2023

Lossy Image Compression with Conditional Diffusion Models

Ruihan Yang, Stephan Mandt
https://arxiv.org/abs/2209.06950

汇报人：周楚骎

End-to-end optimized lossy image compression framework using diffusion generative models

目录

CONTENTS

01

Introduction

Deep Learning Based Image Codecs

Intro

Deep Learning Based Image Codecs

- Tradeoff between rate and distortion, perceptual quality
- State-of-the-art learned codecs(VAEs):
  - Transform coding --- lower dimensional latent space
  - Hierarchical compressive variational autoencoders --- a learned prior model for entropy-coding
  - **Drawback: mode averaging behavior --- loss of detail**
- Solution: expressive generative model

02

Related Work & Background

Transform-coding Lossy Image Compression

DDPM: Denoising Diffusion Probabilistic Models

DDIM: Denoising Diffusion Implict Models

Related

Transform-coding Lossy Image Compression

Classical codecs

JPEG 1991, BPG 2018, WEBP 2022

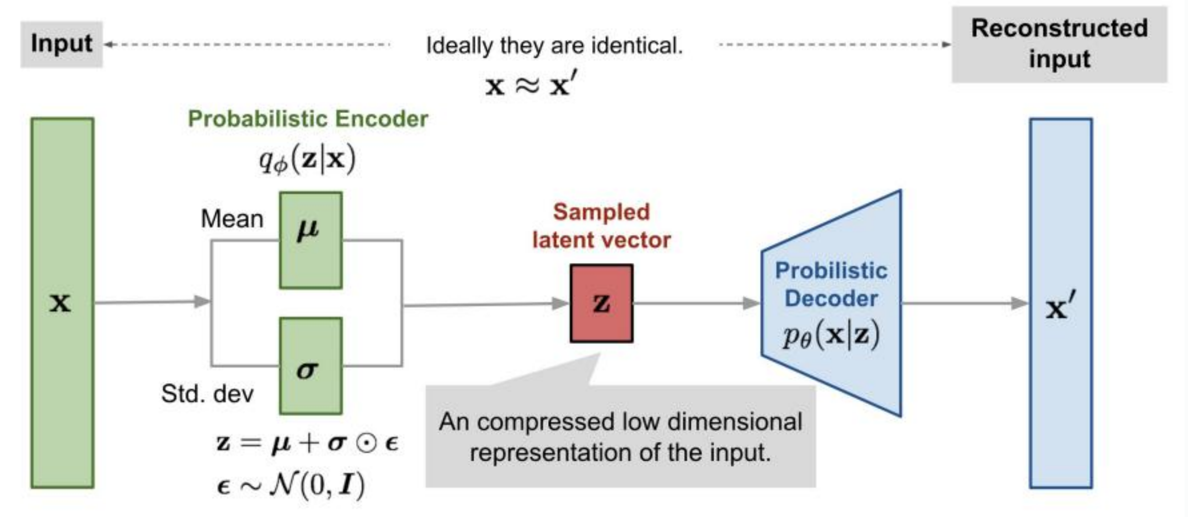
End-to-end learned codecs / VAEs

Minimize KL-divergence

Maximize Evidence Lower Bound (ELBO)

KL(q(z)||p(z|x))

$$\mathcal{L}(\lambda, \mathbf{x}) = \mathcal{D} + \lambda\mathcal{R} = \mathbb{E}_{\mathbf{z}\sim e(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}) - \lambda \log p(\mathbf{z})].$$

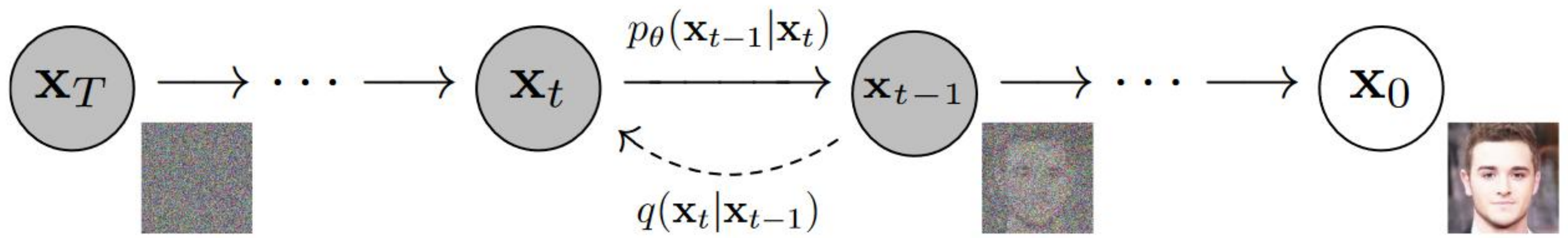


Related

DDPM: Denoising Diffusion Probabilistic Models

DDPM: Denoising Diffusion Probabilistic Models

Generate data by a sequence of iterative stochastic denosing steps



Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Loss function:

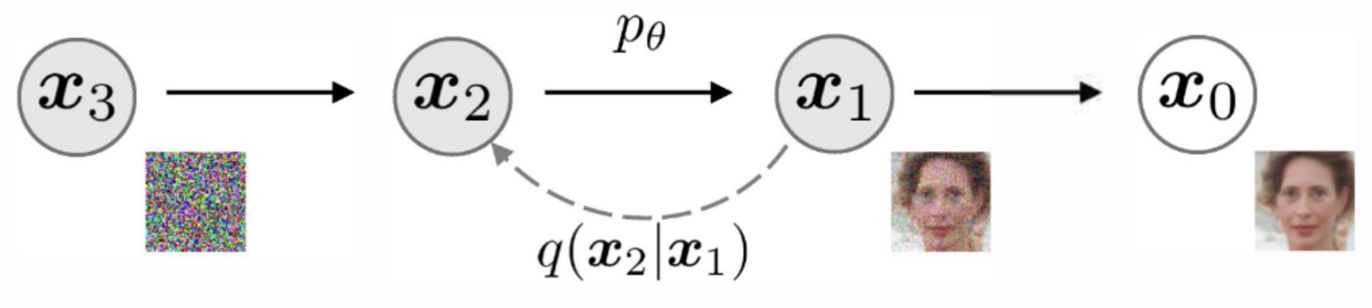
$$L_{\mathrm{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2\right]$$

Related

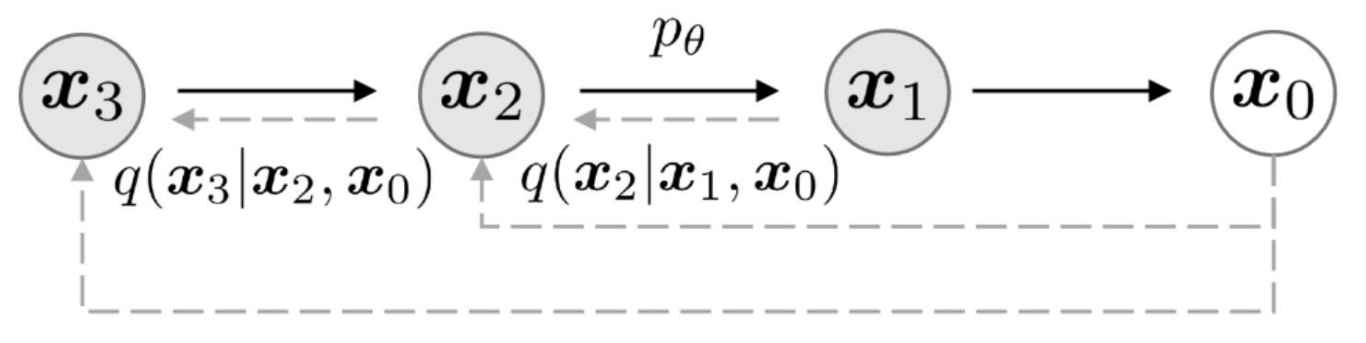
DDIM: Denoising Diffusion Implict Models

DDIM: Denoising Diffusion Implict Models

DDPM: Markovian diffusion process



DDIM: Non-Markovian diffusion process



03

Method

Conditional Diffusion Model for Compression

Method

Algorithm: Training

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

DDPM

根据带噪图像去还原噪声

t 越大，噪声越大

Loss function 仅考虑distortion

Algorithm 1 Training the model (left); Encoding/

Sample $\mathbf{x}_0 \sim$ dataset
repeat
$n \sim \mathcal{U}(0, 1, 2, .., N_{\text{train}})$
$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\bar{\mathbf{x}}_n = \sqrt{\alpha_n}\mathbf{x}_0 + \sqrt{1-\alpha_n}\epsilon$
$\hat{\mathbf{z}} = \mathrm{Enc}_\phi(\mathbf{x}_0) + \mathcal{U}(-0.5, 0.5)$
$\bar{\mathbf{x}}_0 = \mathcal{X}_\theta(\bar{\mathbf{x}}_n, n/N_{\text{train}}, \hat{\mathbf{z}})$
$L_{\mathrm{D}} = \frac{\alpha_n}{1-\alpha_n}|\mathbf{x}_0 - \bar{\mathbf{x}}_0|^2$
$L = (1-\rho)L_{\mathrm{D}} + \rho d(\bar{\mathbf{x}}_0, \mathbf{x}_0) - \lambda \log_2 P(\hat{\mathbf{z}})$
$(\theta, \phi) = (\theta, \phi) - \varepsilon \nabla_{\theta,\phi} L$ (learning rate: ε)
until converge

CDC

根据带噪图像直接还原原图

n/N_{train} (pseudo-continuous)越大，噪声越大

Loss function考虑distortion, bitrate, perceptual metric

Method

Algorithm: Encoding/ Decoding

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

DDPM

根据带噪图像去还原噪声，原图减去噪声

t 越大，噪声越大

/Decoding data $\mathbf{x}_0$ (right). $\mathcal{X}$-prediction model.

Given N_{test}
$\hat{\mathbf{z}} = \lfloor \text{Enc}_\phi(\mathbf{x}_0) \rceil$
$\hat{\mathbf{z}} \xleftrightarrow{P(\hat{\mathbf{z}})}$ binary file (entropy code using $P(\hat{\mathbf{z}})$)
$\bar{\mathbf{x}}_N = \mathbf{0}$ (or $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \gamma^2 \mathbf{I})$ for stochastic decoding)
for n=N_{test} to 1 **do**
$\epsilon_\theta = \frac{\mathbf{x}_n - \sqrt{\alpha_n} \mathcal{X}_\theta(\mathbf{x}_n(\mathbf{x}_0), \mathbf{z}, \frac{n}{N})}{\sqrt{1-\alpha_n}}$
$\bar{\mathbf{x}}_0 = \mathcal{X}_\theta(\bar{\mathbf{x}}_n, n/N_{\text{test}}, \hat{\mathbf{z}})$
$\bar{\mathbf{x}}_{n-1} = \sqrt{\alpha_{n-1}} \bar{\mathbf{x}}_0 + \sqrt{1-\alpha_{n-1}} \epsilon_\theta$
end for return $\bar{\mathbf{x}}_0$

CDC

根据带噪图像直接还原噪声小一点的带噪图像

n/N中的N训练和推理过程可以不一样

Method

Training objective

单击此处输入你的正文，文字是您思想的提炼，为了最终演示发布的良好效果，请尽量言简意赅的阐述观点；根据需要可酌情增减文字…

- Rate-Distortion Function of VAE:

$$\mathcal{L}(\lambda, \mathbf{x}) = \mathcal{D} + \lambda\mathcal{R} = \mathbb{E}_{\mathbf{z}\sim e(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}) - \lambda \log p(\mathbf{z})].$$

- By Jensen's inequality:

$$\mathbb{E}_{\mathbf{z}\sim e(\mathbf{z}|\mathbf{x}_0)}[-\log p(\mathbf{x}_0|\mathbf{z}) - \lambda \log p(\mathbf{z})] \leq \mathbb{E}_{\mathbf{z}\sim e(\mathbf{z}|\mathbf{x}_0)}\left[L_{\text{upper}}(\mathbf{x}_0|\mathbf{z}) - \lambda \log p(\mathbf{z})\right],$$

$$L_{\text{upper}}(\mathbf{x}_0|\mathbf{z}) = -\mathbb{E}_{\mathbf{x}_{1:N}\sim q(\mathbf{x}_{1:N}|\mathbf{x}_0)}\left[\log \frac{p(\mathbf{x}_{0:N}|\mathbf{z})}{q(\mathbf{x}_{1:N}|\mathbf{x}_0)}\right]$$

- Loss Function of DDFM:

$$L(\theta, \mathbf{x}_0) = \mathbb{E}_{n,\epsilon}||\epsilon - \epsilon_\theta(\mathbf{x}_n(\mathbf{x}_0), n)||^2.$$

- Simlify the training objective:

$$L_{\text{upper}}(\mathbf{x}_0|\mathbf{z}) \approx \mathbb{E}_{\mathbf{x}_0,n,\epsilon}||\epsilon - \epsilon_\theta(\mathbf{x}_n, \mathbf{z}, \frac{n}{N_{\text{train}}})||^2 = \mathbb{E}_{\mathbf{x}_0,n,\epsilon}\frac{\alpha_n}{1-\alpha_n}||\mathbf{x}_0 - \mathcal{X}_\theta(\mathbf{x}_n, \mathbf{z}, \frac{n}{N_{\text{train}}})||^2$$

- Optional perceptual metric(LPIPS loss):

$$L_{\text{p}} = \mathbb{E}_{\epsilon,n,\mathbf{z}\sim e(\mathbf{z}|\mathbf{x}_0)}[d(\bar{\mathbf{x}}_0, \mathbf{x}_0)] \text{ and } L_{\text{c}} = \mathbb{E}_{\mathbf{z}\sim e(\mathbf{z}|\mathbf{x}_0)}[L_{\text{upper}}(\mathbf{x}_0|\mathbf{z}) - \frac{\lambda}{1-\rho}\log p(\mathbf{z})]$$

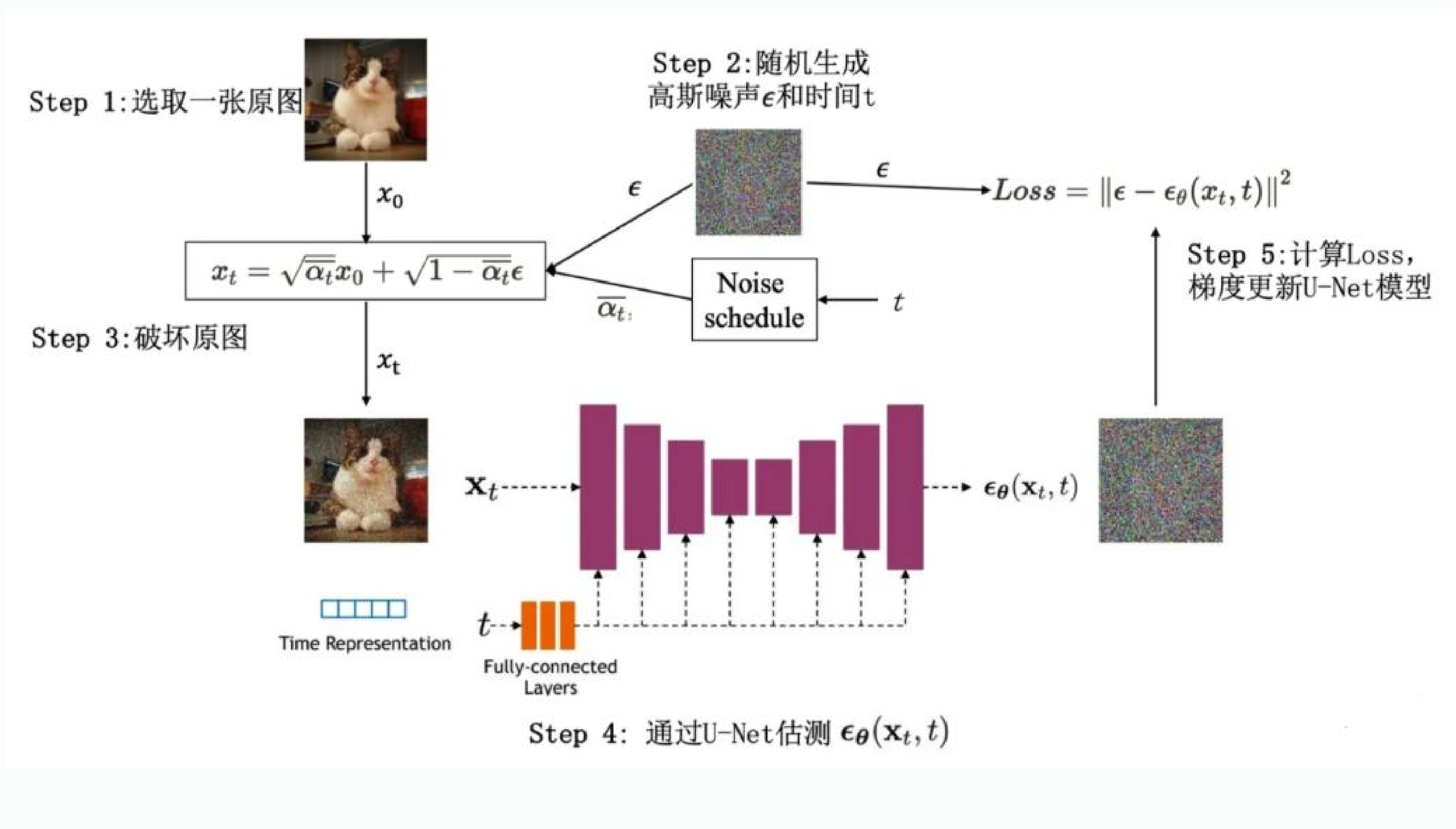
$$L = \rho L_{\text{p}} + (1-\rho)L_{\text{c}}.$$

Method

Training process of DDPM

U-Net

Fullt-connected Layers



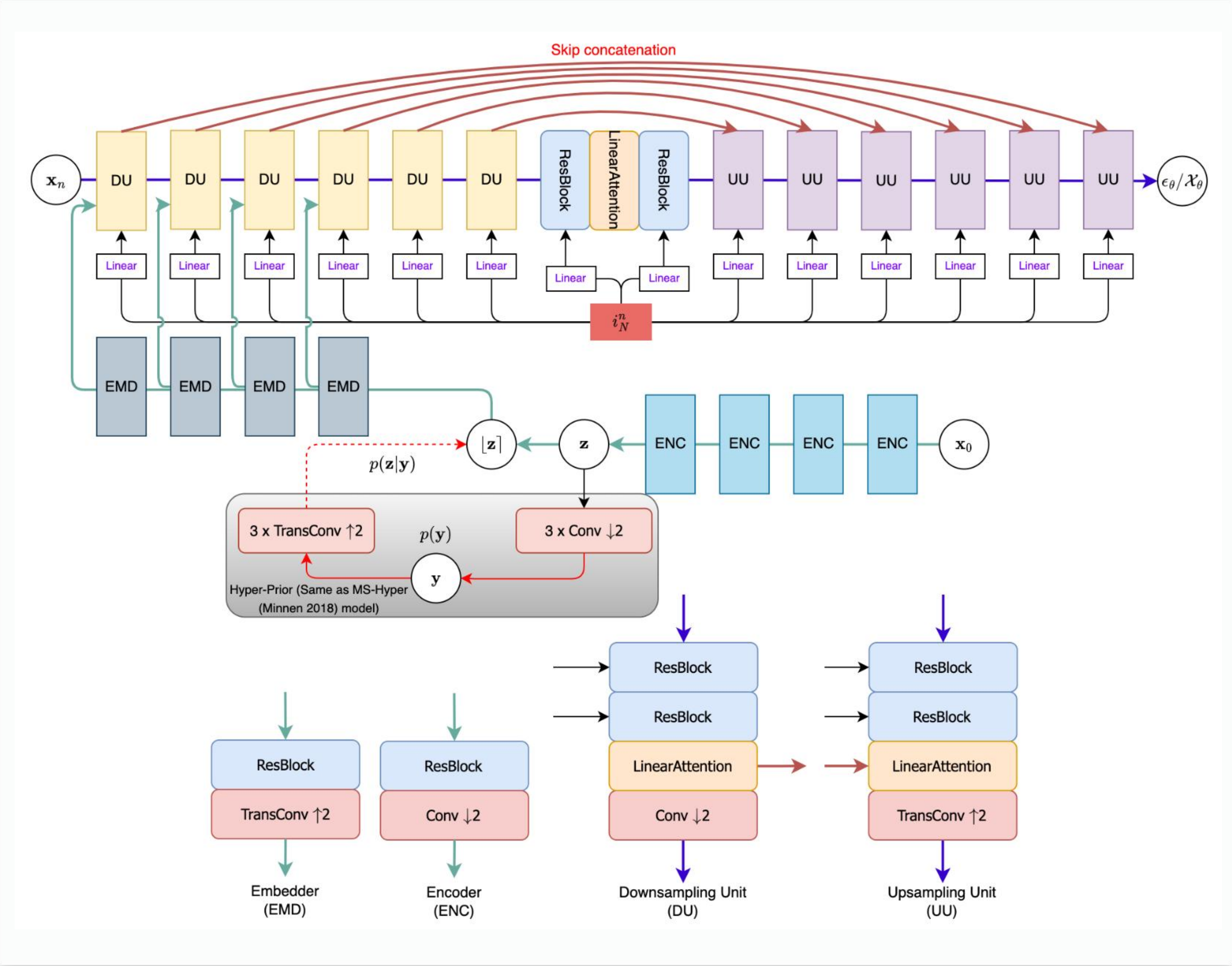
Method

Visualization of their Model Architecture

Skip concatenation

Linear Attention

Hyper-Prior



04

Experiments

Conditional Diffusion Model for Compression

Experiments

Metrics

- Perceptual metrics:
  - FID, LPIPS, PieAPP, DISTS
- Distortion metrics:
  - FSIM, MS-SSIM, SSIM, PSNR

Test Data

- Kodak: 768x512
- Tecnick: 600x600
- DIC2K: 768x768
- COCO2017: 384x384

Model Training

Vimeo-90k:

9000 clips each select one frame and crop to 256x256

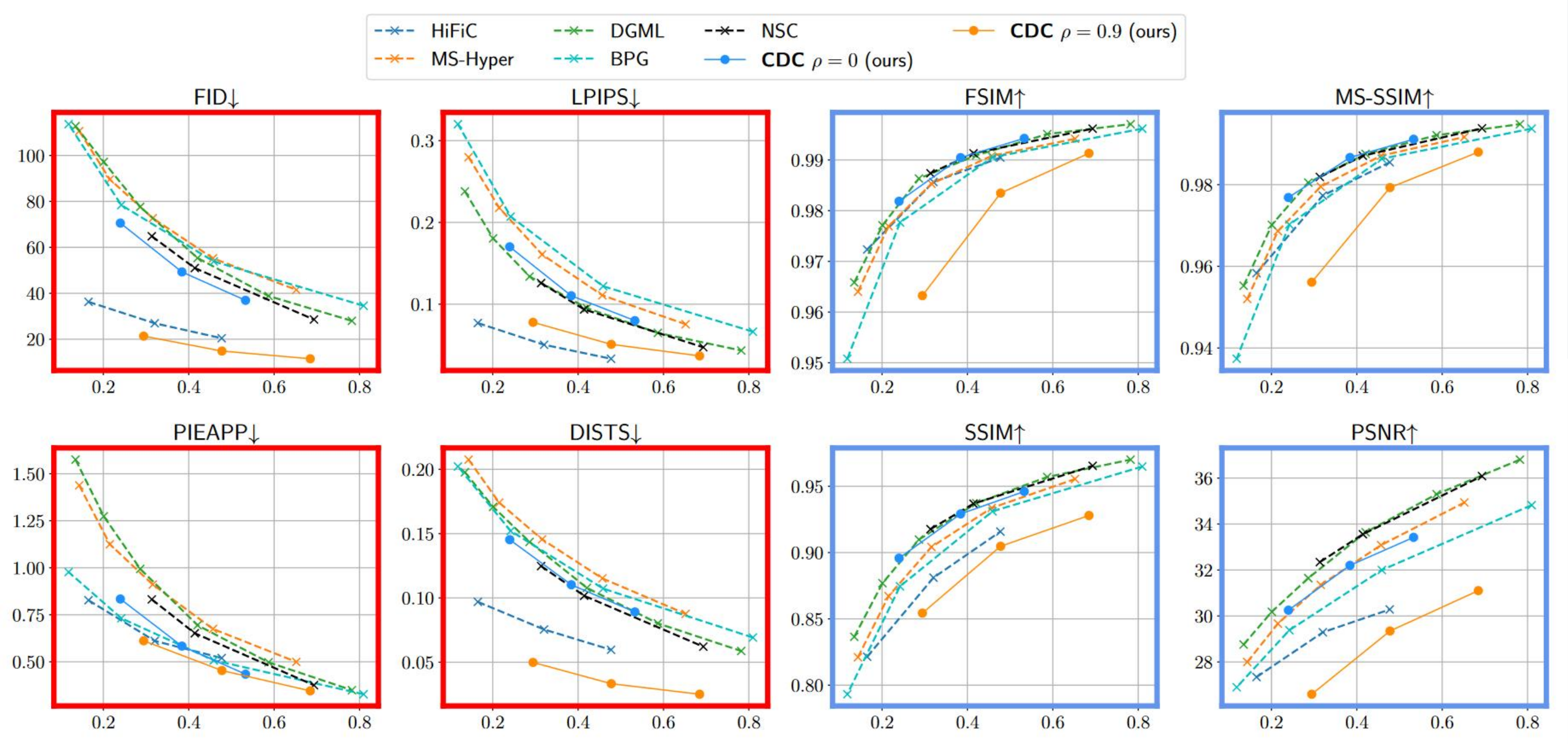
Baselines

Additive perceptual metric

Without perceptual metric

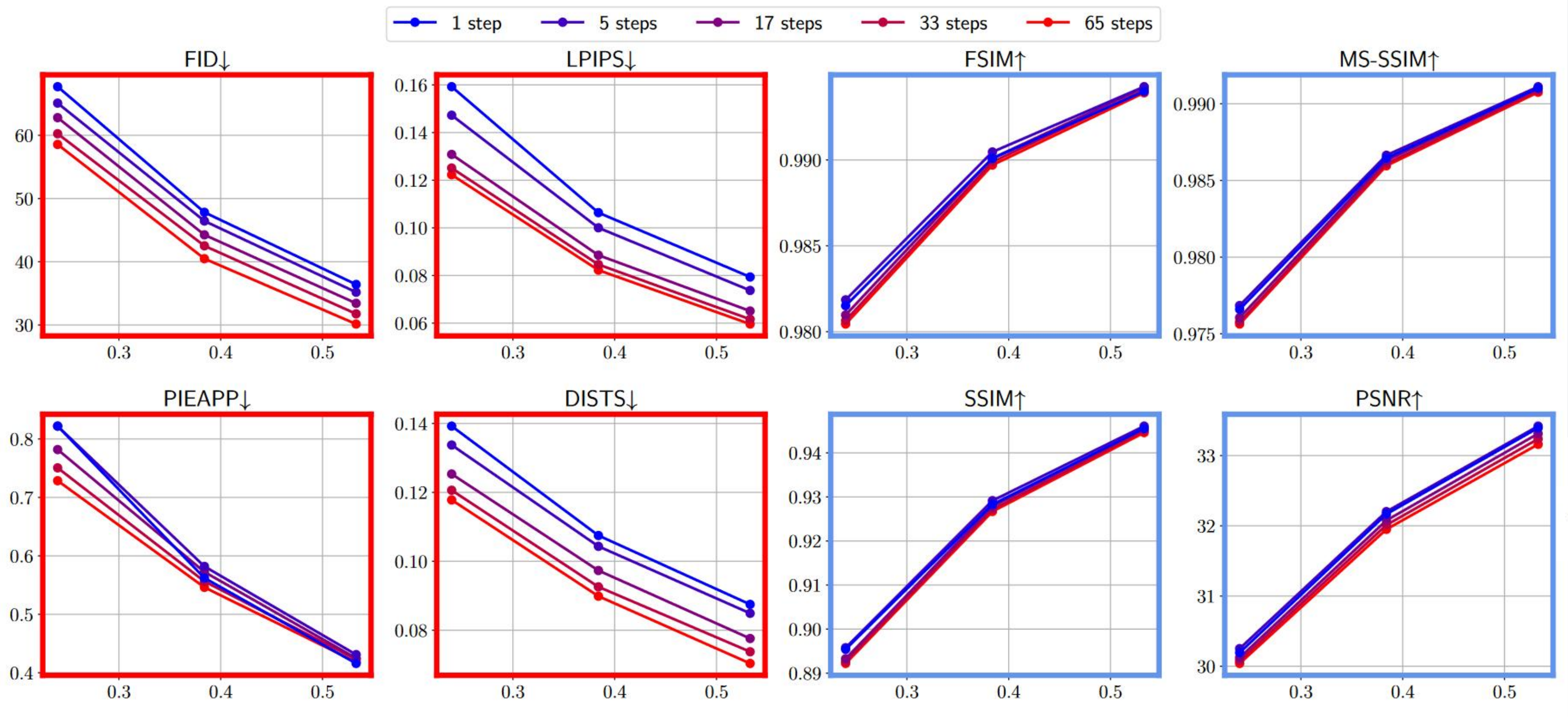
HiFiC,DGML,

NSC,MS-Hyper,BPG



- Perceptual Metrics(red): CDC p=0.9 (orange circle)
- Distortion Metrics(blue): CDC p=0 (blue circle)

$\gamma = 1$



$\gamma = 0$

$\bar{\mathbf{x}}_N = \mathbf{0}$ (or $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \gamma^2\mathbf{I})$ for stochastic decoding)

- When employing **stochastic decoding**, the model consistently produces better perceptual results as the number of decoding steps increases.
- However, in the case of **deterministic decoding**, more decoding steps do not lead to a substantial improvement in distortion.

05

Summary

Conditional Diffusion Model for Compression

Sum

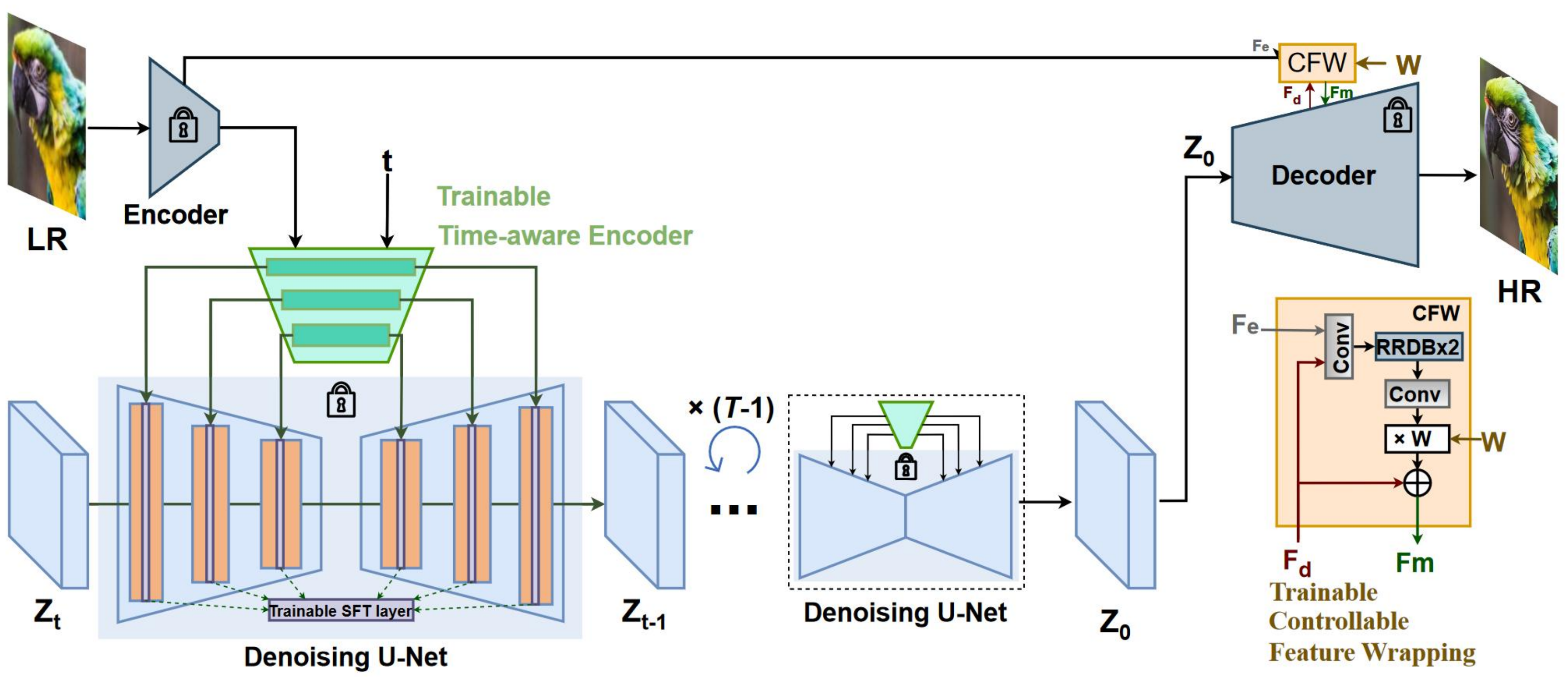
Conditional Diffusion Model for Compression

- Tradeoff between rate and distortion, perceptual quality
- Reconstruct image with less noise from image with noise
  - DDPM: construct noise from image with noise
- A variable that characterizes the intensity of noise
- Improvement:
  - Integrate advanced techniques such as autoregressive entropy models or iterative encoding

Others

Exploiting Diffusion Prior for Real-World Image Super-Resolution

- Fine-tuning Stable Diffusion
- Time-aware Encoder
- Spatial feature transformations(SFT)
- Controllable Feature Warpping



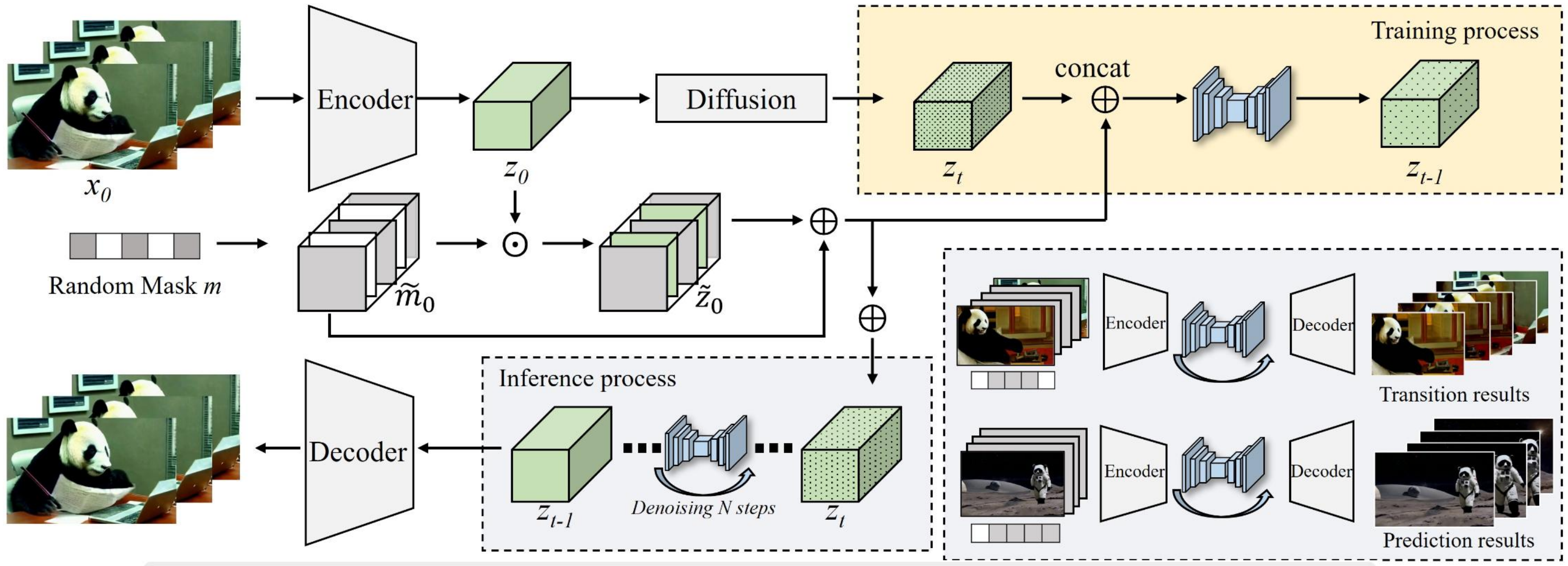
- Time-aware Encoder: Image content is rapidly populated when the SNR approaches 5e-2
- Controllable feature warpping module: continuous fidelity-realism trade-off

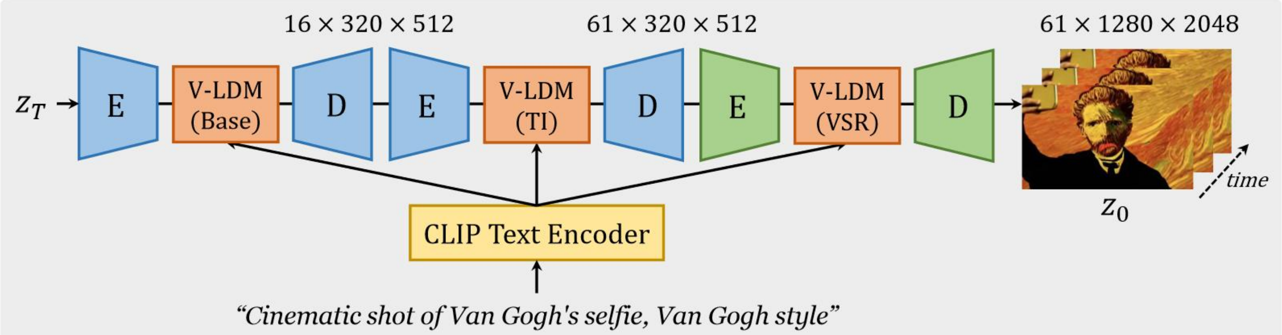
Others

SEINE: Short-to-Long Video Diffusion Model For Generative Transition And Prediction

Random-mask video diffusion model

LaVie: Pre-trained diffusion-based T2V model





- Adapting the conventional 2D UNet architecture into a spatial-temporal 3D network
- Latent diffusion models (LDMs)
- Generate frames for any given frames at arbitrary positions

2 0 2 3

谢谢观看