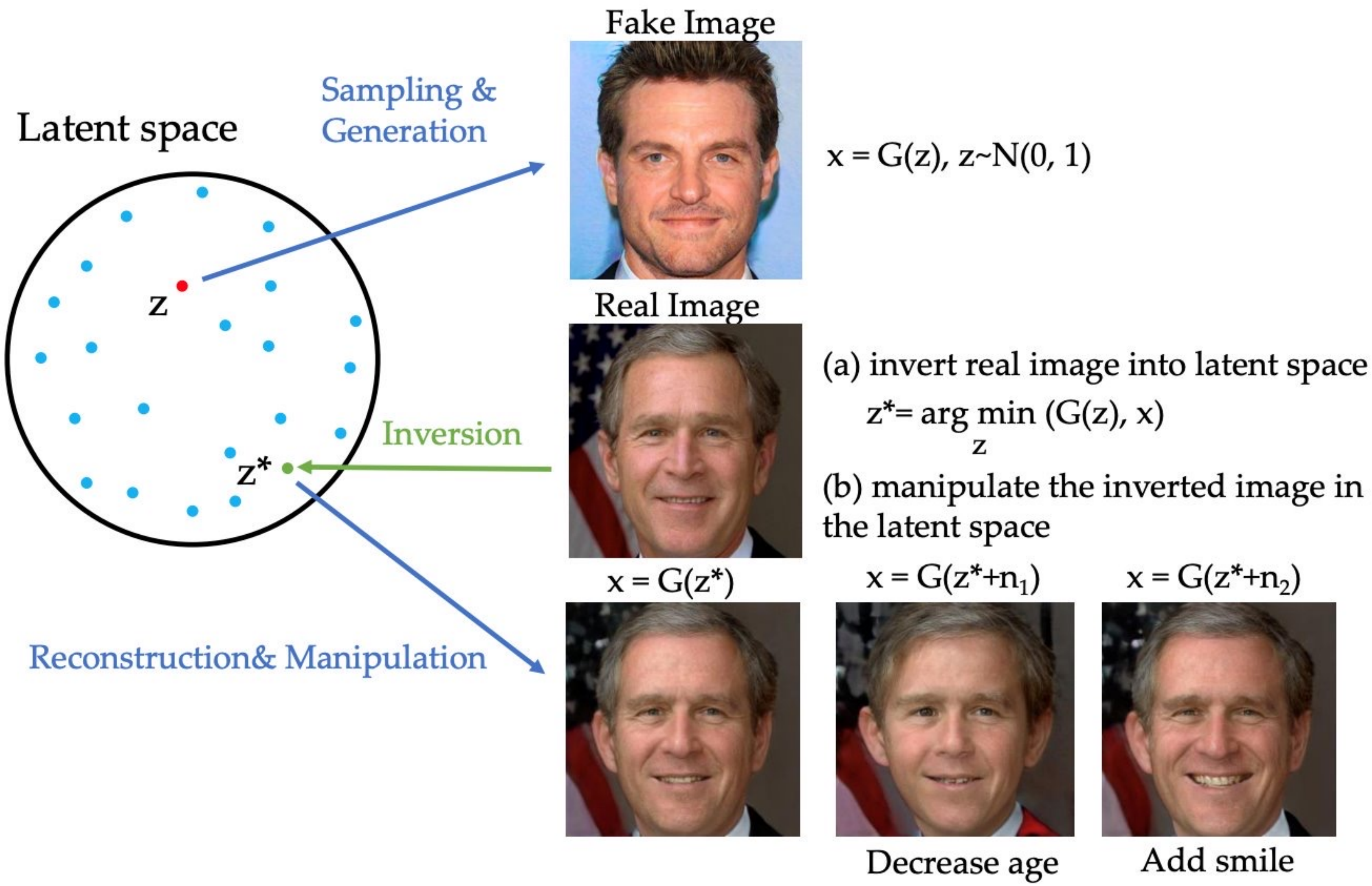
Outline

- 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
  - **DDIM, P2P**

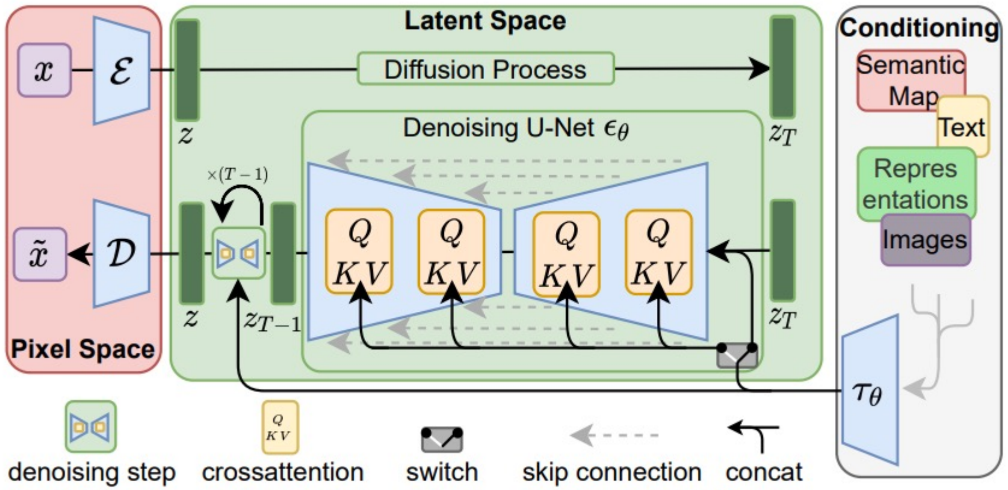
Latent Space Image Editing: Inversion + Manipulation



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

Text-to-Image Diffusion Models

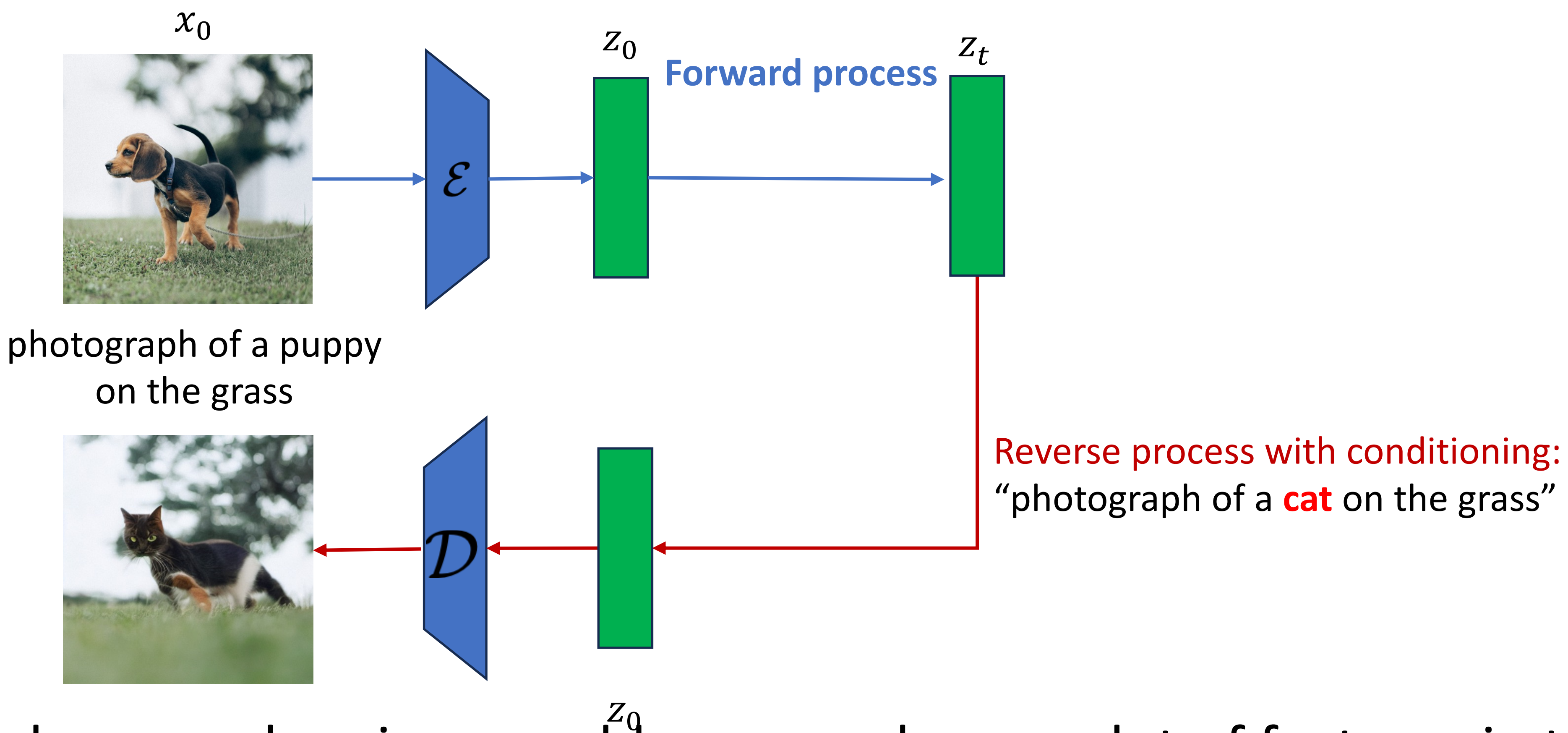
- Last lecture: stable diffusion can perform conditional generation using a text prompt



Text prompt: "photograph of a puppy on the grass"

Naïve Image Editing Idea

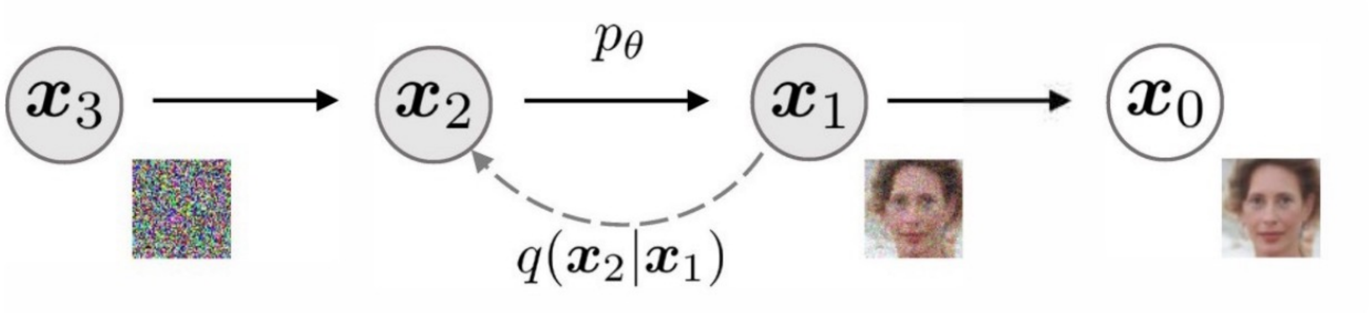
- Instead of starting from pure noise, let us perform naïve inversion using the forward process and a fixed image



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

Better Inversion?

- Problems
  - Randomness in model: if we encode x_0 to x_t using forward process and then rerun reverse process, we won't get x_0 back
  - Reverse process requires T sequential steps, which can be slow.



- What if we had a different sampling mechanism?
  - We will derive a sampling mechanism for pixel-space diffusion models (DDIM) that will allow us to achieve better inversion as a result.

Recap of ELBO

- ELBO

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

- Letting $\overline{\alpha_t} = \prod_{i=1}^{t} \alpha_i$, recall that $q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I\right)$

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\overline{\alpha_t}} x_0, (1-\overline{\alpha_t}) I\right)$$

- Therefore, it holds that

$$\begin{aligned} q(x_{t-1} \mid x_t, x_0) &= \frac{q(x_t \mid x_{t-1}, x_0) q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \\ &= \frac{\mathcal{N}\left(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I\right) \mathcal{N}\left(x_{t-1}; \sqrt{\overline{\alpha_{t-1}}} x_0, (1-\overline{\alpha_{t-1}}) I\right)}{\mathcal{N}\left(x_t; \sqrt{\overline{\alpha_t}} x_0, (1-\overline{\alpha_t}) I\right)} \\ &\propto \mathrm{N}\Bigg(x_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\overline{\alpha_{t-1}}) x_t + \overline{\alpha}_{t-1}(1-\alpha_t) x_0}{1-\overline{\alpha_t}}}_{\mu_q(x_t, x_0)}, \underbrace{\frac{(1-\alpha_t)(1-\overline{\alpha_{t-1}})}{1-\overline{\alpha_t}} I}_{\Sigma_q(t)}\Bigg) \end{aligned}$$

Denoising Diffusion Implicit Models (DDIM)

- Recall our ELBO derivation

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

- 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

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

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

- Choose variance of p to match exactly variance of q

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

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

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

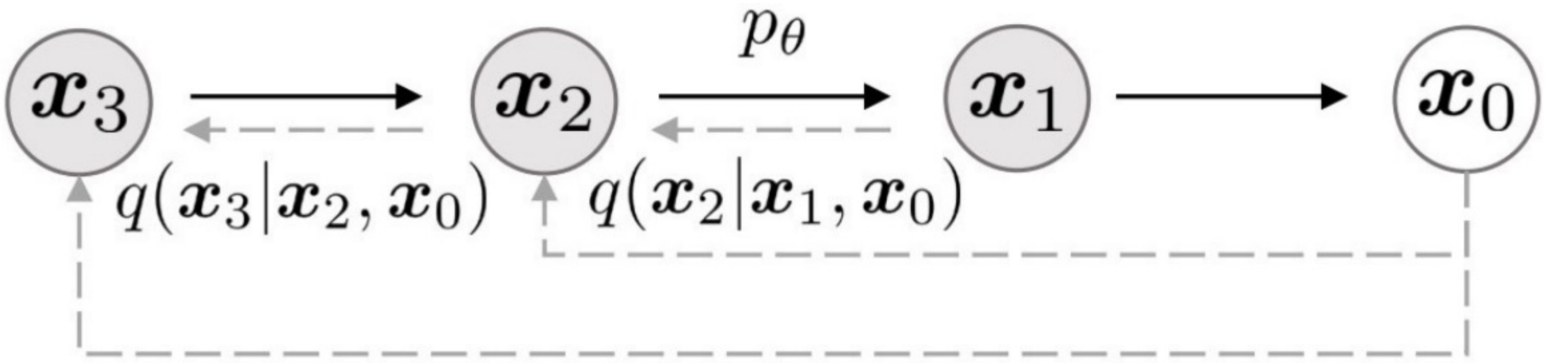
$$\mu_\theta(x_t, t) = \sqrt{\overline{\alpha}_{t-1}}\widehat{x_\theta}(x_t, t) + \sqrt{1 - \overline{\alpha}_{t-1} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha}_t}\widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha}_t}}$$

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

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

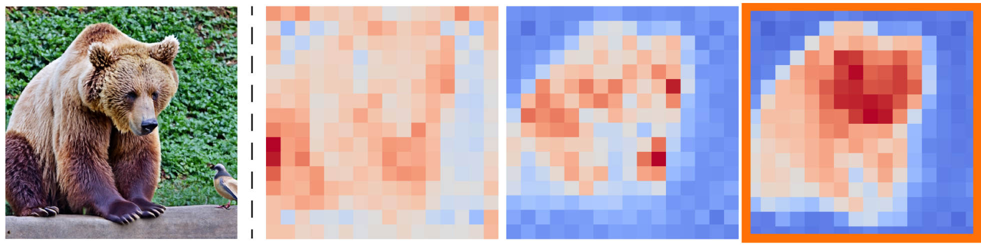
$$x_{t-1} = \underbrace{\sqrt{\overline{\alpha_{t-1}}}\,\widehat{x_\theta}(x_t, t)}_{\text{Predicted } x_0} + \underbrace{\sqrt{1 - \overline{\alpha_{t-1}} - \sigma_t^2}\,\frac{x_t - \sqrt{\overline{\alpha_t}}\,\widehat{x_\theta}(x_t, t)}{\sqrt{1 - \overline{\alpha_t}}}}_{\text{Direction pointing to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



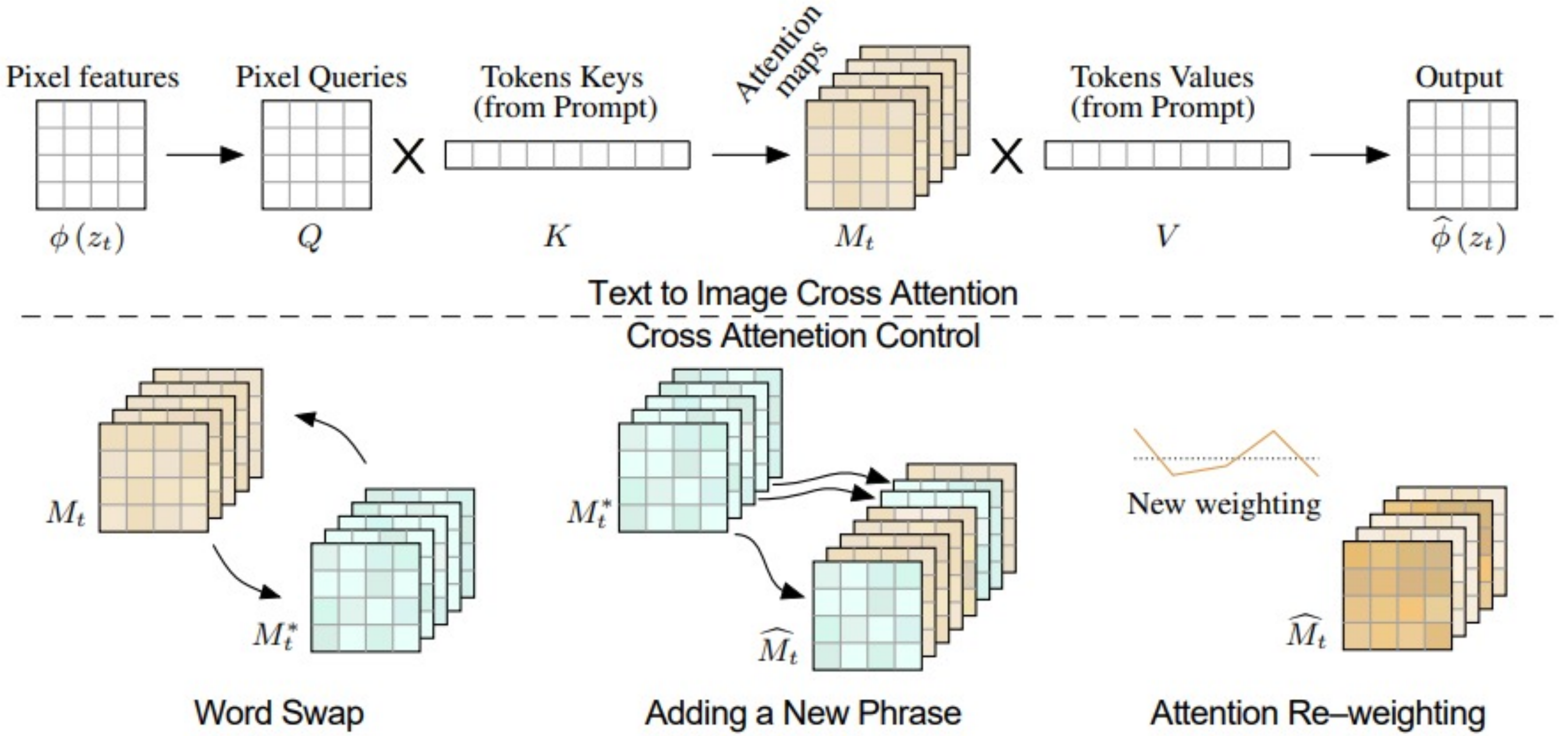
DDIM Inversion

- 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