Diffusion Autoencoders: Toward a Meaningful and Decodable Representation

Konpat Preechakul   Nattanat Chatthee
Suttisak Wizadwongsa   Supasorn Suwajanakorn

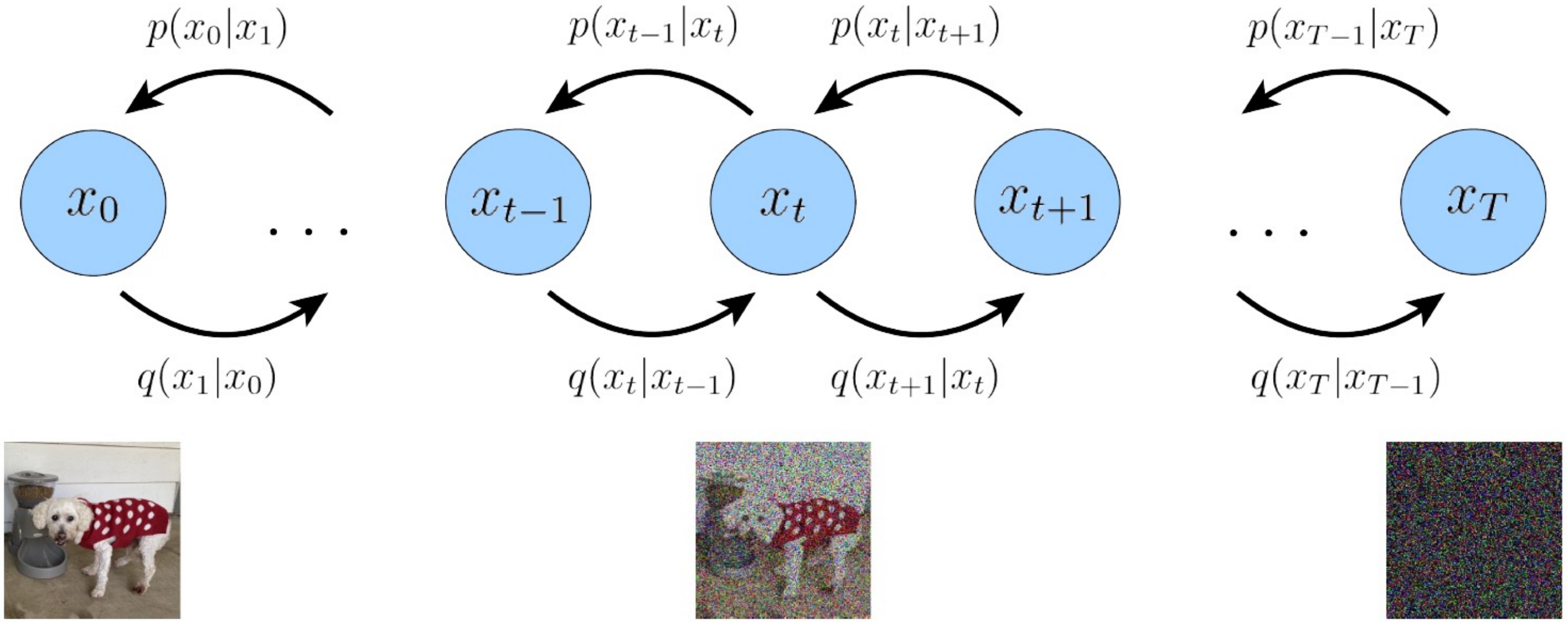
VISTEC, Thailand

Outline

- Authorship
- Background
- Method
- Conclusion

Background

➢Denoising diffusion probabilistic model



Background

➢Denoising diffusion probabilistic model

- Forward process
  - projection from original image to a Gaussian noise by adding Gaussian noise gradually

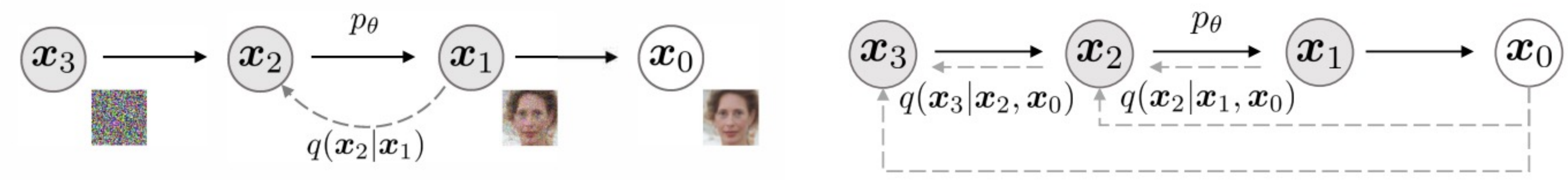
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

- Reverse process
  - reversion of forward process, just as the name suggests

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Background

➢Denoising diffusion implicit model



Graphical models for diffusion and non-Markovian inference models

Background

➢Denoising diffusion implicit model

- Forward process
  - Non-markovian forward process

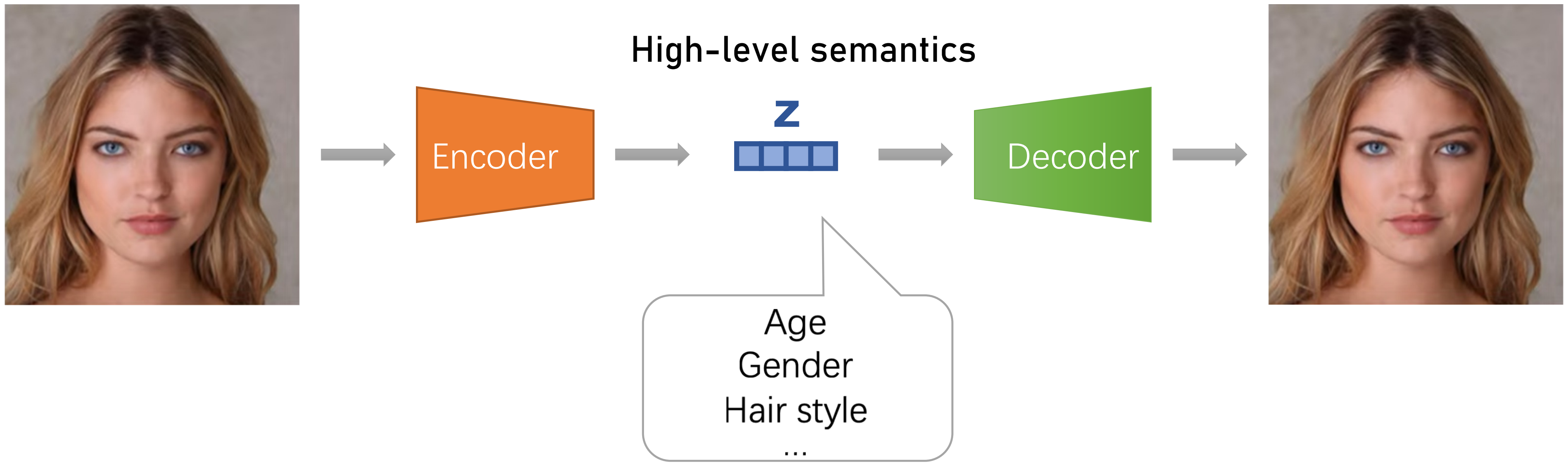
$$q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_T}\boldsymbol{x}_0, (1-\alpha_T)\boldsymbol{I})$$

- Reverse process

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \boldsymbol{I}\right)$$

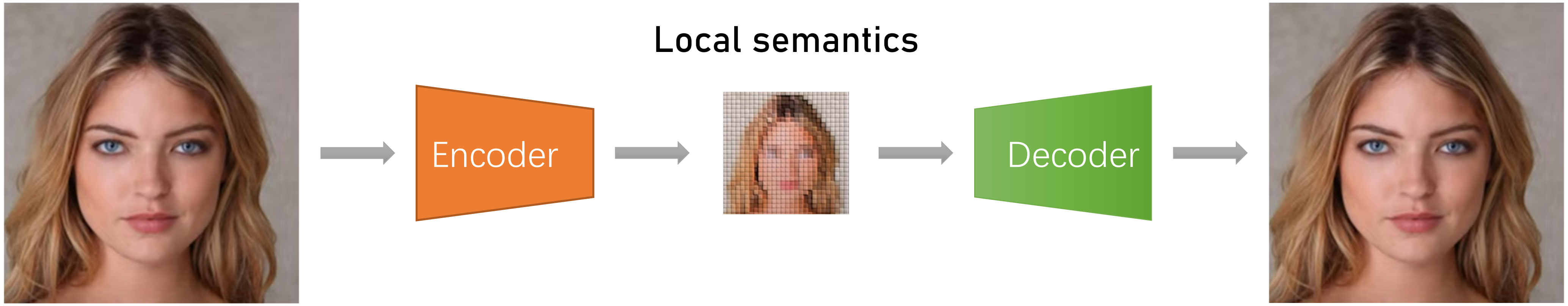
Background

➢Representation learning



Background

➢Representation learning



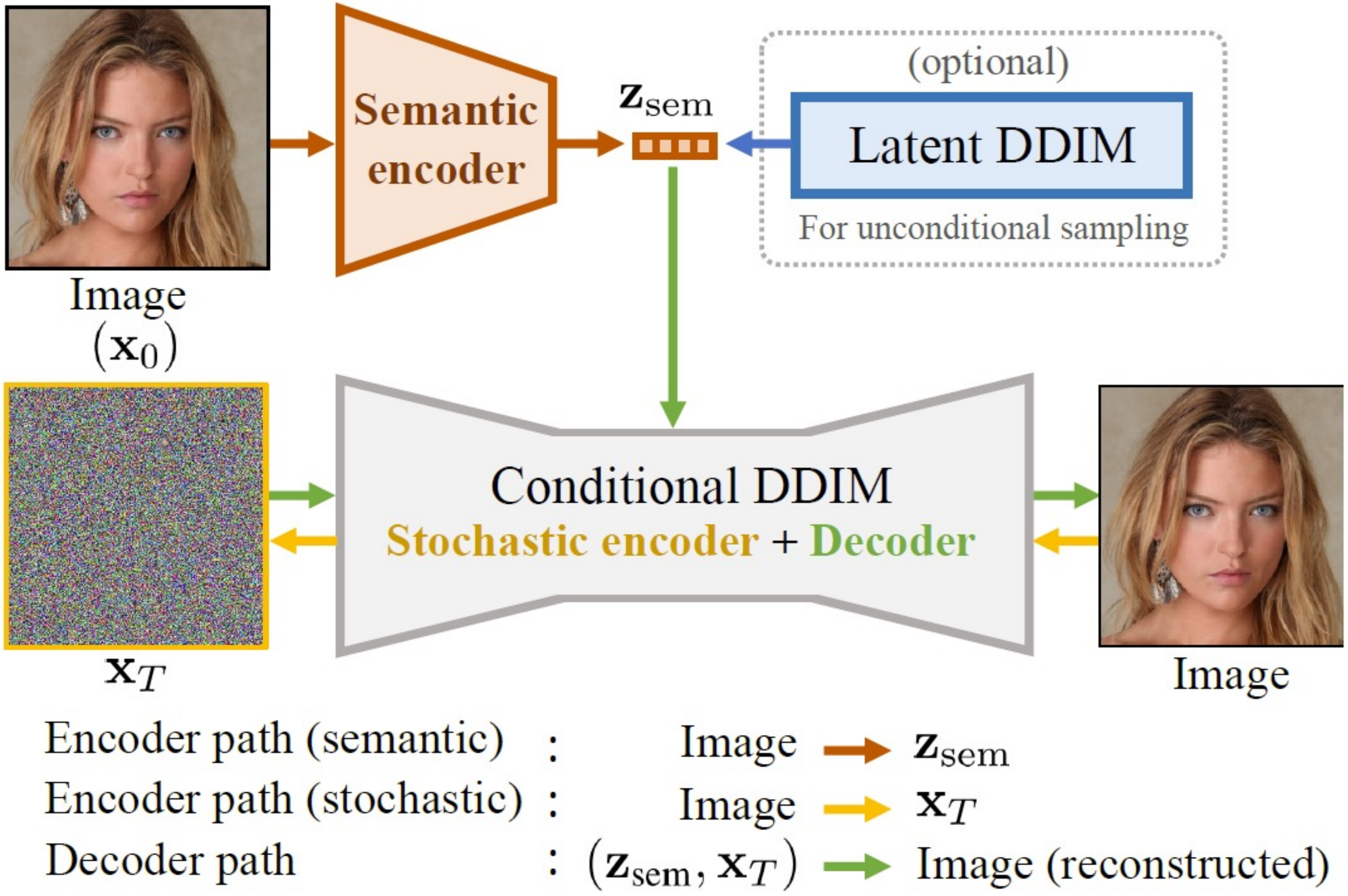
Method

➢Target

- Learn a representation
  - High-level semantics
  - Allowing near-exact reconstruction

Method

- ➢Pipeline



Method

➢Diffusion-based Decoder

- Generative process

$$p_\theta(\mathbf{x}_{0:T} \mid \mathbf{z}_{\text{sem}}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{z}_{\text{sem}})$$

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{z}_{\text{sem}}) = \begin{cases} \mathcal{N}(\mathbf{f}_\theta(\mathbf{x}_1, 1, \mathbf{z}_{\text{sem}}), \mathbf{0}) & \text{if } t = 1 \\ q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}})) & \text{otherwise} \end{cases}$$

Method

➢Diffusion-based Decoder

- Noise prediction network

$$\mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}})\right)$$

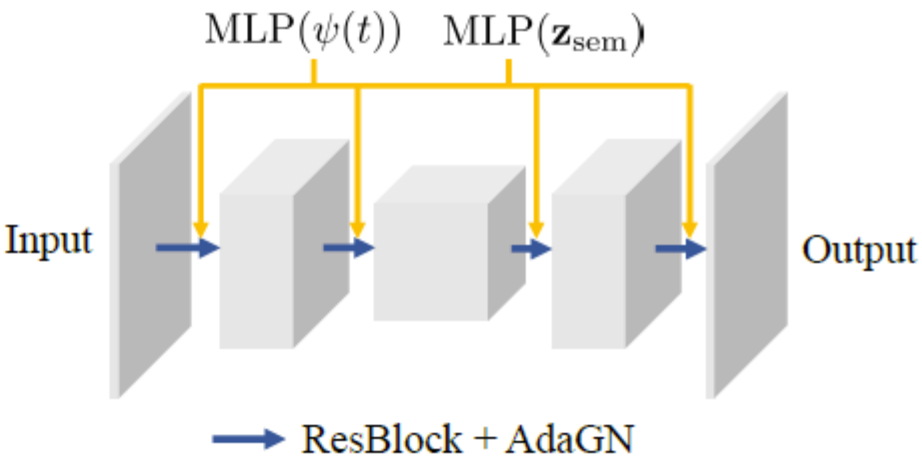
- Loss function

$$L_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0, \epsilon_t}\left[\|\epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) - \epsilon_t\|_2^2 \right]$$

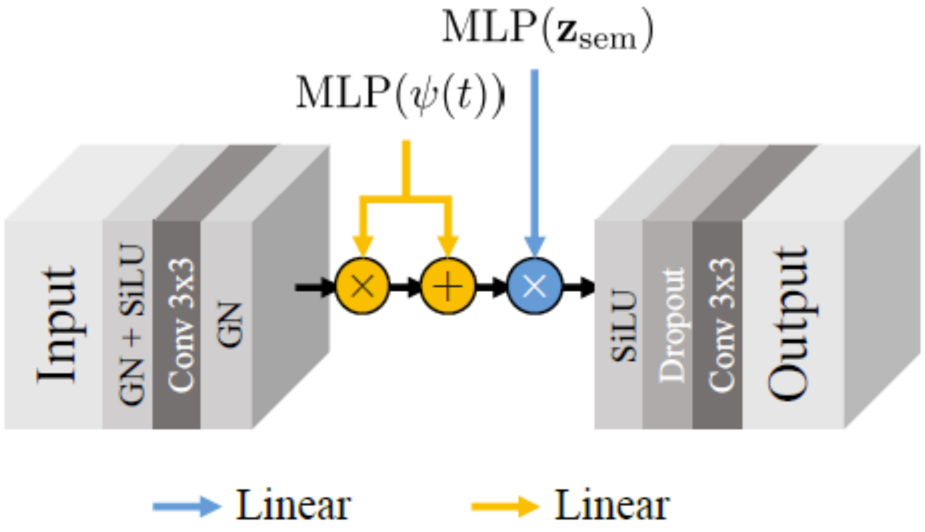
Method

➢Diffusion-based Decoder

- Architecture



(a) Diffusion autoencoder (Diff-AE)'s UNet decoder conditioned by $\mathbf{z}_{\text{sem}}$.



(b) ResBlock + AdaGN. The residual path is not depicted.

Method

➢Stochastic encoder

- Deterministic generative process backward

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}}\mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) + \sqrt{1-\alpha_{t+1}}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}})$$

Method

➢Latent DDIM

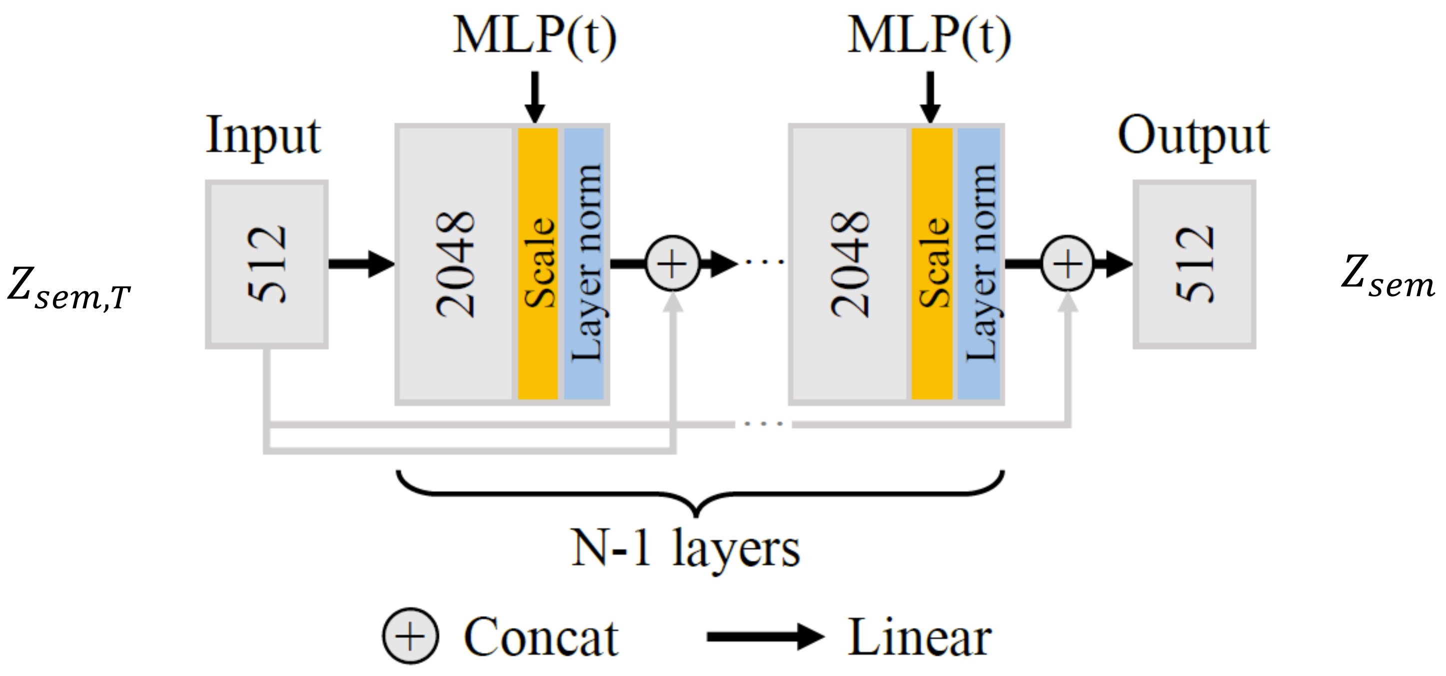
- Sampling with diffusion autoencoder

$$L_{\text{latent}} = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{z}_{\text{sem}},\epsilon_t}\Big[\, \|\epsilon_\omega(\mathbf{z}_{\text{sem},t}, t) - \epsilon_t\|_1 \Big]$$

Method

➢Latent DDIM

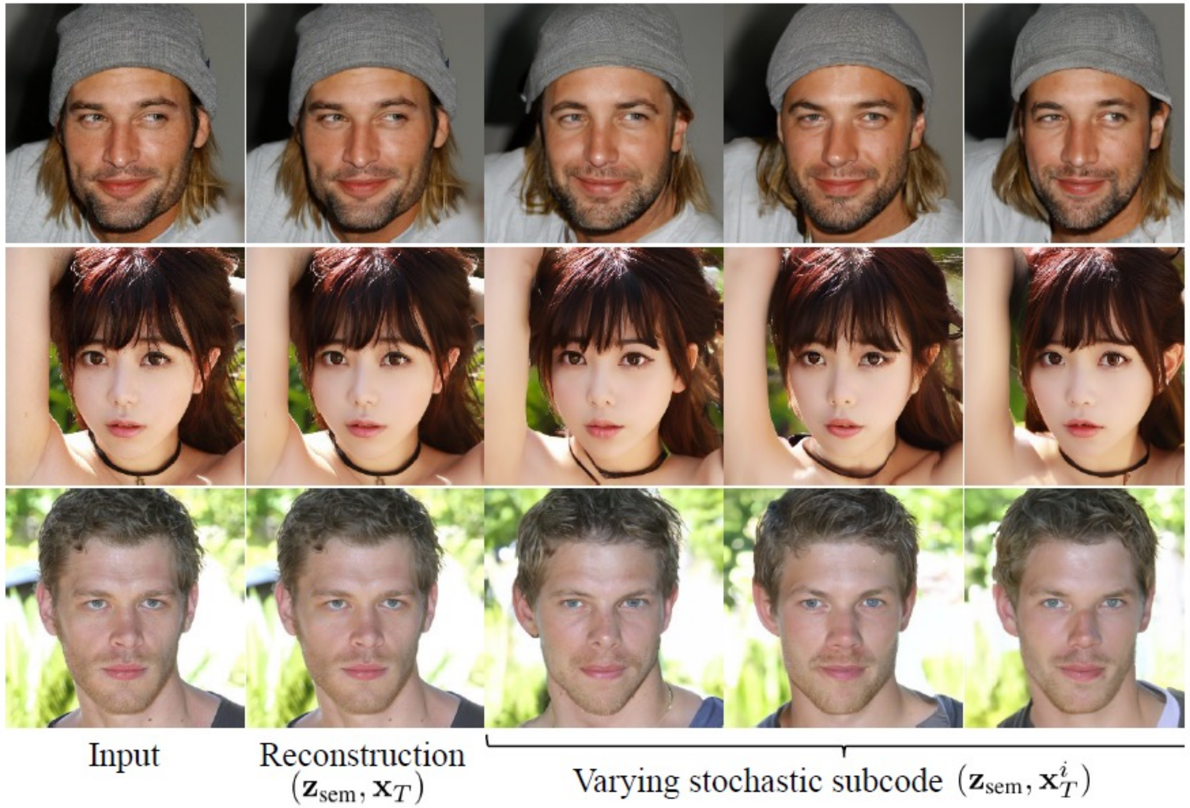
- Architecture



Experiment

➢Latent code

- Latent code captures both high-level semantics and low-level stochastic variations



Experiment

➢Latent code

- Latent code captures both high-level semantics and low-level stochastic variations

Model	SSIM ↑				LPIPS ↓				MSE ↓			
	T=10	T=20	T=50	T=100	T=10	T=20	T=50	T=100	T=10	T=20	T=50	T=100
DDIM (@130M) [47]	0.600	0.760	0.878	0.917	0.227	0.148	0.087	0.063	0.019	0.008	0.003	0.002
Ours (@130M, 512D $\mathbf{z}_{\text{sem}}$)	0.827	0.927	0.978	**0.991**	0.078	0.050	0.023	**0.011**	0.001	0.001	0.000	**0.000**
a) No encoded $\mathbf{x}_T$	0.707	0.695	0.683	0.677	0.085	0.078	0.074	0.073	0.006	0.007	0.007	0.007
b) No encoded $\mathbf{x}_T$, @48M, 512D $\mathbf{z}_{\text{sem}}$	0.662	0.650	0.637	0.631	0.102	0.096	0.093	0.092	0.009	0.009	0.009	0.010
c) No encoded $\mathbf{x}_T$, @48M, 256D $\mathbf{z}_{\text{sem}}$	0.637	0.624	0.612	0.606	0.116	0.109	0.106	0.105	0.010	0.011	0.011	0.011
d) No encoded $\mathbf{x}_T$, @48M, 128D $\mathbf{z}_{\text{sem}}$	0.613	0.600	0.588	0.582	0.133	0.127	0.125	0.124	0.012	0.012	0.013	0.013
e) No encoded $\mathbf{x}_T$, @48M, 64D $\mathbf{z}_{\text{sem}}$	0.551	0.538	0.527	0.521	0.168	0.165	0.163	0.162	0.018	0.019	0.020	0.020

Experiment

- Semantically meaningful latent interpolation

(a) StyleGAN2 interpolation after $\mathcal{W}$ space inversion.

(b) StyleGAN2 interpolation after $\mathcal{W}+$ space inversion.

(c) DDIM interpolation.

(d) Our diffusion autoencoder interpolation.

Experiment

➢Attribute manipulation

Figure 5. Real-image attribute manipulation results on two global attributes (gender, age) and two local attributes (smile, wavy hair) by moving z_{sem} along the positive or negative direction found by linear classifiers. The top two are from FFHQ [27] and the bottom two are from CelebA-HQ [26]. Our method synthesizes highly-plausible and realistic results that preserve an unprecedented level of detail.

Experiment

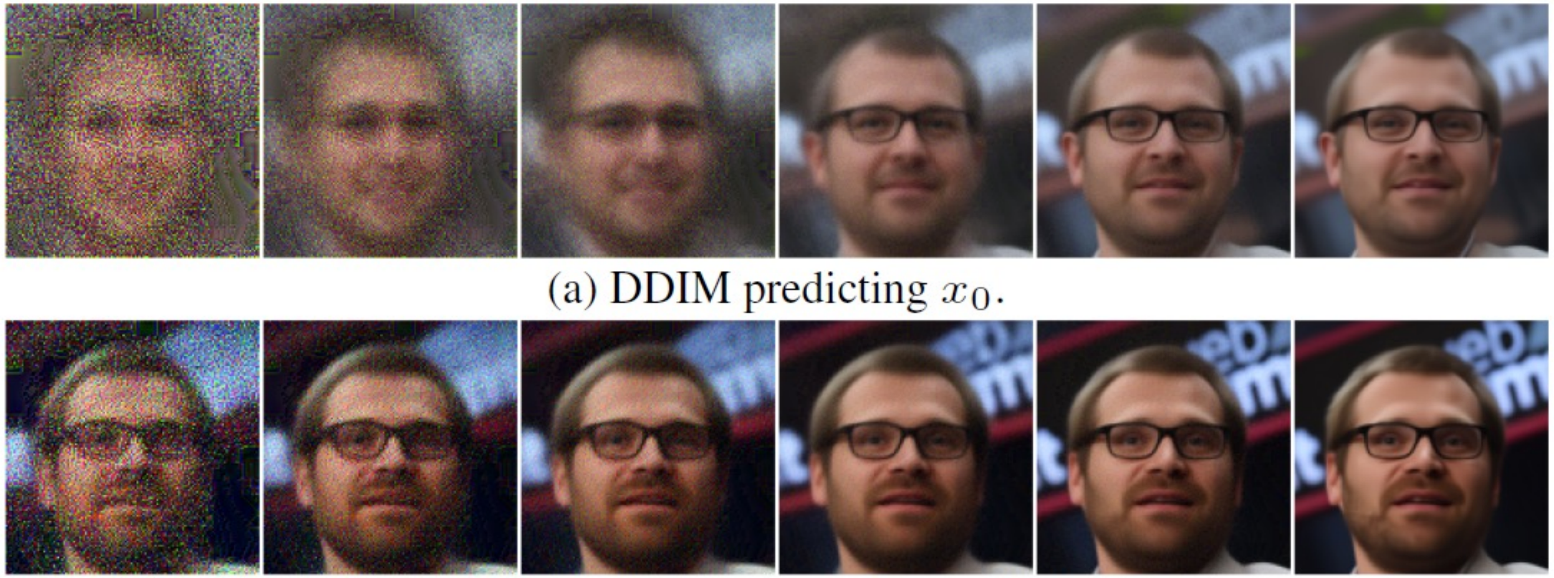
➢Autoencoding reconstruction quality

Table 1. Autoencoding reconstruction quality of models trained on FFHQ [27] and tested on unseen CelebA-HQ [26]. Our model is competitive with state-of-the-art NVAE while producing readily useful high-level semantics in a compact 512D $\mathbf{z}_{\text{sem}}$.

Model	**Latent dim**	**SSIM** ↑	**LPIPS** ↓	**MSE** ↓
StyleGAN2 ($\mathcal{W}$) [28]	512	0.677	0.168	0.016
StyleGAN2 ($\mathcal{W}$+) [28]	7,168	0.827	0.114	0.006
VQ-GAN [13]	65,536	0.782	0.109	3.61e-3
VQ-VAE2 [38]	327,680	0.947	0.012	4.87e-4
NVAE [50]	6,005,760	0.984	**0.001**	**4.85e-5**
DDIM (T=100, 128^2) [47]	49,152	0.917	0.063	0.002
Ours (T=100, 128^2, no $\mathbf{x}_T$)	512	0.677	0.073	0.007
Ours (T=100, 128^2)	49,664	**0.991**	0.011	6.07e-5

Experiment

➢Faster denoising process



(a) DDIM predicting x_0.

(b) Our diffusion autoencoder predicting x_0.

Figure 6. Predicted $\mathbf{x}_0$ at $t_{9,8,7,5,2,0}$ (T=10). By conditioning on $\mathbf{z}_{\text{sem}}$, our method predicts images that resemble $\mathbf{x}_0$ much faster.

Experiment

- Class conditional sampling

Table 3. FID scores (↓) for class-conditional generation on CelebA 64 dataset computed between 5k sampled images and the target subset. ± represents one standard deviation (n=3). D2C [45] results come from their paper (n=1 run of FID computation on 5k samples). Binary classifier was trained with 50 positives and 50 negatives. Positive-unlabeled (PU) classifier was trained with 100 positives and 10,000 unlabeled examples (as negatives). Naive FIDs were computed between all images and the target subset.

Scenario	Classes	Ours	D2C [45]	Naive
Binary	Male	**11.52** ± 1.19	13.44	23.83
	Female	**7.29** ± 0.44	9.51	13.64
	Blond	**16.10** ± 2.00	17.61	25.62
	Non-Blond	**8.48** ± 0.52	8.94	0.96
PU	Male	**9.54** ± 0.54	16.39	23.83
	Female	**9.21** ± 0.19	12.21	13.64
	Blond	**7.01** ± 0.25	10.09	25.62
	Non-Blond	**7.91** ± 0.15	9.09	0.96

Experiment

➢Unconditional sampling

Table 4. FID scores (↓) for unconditional generation. Our method is competitive with DDIM baselines. "+ autoencoding" refers to diffusion autoencoders that infer ground-truth semantic subcode from the test set and do not sample from the latent DDIM.

Dataset	Model	FID ↓			
		T=10	T=20	T=50	T=100
FFHQ 128	DDIM	29.56	21.45	15.08	12.03
	Ours	**20.80**	**16.70**	**12.57**	**10.59**
	+ autoencoding	14.43	10.70	6.69	4.56
Horse 128	DDIM	22.17	12.92	7.92	**5.97**
	Ours	**11.97**	**9.37**	**7.44**	6.71
	+ autoencoding	9.27	6.23	3.87	2.92
Bedroom 128	DDIM	13.70	9.23	7.14	5.94
	Ours	**10.69**	**8.19**	**6.50**	**5.70**
	+ autoencoding	6.36	4.88	3.61	2.88
CelebA 64	DDIM	16.38	12.70	8.52	5.83
	Ours	**12.92**	**10.18**	**7.05**	**5.30**
	+ autoencoding	12.78	9.06	5.15	3.11

Conclusion

- Autoencoder with **near-perfect** reconstruction
- Framework for learning **semantic** representation
- **Simple solution** for many real-image applications