LDM(Latent Diffusion Model)

ソース

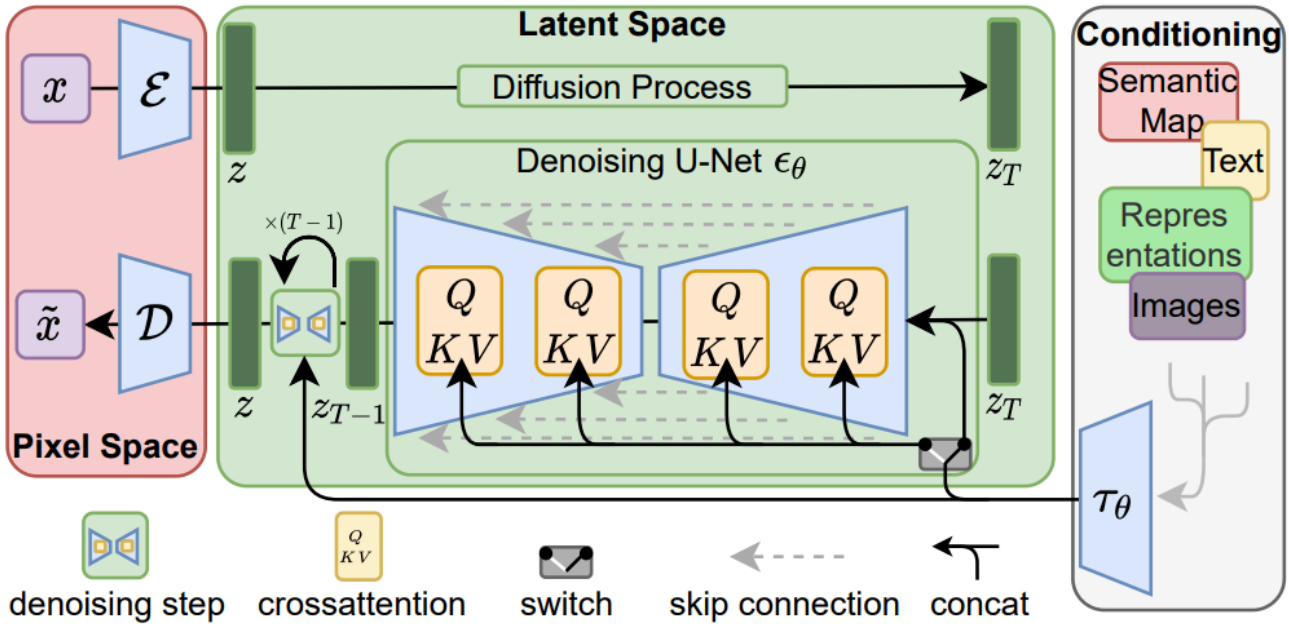
- High-Resolution Image Synthesis with Latent Diffusion Models

概要

- Stable Diffusionの論文
- diffusion modelのパラメータ数を減らすためにVAEを導入
  - encoderを通して潜在空間上でdiffusion modelに入力、出力をdecoderに入力して戻す
- 条件付けをU-netにcross attentionを導入することで実現

提案手法

- 全体図は以下(論文より引用)



- 上半分が拡散過程(学習時)で下半分が逆拡散過程(生成時)
- 横のconditioningが条件付け方法
  - 条件の内容はテキストでも画像でもなんでもよくて、encoder τ_θを通して潜在空間ベクトルを得られることが大事

VAEの導入

- diffusion modelのパラメータ数を減らすためにVAEを通して潜在空間上でdiffusion modelを使うようにした
- 厳密に説明すると以下
- 拡散過程の前に、encoder $\mathcal{E}$を用いて入力画像 xから潜在表現 $z = \mathcal{E}(x)$を得る
- 逆拡散過程の後に、decoder $\mathcal{D}$を用いて出力zから$\tilde{x} = \mathcal{D}(z)$を得る
  - $x \in \mathbb{R}^{H \times W \times c}$,$z \in \mathbb{R}^{h \times w \times c}$
  - $f = \frac{H}{h} = \frac{W}{w}$ をダウンサンプリングの比率として定義
  - $f = 2^m (m \in \mathbb{N})$　とかける
- 以上のように潜在空間上で拡散モデルを扱うことから、diffusion modelにおける損失関数L_{DM}はL_{LDM}に変わる

$$
\begin{aligned}
L_{\mathrm{DM}} &= \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,1), t}[\| \epsilon - \epsilon_\theta(x_t, t) \|_2^2] \\
L_{\mathrm{LDM}} &= \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t}[\| \epsilon - \epsilon_\theta(z_t, t) \|_2^2]
\end{aligned} \tag{1}
$$

条件付け

- 上図のようにU-Netのskip connectionされている各高さにおいてcross attentionを導入
- 条件yにたいしてその分野におけるencoder τ_θを用意して潜在表現$\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$を得て、これから$K, V$を得る
- QはU-netの前層の出力から得る

$$
\begin{aligned}
\mathrm{Attention}(Q, K, V) &= \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \\
Q &= W_Q^{(i)} \cdot \varphi_i(z_t) \\
K &= W_K^{(i)} \cdot \tau_\theta(y) \\
V &= W_V^{(i)} \cdot \tau_\theta(y)
\end{aligned} \tag{2}
$$

- $\varphi(z_t) \in \mathbb{R}^{N \times d_\epsilon^i}$ はU-netの中間層出力
- $W_V^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}, W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}, W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$
- このとき、損失関数は以下
  $L_{\mathrm{LDM}} = \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t}[\| \epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y)) \|_2^2]$