CVPR Nashville JUNE 11-15, 2025

One-Way Ticket : Time-Independent Unified Encoder for Distilling Text-to-Image Diffusion Models

Senmao Li

南開大學
Nankai University

Background: Visual Generation

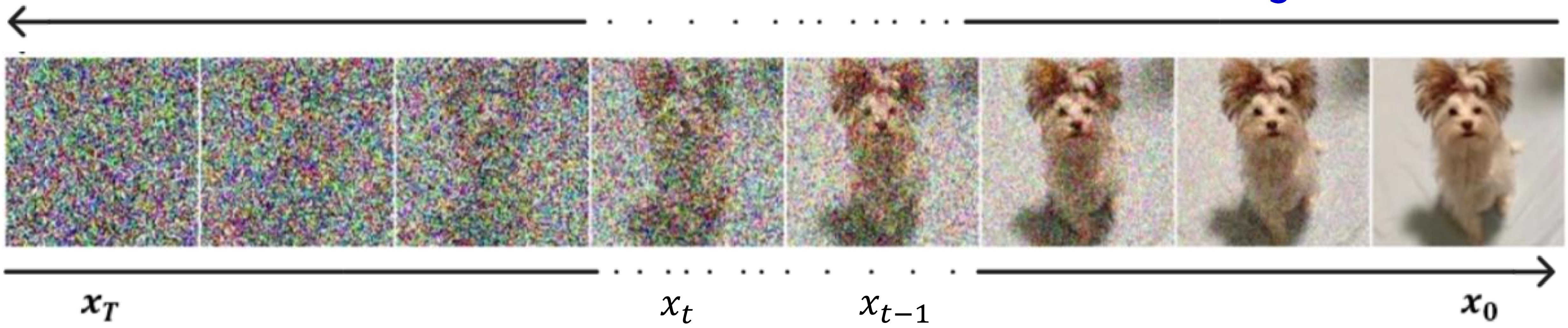
- Image and Video Generation

Stable Diffusion 3

Background

◆ Diffusion Models

Training/Add noise



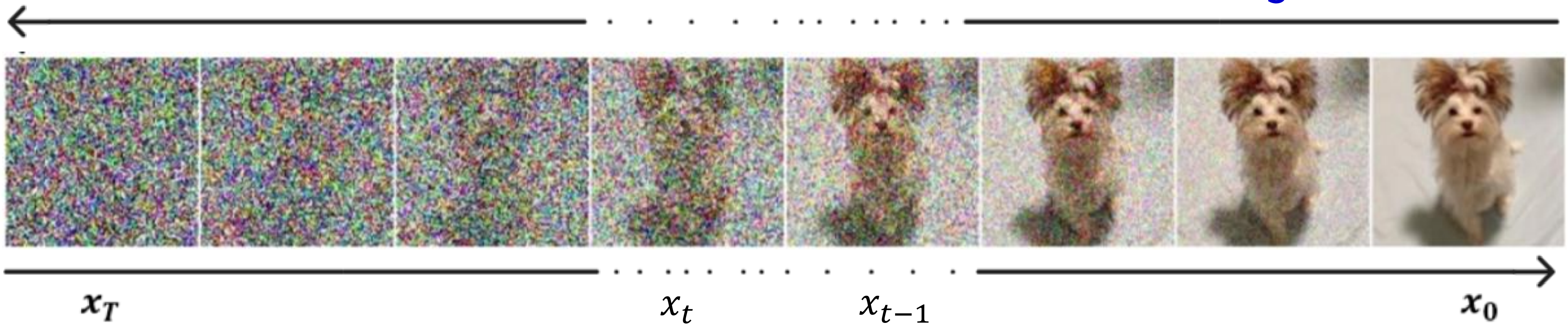
DDPM: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$ 1000 steps Sample/denoise

Ho, Jonathan et al. "Denoising diffusion probabilistic models." *NeurIPS* 33 (2020): 6840-6851.

Background

- Diffusion Models

Training/Add noise



DDPM: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$ 1000 steps

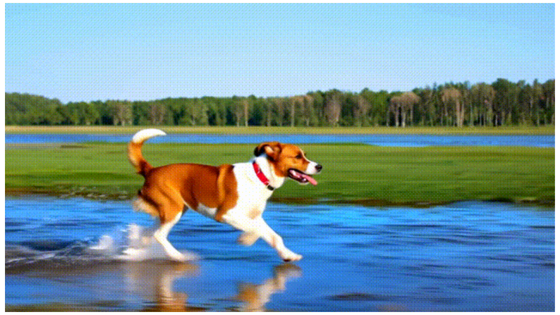
Sample/denoise

DDIM: $\boldsymbol{x}_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}\boldsymbol{x}_t + \sqrt{\alpha_{t-1}}\left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}\right) \cdot \epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ 50 steps

Song, et al. "Denoising Diffusion Implicit Models." *International Conference on Learning Representations 2021*.

Background

◆ Diffusion Models —— Efficiency Latency Challenge



SD 512x512 in RTX3090
(DDPM 1000 steps): 37.6s
(DDIM 50 steps): 2.5s

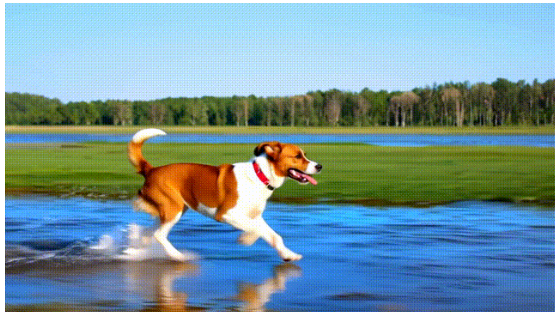
Wan2.1 820x480 in A6000
(DDIM 50 steps): 352s

Song, et al. "Denoising Diffusion Implicit Models." *International Conference on Learning Representations 2021*.

Background

◆ Diffusion Models —— Efficiency Latency Challenge

SD 512x512 in RTX3090
(DDPM 1000 steps): 37.6s
(DDIM 50 steps): 2.5s

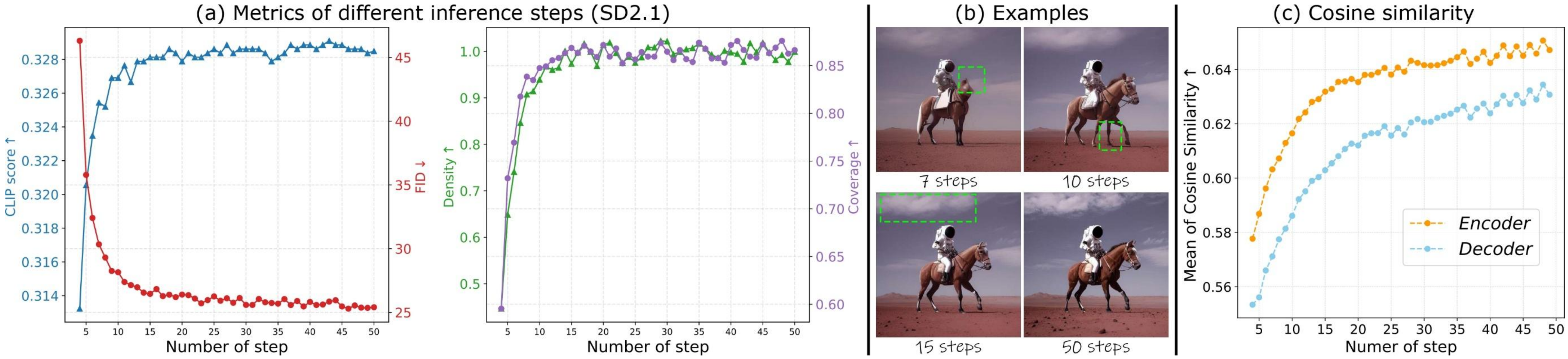


Wan2.1 820x480 in A6000
(DDIM 50 steps): 352s

GANs 512x512
(1 step): 0.02s

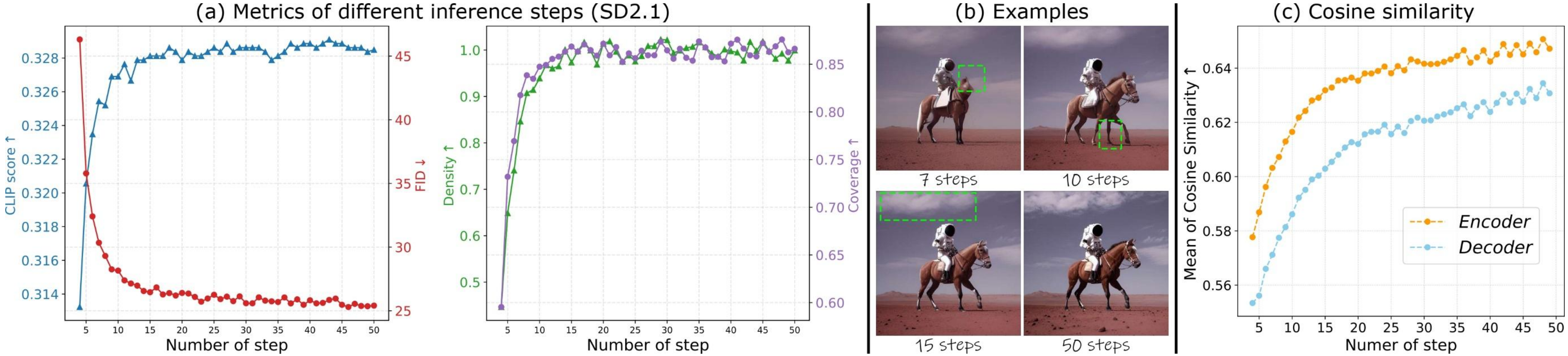
Song, et al. "Denoising Diffusion Implicit Models." *International Conference on Learning Representations 2021*.

Motivation



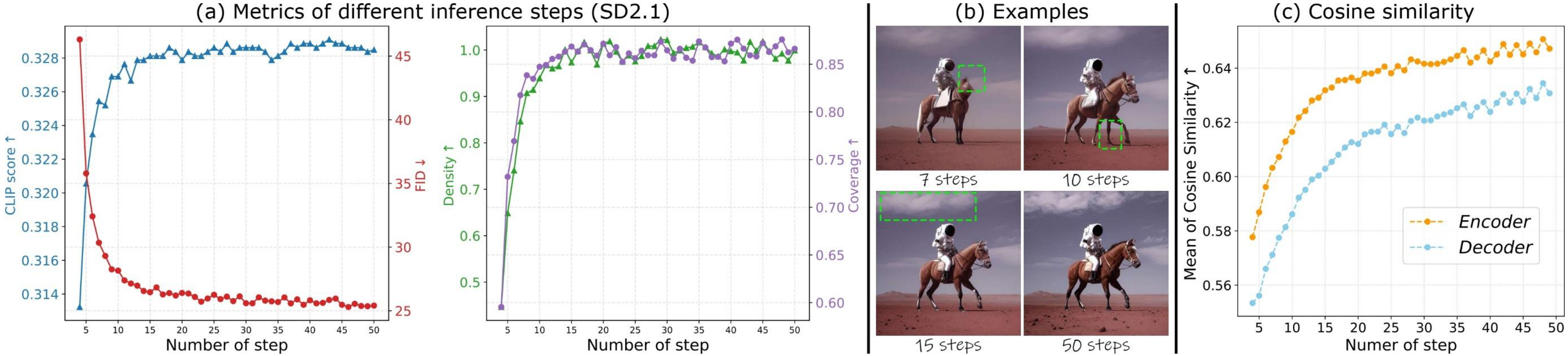
(a) Metrics of different inference steps (SD2.1)
CLIP score ↑
FID ↓
Number of step
Density ↑
Coverage ↑
Number of step
(b) Examples
7 steps
10 steps
15 steps
50 steps
(c) Cosine similarity
Mean of Cosine Similarity ↑
Numer of step
Encoder
Decoder

Motivation



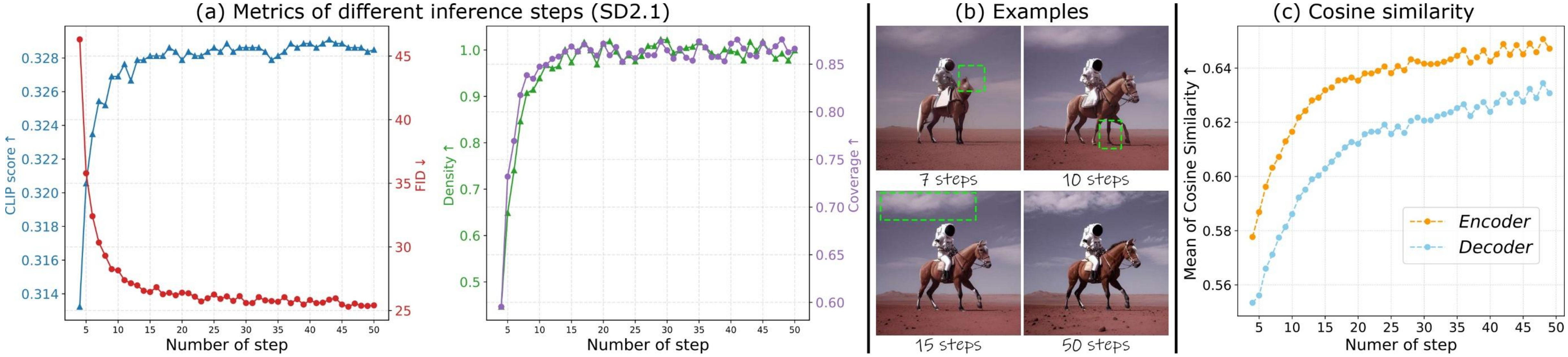
➢ **Above a certain threshold of steps**, such as 15 steps in SD2.1, the model maintains image generation quality (Fig.a-b) while the features show high similarity (Fig.c).

Motivation



- **Above a certain threshold of steps**, such as 15 steps in SD2.1, the model maintains image generation quality (Fig.a-b) while the features show high similarity (Fig.c).
- **Below this threshold**, feature similarity deteriorates along with worse generation quality, accompanied by a degradation in image generation quality as sampling steps reduce.

Motivation



- **Above a certain threshold of steps**, such as 15 steps in SD2.1, the model maintains image generation quality (Fig.a-b) while the features show high similarity (Fig.c).
- **Below this threshold**, feature similarity deteriorates along with worse generation quality, accompanied by a degradation in image generation quality as sampling steps reduce.

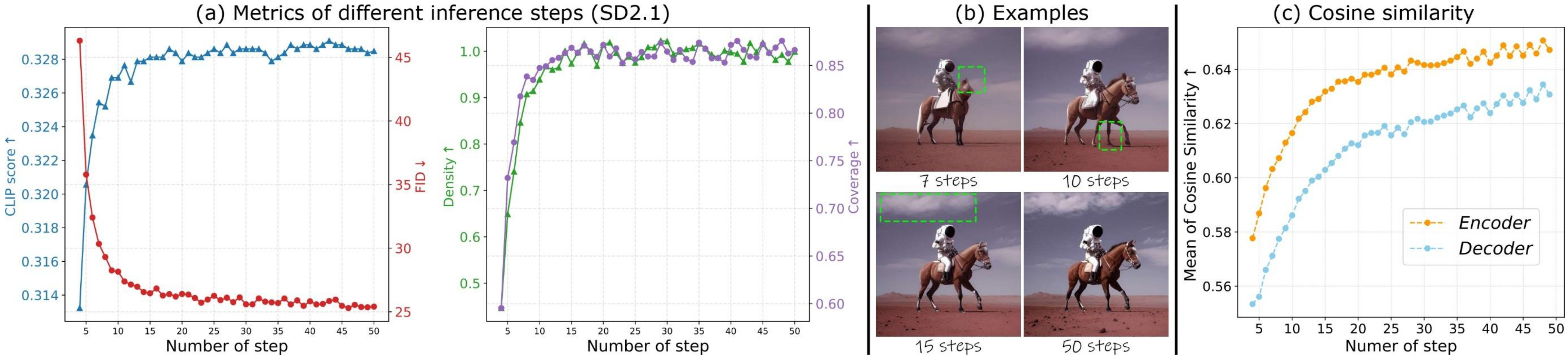

1-step


2-step

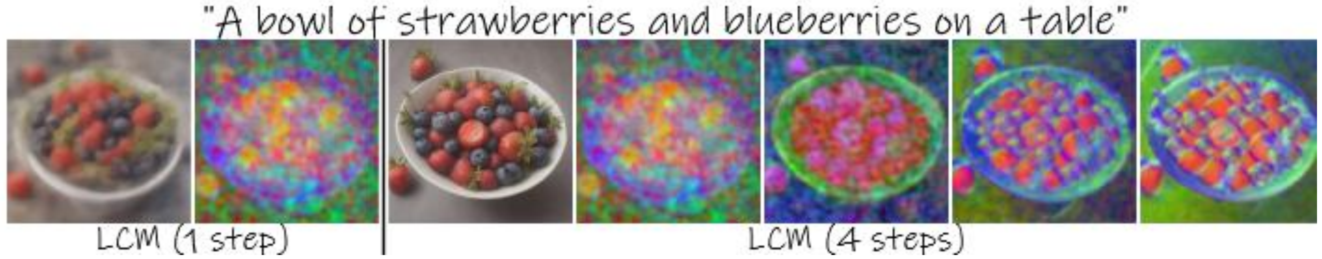

3-step


4-step

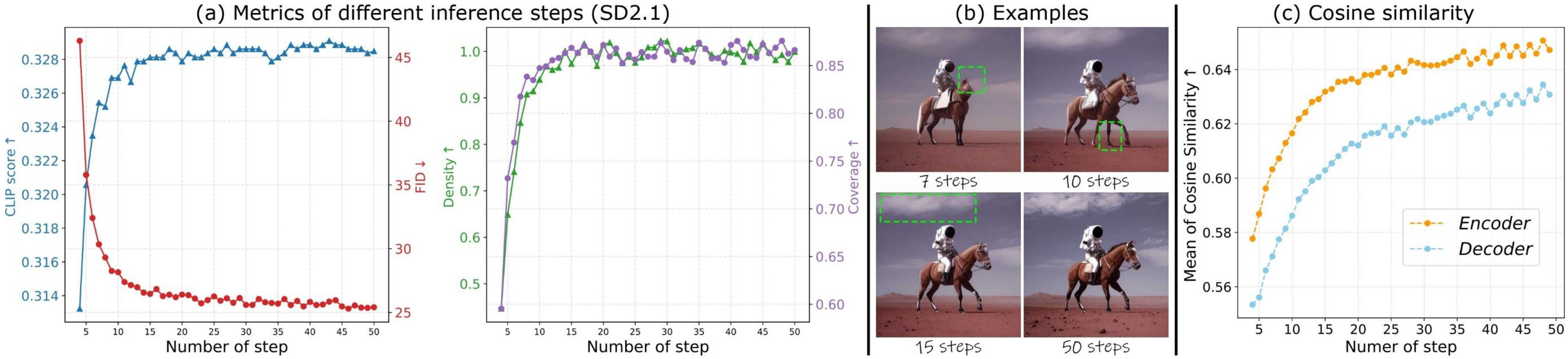
Motivation



- **Above a certain threshold of steps**, such as 15 steps in SD2.1, the model maintains image generation quality (Fig.a-b) while the features show high similarity (Fig.c).
- **Below this threshold**, feature similarity deteriorates along with worse generation quality, accompanied by a degradation in image generation quality as sampling steps reduce.

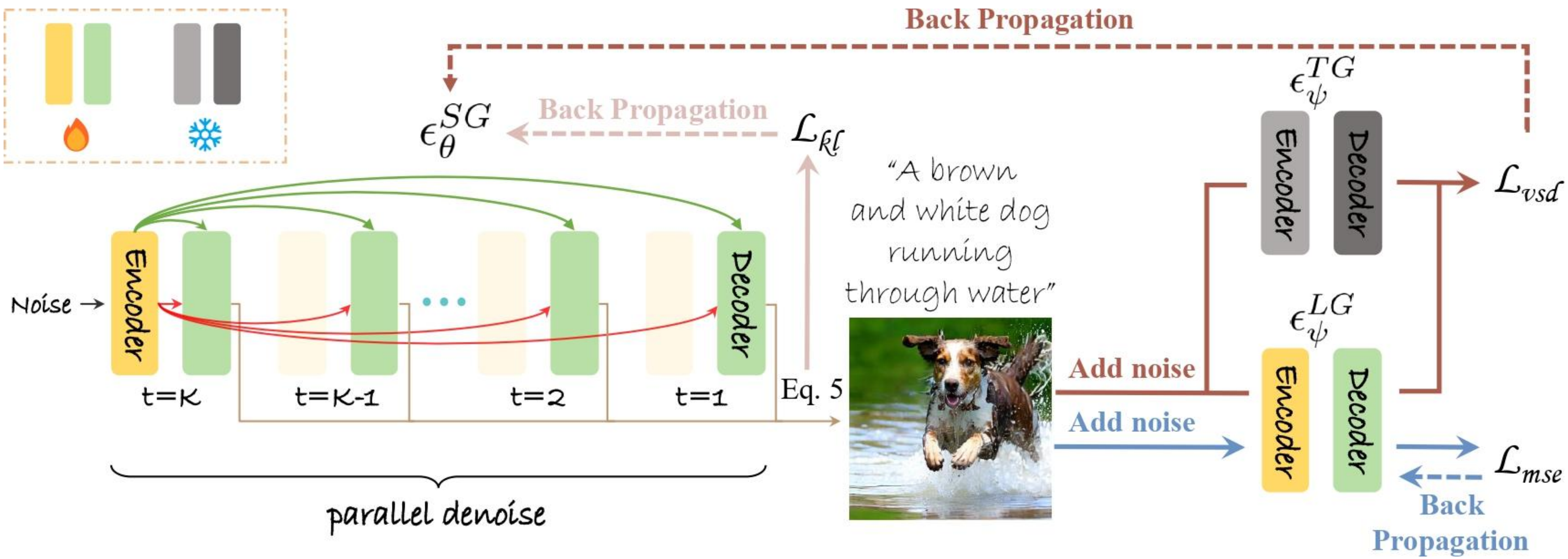


Motivation



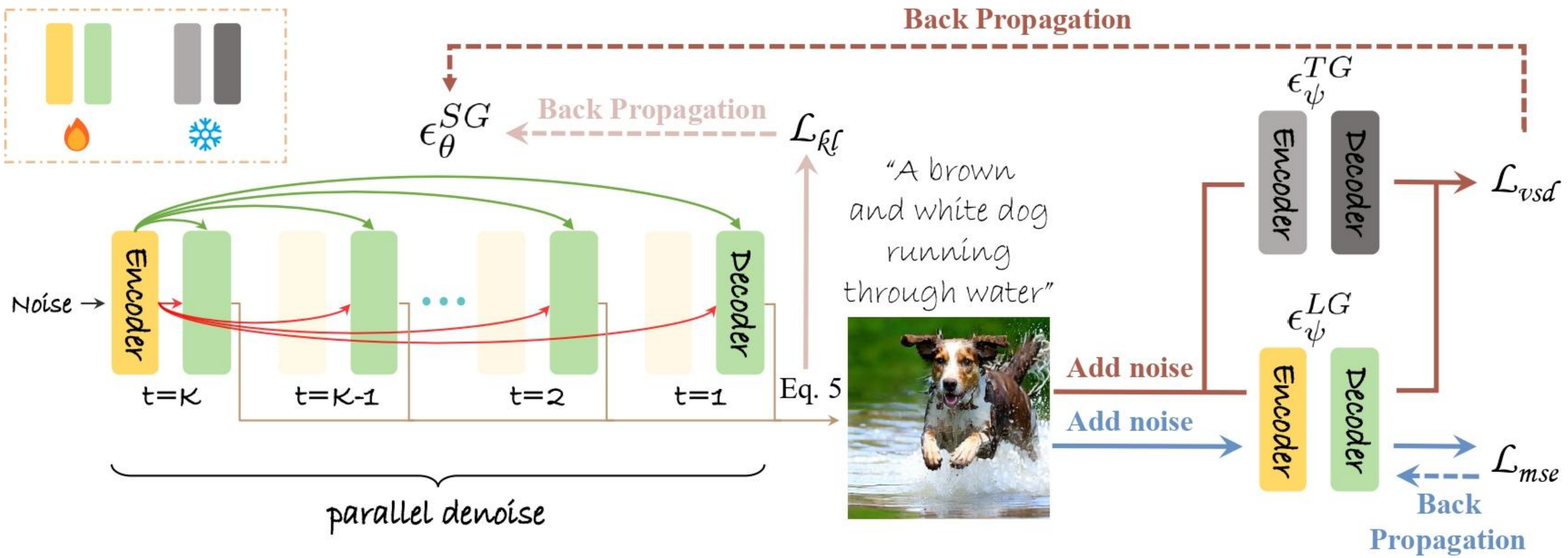
- **Above a certain threshold of steps**, such as 15 steps in SD2.1, the model maintains image generation quality (Fig.a-b) while the features show high similarity (Fig.c).
- **Below this threshold**, feature similarity deteriorates along with worse generation quality, accompanied by a degradation in image generation quality as sampling steps reduce.
- Furthermore, the encoder features consistently exhibit higher similarity than the decoder across all sampling steps (Fig.c).

Method



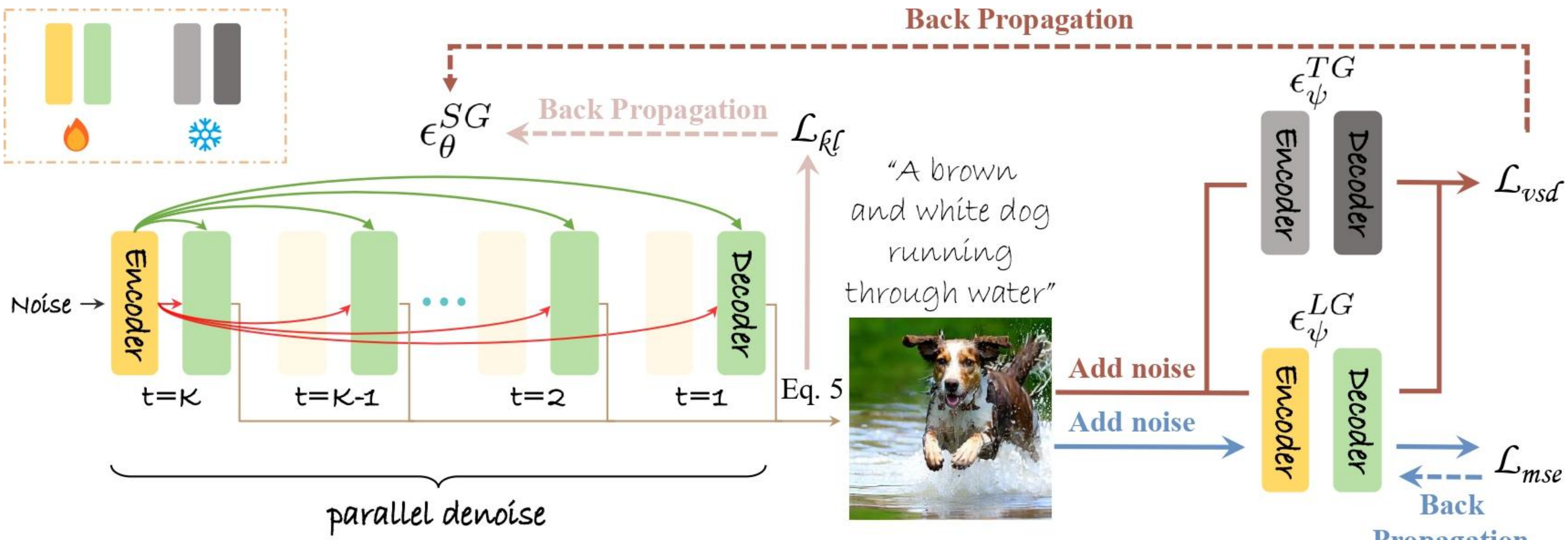
- Hence, we use a novel design with 1-step encoder and a 4-step decoder (Time-independent Unified Encoder architecture), achieving near 1-step inference. Since the 4-step decoder captures richer semantics, ours aligns the generation quality with multi-step DMs.

Method



- Existing 1-step distillation models completely **break away from** the iterative denoising process characteristic of diffusion models.
- Existing 4-step distillation models demonstrate significantly **lower sampling efficiency** compared to 1-step models.

Method

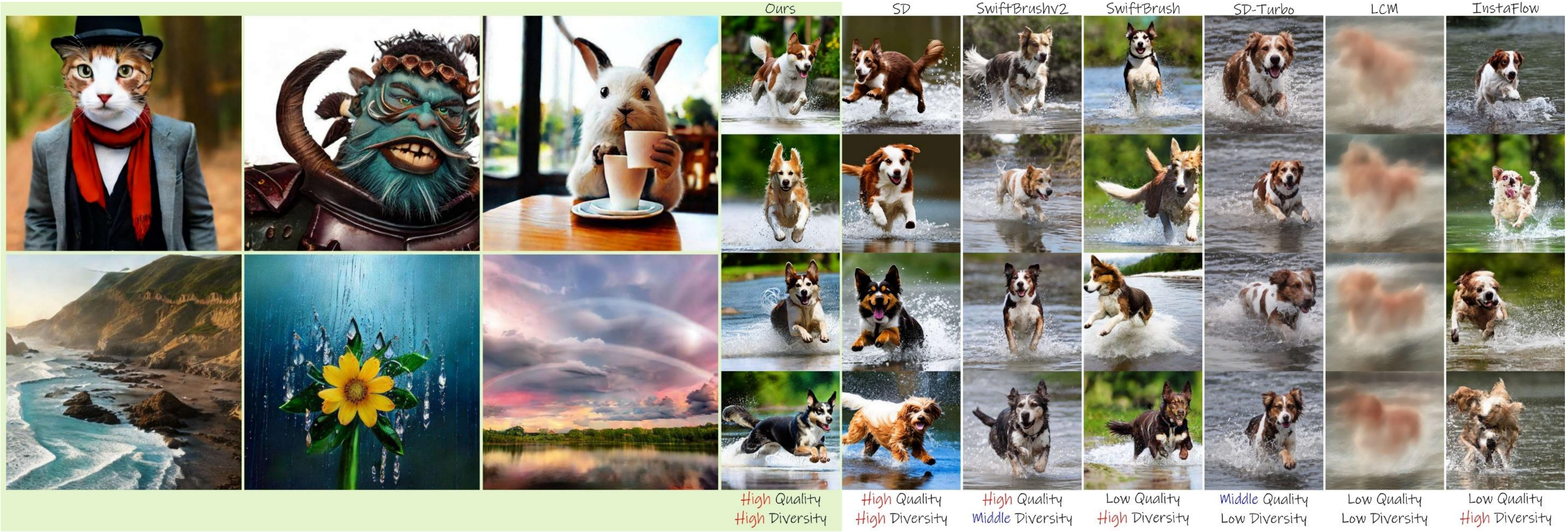


$$\nabla_\theta \mathcal{L}_{VSD} = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\psi(x_t,t,y) - \epsilon_\phi(x_t,t,y))\frac{\partial g(\theta)}{\partial \theta}\right]$$

$$\mathcal{L}_{\mathrm{KL}} = [\mathcal{D}_{\mathrm{KL}}(\epsilon_\theta^{SG}(\epsilon, T_1, t, \boldsymbol{y}) || \mathcal{N}(0, I))],$$

$$\min_{\epsilon_\phi} \mathbb{E}_{t,\epsilon} \underbrace{\|\epsilon_\phi(x_t,t,y) - \epsilon\|_2^2}_{\mathcal{L}_{mse}}$$

Experiments



Experiments

Method	Base Model	Step	Param	COCO2014-30K FID↓	COCO2014-30K CLIP↑	COCO2014-30K Precision↑	COCO2014-30K Recall↑	COCO2014-30K F1↑	COCO2017-5K FID↓	COCO2017-5K CLIP↑	COCO2017-5K Precision↑	COCO2017-5K Recall↑	COCO2017-5K F1↑	Inference↓ Time (ms)	Inference↓ Memory (GB)	Training Data Size↓	Training Data Image Free	A100 Days↓
SD1.5 [58] (*cfg*=7.5)†	–	50	860M	16.08	0.325	0.717	0.527	0.607	23.39	0.326	0.776	0.587	0.668	2503.0	4.04	5B	✗	4783
SD1.5 [58] (*cfg*=4.5)†	–	50	860M	9.90	0.322	0.727	0.585	0.648	19.87	0.323	0.764	0.649	0.702	2503.0	4.04	5B	✗	4783
SD2.1 [58] (*cfg*=7.5)†	–	50	865M	16.10	0.328	0.723	0.489	0.583	25.40	0.328	0.769	0.561	0.649	2244.2	3.89	5B	✗	8332
SD2.1 [58] (*cfg*=4.5)†	–	50	865M	12.22	0.325	0.734	0.526	0.614	22.24	0.298	0.788	0.606	0.685	2244.2	3.89	5B	✗	8332
GigaGAN [26]*	GAN	1	1.0B	9.24	0.325	0.724	0.547	0.623	–	–	–	–	–	–	–	2.7B	✗	6250
InstaFlow [40]†	SD1.5	1	0.9B	<u>13.78</u>	0.288	0.654	<u>0.521</u>	<u>0.580</u>	**19.00**	0.293	0.729	<u>0.613</u>	<u>0.666</u>	111.3	3.99	3.2M	✗	183.2
LCM [44]†	SD1.5	1	860M	132.09	0.230	0.109	0.194	0.140	143.73	0.229	0.118	0.291	0.168	236.2	5.88	12M	✗	1.3
SD-Turbo [64]†	SD2.1	1	865M	19.51	<u>0.331</u>	<u>0.758</u>	0.458	0.571	29.35	<u>0.331</u>	<u>0.786</u>	0.445	0.568	140.0	3.86	unk.	✗	unk.
SwiftBrush [53]†	SD2.1	1	865M	17.20	0.301	0.672	0.458	0.545	27.18	0.314	0.729	0.527	0.612	95.0	3.85	1.4M	✓	4.1
SwiftBrushv2 [7]‡	SD2.1	1	865M	15.98	0.326	**0.782**	0.457	0.577	26.28	0.326	**0.816**	0.543	0.652	139.6	4.91	1.4M	✓	24.1
LCM [44]†	SD1.5	4	860M	23.21	0.262	0.666	0.346	0.455	40.37	0.303	0.713	0.460	0.559	592.3	5.88	12M	✗	1.3
SD-Turbo [64]†	SD2.1	4	865M	16.14	**0.335**	0.633	0.394	0.468	26.14	**0.335**	0.694	0.375	0.487	272.2	3.86	unk.	✗	unk.
Ours	SD2.1	1	865M	**13.09**	0.313	0.634	**0.622**	**0.628**	<u>23.11</u>	0.313	0.697	**0.668**	**0.682**	164.7	4.98	1.4M	✓	3.9

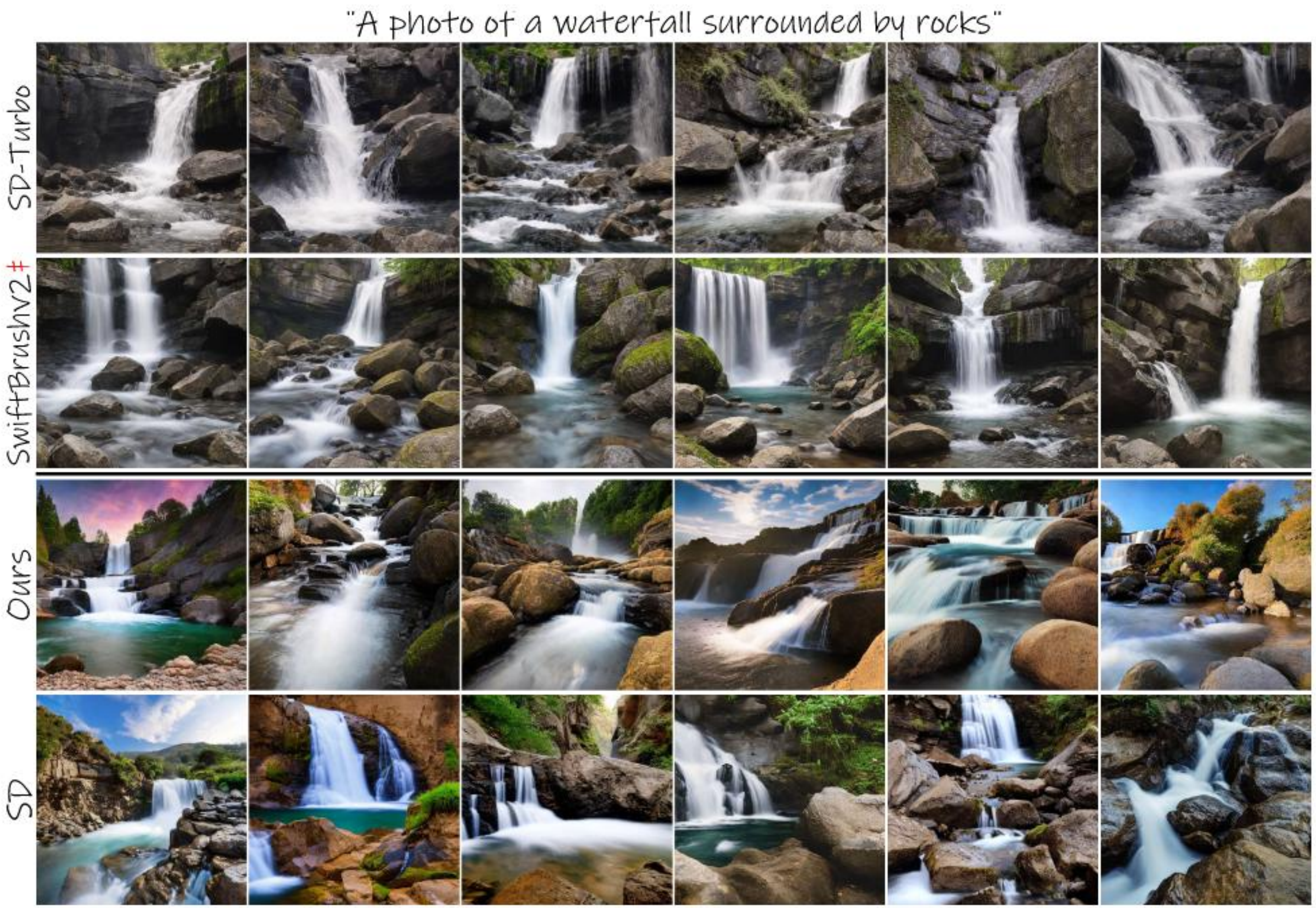
Table 1. Comparison of our distillation method against other works. Inference Time (ms) and Memory (GB). † indicates that we report results using the provided official code and pretrained models. ‡ denotes that we re-implemented the work and are providing the scores. * indicates that we report results using the provided generated images. "unk." denotes unknown. The best and second-best scores are highlighted in **bold** and <u>underlined</u>, respectively, with both the parameter count and training data size being below the billion level.

Experiments

Dataset / Method \ Metrics	Base Model	Step	AFHQ FID↓	AFHQ Density↑	AFHQ Coverage↑	CelebA-HQ FID↓	CelebA-HQ Density↑	CelebA-HQ Coverage↑	DrawBench FID↓	DrawBench Density↑	DrawBench Coverage↑	PartiPrompts FID↓	PartiPrompts Density↑	PartiPrompts Coverage↑	Training Data Image Free
SD1.5 [58] (*cfg*=7.5)†	–	50	47.16	0.066	0.030	93.94	0.053	0.013	11.95	0.510	0.622	7.36	0.730	0.887	✗
SD2.1 [58] (*cfg*=7.5)†	–	50	51.67	0.053	0.022	89.57	0.018	0.013	0	1	1	0	1	1	✗
InstaFlow [40]†	SD1.5	1	**51.97**	0.058	<u>0.029</u>	131.99	0.026	0.007	25.08	0.223	0.337	17.64	0.457	0.670	✗
LCM [44]†	SD1.5	1	155.63	0.012	0.033	165.74	0.001	0.004	120.98	0.058	0.014	95.65	0.095	0.072	✗
SD-Turbo [64]†	SD2.1	1	77.75	**0.142**	0.033	146.22	0.047	0.006	25.75	0.597	0.488	17.40	0.770	0.775	✗
SwiftBrush [53]†	SD2.1	1	67.60	0.039	0.014	144.03	0.014	0.002	21.48	0.402	0.441	<u>14.43</u>	0.579	0.737	✓
SwiftBrushv2 [7]‡	SD2.1	1	64.99	<u>0.110</u>	0.025	131.89	<u>0.055</u>	0.012	**18.57**	<u>0.682</u>	<u>0.597</u>	**11.32**	<u>0.850</u>	**0.865**	✓
LCM [44]†	SD1.5	4	78.00	0.054	0.008	<u>122.44</u>	0.045	<u>0.045</u>	46.23	0.183	0.187	26.84	0.512	0.575	✗
SD-Turbo [64]†	SD2.1	4	77.23	0.011	0.005	193.08	0.013	0.001	27.80	0.281	0.371	22.84	0.500	0.648	✗
Ours	SD2.1	1	<u>54.48</u>	0.068	**0.071**	**116.82**	**0.116**	**0.068**	<u>21.10</u>	**0.685**	**0.616**	16.28	**0.852**	<u>0.840</u>	✓

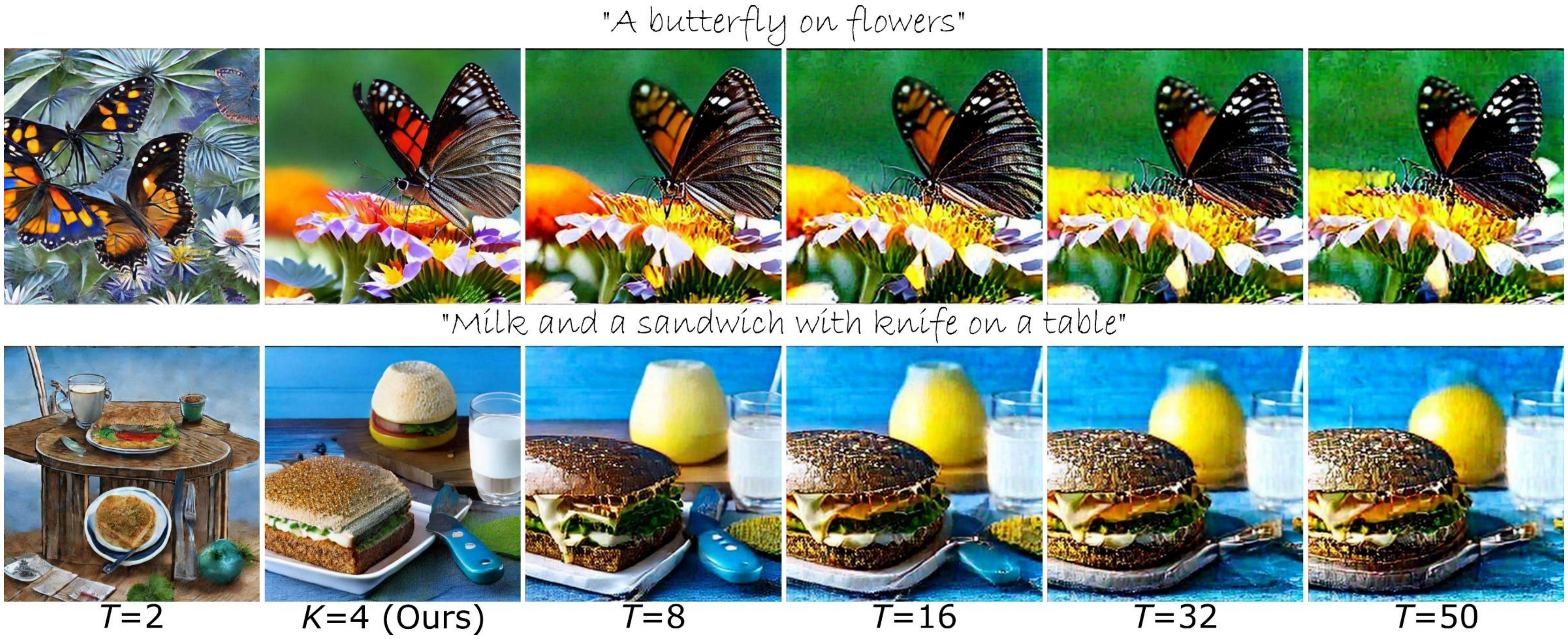
Table 2. Quantitative comparison of our distillation method with other approaches based on FID, Density, and Coverage metrics to assess diversity. † indicates that we report results using the provided official code and pretrained models. ‡ denotes that we re-implemented the work and are providing the scores. The best and second-best numbers are marked with **bold** and <u>underlined</u> respectively.

Experiments



- Both SD-Turbo and SwiftBrushv2 tend to generate results with similar scenery and style when given the same prompt, resulting in a lack of diversity

Experiments



- The well-trained student model with time-steps ***K***=4 (the second column) serves as a good starting point for students set at other time-steps (e.g., 2, 8, 16, 32, and 50)

Take Aways

- Below the step threshold (e.g., 15 steps), image quality degrades as sampling steps decrease
- Encoder features show consistently higher similarity than decoder features across all step settings
- 1-step encoder and a 4-step decoder architecture

Thank you for your attention!
Any Question?

Senmao Li[1], Lei Wang[1], Kai Wang[2], Tao Liu[1], Jiehang Xie[3], Joost van de Weijer[2]
Fahad Shahbaz Khan[4,5], Shiqi Yang[6], Yaxing Wang[1,7#], Jian Yang[1]

[1]VCIP, CS, Nankai University [2]Computer Vision Center, Universitat Auto`noma de Barcelona,
[3]School of Big Data and Computer Science, Guizhou Normal University
[4]Mohamed bin Zayed University of AI [5]Linkoping University [6]SB Intuitions, SoftBank
[7]Nankai International Advanced Research Institute (Shenzhen Futian), Nankai University

https://github.com/sen-mao/Loopfree