Faster Diffusion: Rethinking the Role of the Encoder for Diffusion Model Inference

Senmao Li[1*], Taihang Hu[1*], Joost van de Weijer[2], Fahad Shahbaz Khan[3,4], Tao Liu[1] Linxuan Li[1], Shiqi Yang[5], Yaxing Wang[1#], Ming-Ming Cheng[1], Jian Yang[1]

[1]VCIP, CS, Nankai University, [2]Computer Vision Center, Universitat Autònoma de Barcelona
[3]Mohamed bin Zayed University of AI, [4]Linkoping University, [5]Independent Researcher, Tokyo

{senmaonk, hutaihang00, ltolcy0, linxuanli520, shiqi.yang147.jp}@gmail.com

joost@cvc.uab.es, fahad.khan@liu.se, {yaxing, cmm, csjyang}@nankai.edu.cn

* Equal contribution

The corresponding author

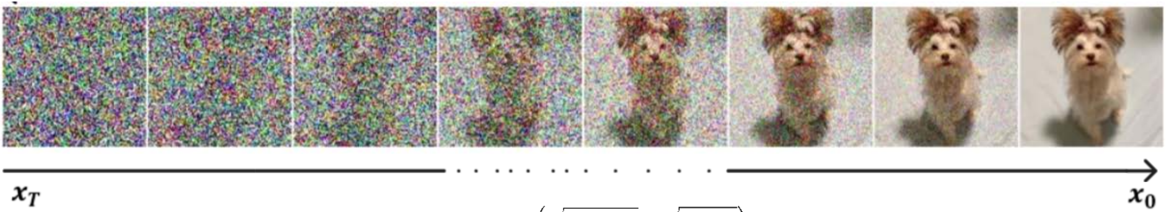
GANs vs. Diffusion Models (2022)

GANs (single-step 0.02s)

DMs (1000-step 37.6s)

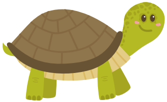
Background

Latent Diffusion Model

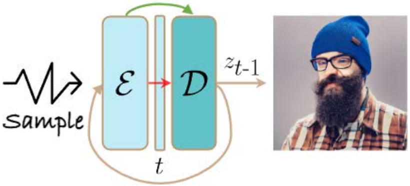


$$\boldsymbol{x}_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}\boldsymbol{x}_t + \sqrt{\alpha_{t-1}}\left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}\right) \cdot \epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$$

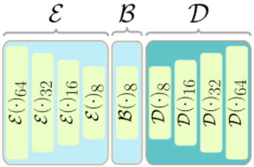
t= {T, ..., 1}, T=1000 or 50



Background



StableDiffusion sampling

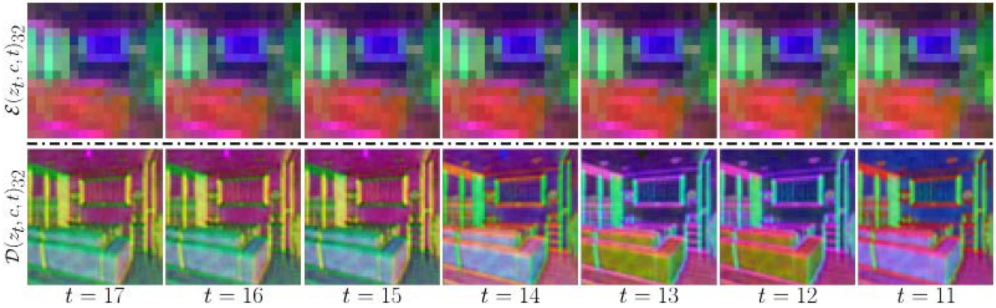


UNet architecture

- ControlNet fine-tunes an additional encoder, and inject features into the decoder.
- PnP performs text-guided image-to-image translation by leveraging the decoder features.
- DIFT finds an emergent correspondence phenomenon that mainly exists in the decoder features.
- ...

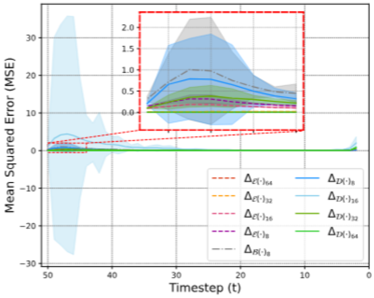
Analysis

The encoder features change minimally and havesimilarities at many time-steps, while the decoder features exhibit substantial variations acrossdifferent time-steps



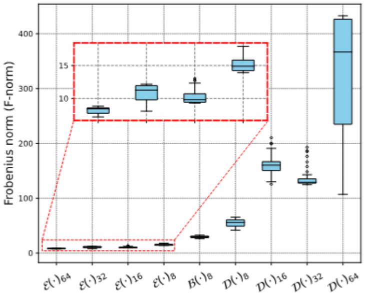
Analysis-StableDiffusion

The encoder features change minimally and havesimilarities at many time-steps, while the de-coder features exhibit substantial variations acrossdifferent time-steps

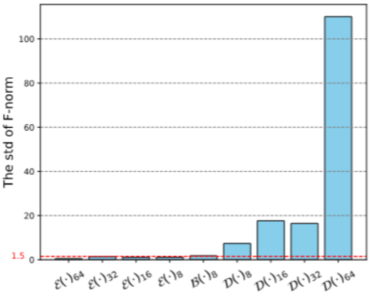


Feature evolving across adjacent time-stepsis measured by MSE.

$$\Delta_{\mathcal{E}(\cdot)_s} = \frac{1}{d \times s^2} \|\mathcal{E}(z_t, c, t)_s - \mathcal{E}(z_{t-1}, c, t-1)_s\|_2^2,$$

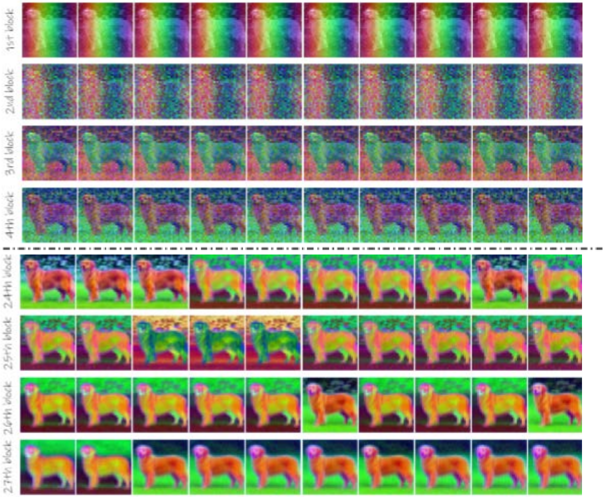


The Frobenius norm of the features of different layers of the UNet



The std of F-norm

Analysis-DiT



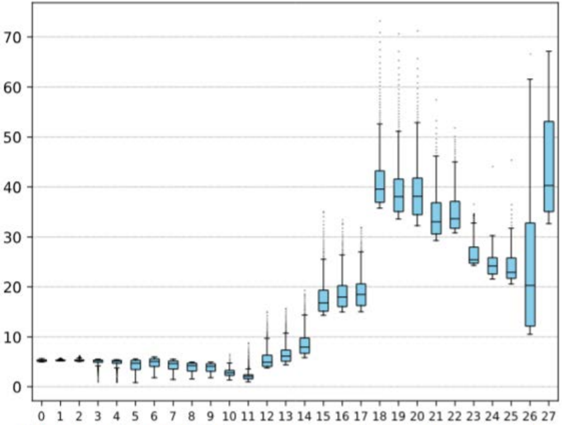
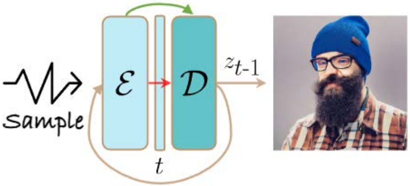
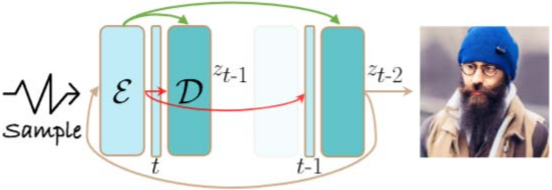


Figure 11: DiT feature statistics (F-norm)

Method



StableDiffusion sampling

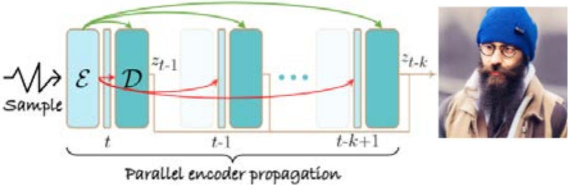


Encoder propagation

Eq. 1: $z_{t-2} = \sqrt{\frac{\alpha_{t-2}}{\alpha_{t-1}}} z_{t-1} +$

$$\sqrt{\alpha_{t-2}}\left(\sqrt{\frac{1}{\alpha_{t-2}}-1} - \sqrt{\frac{1}{\alpha_{t-1}}-1}\right)\cdot \epsilon_\theta(\cancel{z_{t-1}}, t-1, \mathbf{c})$$

Note, at time-step t-1,
predicting noise does not require $z_{t\text{-}1}$



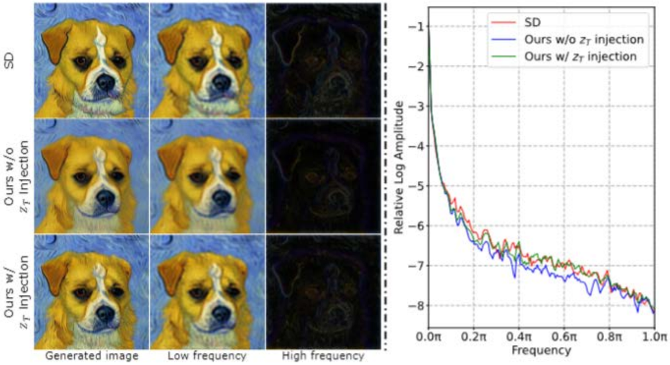
Non-uniform encoder propagation

Method

Comparing with SD (left), encoder prop-agation reduces the sampling time by 24% (mid-dle). Furthermore, parallel encoder propagationachieves a 41% reduction in sampling time (right).

Method

Prior noise injection: The loss of texture information occurs in all frequencies of the frequency domain. This approach ensures a close resemblance of generated results in the frequency domain, with the generated images maintaining the desired fidelity.



Experiments

Table 1: Quantitative evaluation[7] for both SD and DeepFloyd-IF diffusion models.

DM	Sampling Method	T	FID↓	Clip-score↑	GFLOPs/ image↓	s/image ↓ Unet of DM	s/image ↓ DM
Stable Diffusion	DDIM	50	21.75	0.773	37050	2.23	2.42
	DDIM w/ Ours	50	21.08	0.783	**27350** 27%↓	**1.21** 45%↓	**1.42** 41%↓
	DPM-Solver	20	21.36	0.780	14821	0.90	1.14
	DPM-Solver w/ Ours	20	21.25	0.779	**11743** 21%↓	**0.46** 48%↓	**0.64** 43%↓
	DPM-Solver++	20	20.51	0.782	14821	0.90	1.13
	DPM-Solver++ w/ Ours	20	20.76	0.781	**11743** 21%↓	**0.46** 48%↓	**0.64** 43%↓
	DDIM + ToMe	50	22.32	0.782	35123	2.07	2.26
	DDIM + ToMe w/ Ours	50	20.73	0.781	**26053** 26%↓	**1.15** 44%↓	**1.33** 41%↓
DeepFloyd-IF	DDPM	225	23.89	0.783	734825	33.91	34.55
	DDPM w/ Ours	225	23.73	0.782	**626523** 15%↓	**25.61** 25%↓	**26.27** 24%↓
	DPM-Solver++	100	20.79	0.784	370525	15.19	16.09
	DPM-Solver++ w/ Ours	100	20.85	0.785	**313381** 15%↓	**12.02** 21%↓	**12.97** 20%↓

Experiments

Table 3: Quantitative evaluation for DiT.

Sampling Method	T	Image Res.	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑	s/image
DiT	250	256	2.27	4.60	278.24	0.83	0.57	5.13
DiT w/ Ours	250	256	**2.31**	**4.55**	**276.05**	**0.82**	**0.57**	**3.62** 29%↓
DiT	250	512	3.04	5.02	240.82	0.84	0.54	26.25
DiT w/ Ours	250	512	**3.25**	**5.05**	**245.13**	**0.83**	**0.51**	**17.35** 34%↓


DiT 26.25s


DiT w/ Ours 17.35s 34%↓

Experiments

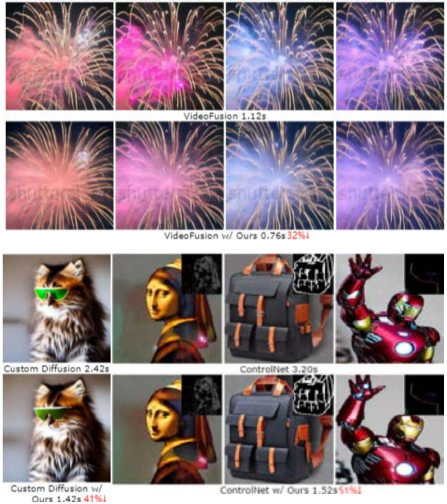


Table 4: Quantitative evaluation on text-to-video, personalized generation and reference-guided generation tasks. † and ‡ indicate “edges” and “scribble” conditions, respectively.

Method	T	FID↓	Clip-score↑	GFLOPs/image↓	s/image↓ Unet of SD	s/image↓ SD
Text2Video-zero	50	-	0.732	39670	12.59/8	13.65/8
Text2Video-zero w/ Ours	50	-	0.731	**30690**$_{22\%\downarrow}$	**9.46/8**$_{25\%\downarrow}$	**10.54/8**$_{23\%\downarrow}$
VideoFusion	50	-	0.700	224700	16.71/16	17.93/16
VideoFusion w/ Ours	50	-	0.700	**148680**$_{33\%\downarrow}$	**11.1/16**$_{34\%\downarrow}$	**12.2/16**$_{32\%\downarrow}$
ControlNet (†)	50	13.78	0.769	49500	3.09	3.20
ControlNet (†) w/ Ours	50	14.65	0.767	**31400**$_{37\%\downarrow}$	**1.43**$_{54\%\downarrow}$	**1.52**$_{51\%\downarrow}$
ControlNet (‡)	50	16.17	0.775	56850	3.85	3.95
ControlNet (‡) w/ Ours	50	16.42	0.775	**35990**$_{37\%\downarrow}$	**1.83**$_{53\%\downarrow}$	**1.93**$_{51\%\downarrow}$
Dreambooth	50	-	0.640	37050	2.23	2.42
Dreambooth w/ Ours	50	-	0.660	**27350**$_{27\%\downarrow}$	**1.21**$_{45\%\downarrow}$	**1.42**$_{41\%\downarrow}$
CustomDiffusion	50	-	0.640	37050	2.21	2.42
CustomDiffusion w/ Ours	50	-	0.650	**27350**$_{27\%\downarrow}$	**1.21**$_{45\%\downarrow}$	**1.42**$_{41\%\downarrow}$

Experiments

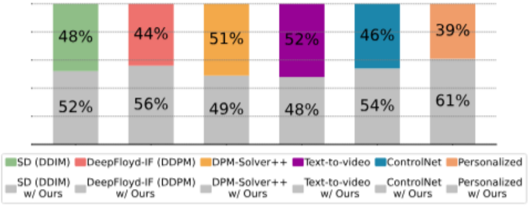


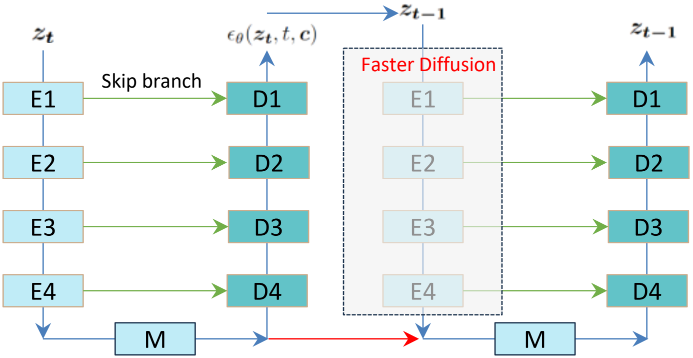
Figure 8: User study results.

Table 6: Quantitative evaluation for prior noise injection.

Sampling Method	SD (DDIM)	SD (DDIM) + Ours w/o z_T injection	SD (DDIM) + Ours w/ z_T injection
FID $\downarrow$	21.75	21.71	21.08
Clipscore $\uparrow$	0.773	0.779	0.783

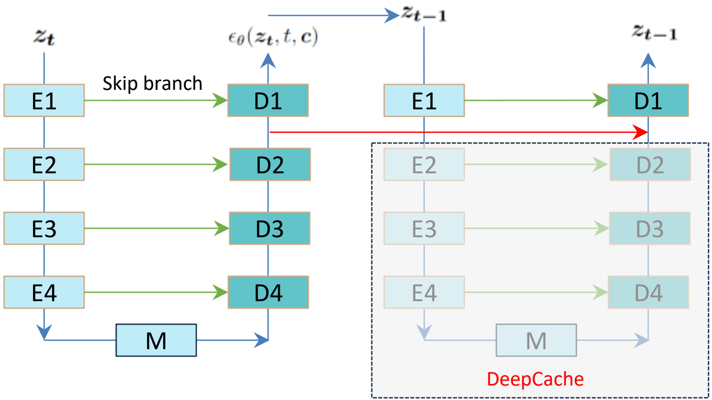
Experiments: FasterDiffusion vs. DeepCache

FasterDiffusion conducts encoder propagation for efficient diffusion sampling, reducing time on both the UNet-based and the transform-based diffusion models

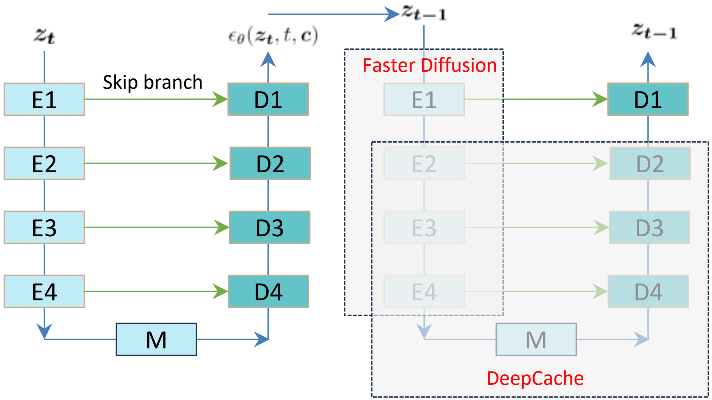


Experiments: FasterDiffusion vs. DeepCache

DeepCache employs the similarity observed in high-level features across adjacent steps of the diffusion model, thereby mitigating the computational.



Experiments: FasterDiffusion vs. DeepCache



Experiments

Table 2: Comparison with DeepCache and CacheMe. CacheMe is not open-source.

Sampling Method	T	Parallel	FID ↓	Clipscore ↑	s/image
DDIM	50	×	21.75	0.773	2.42
DDIM w/ DeepCache	50	×	21.53	0.770	$1.05_{56\%\downarrow}$
DDIM w/ CacheMe	50	×	–	–	$1.30_{44\%\downarrow}$
DDIM w/ Ours	50	✓	21.62	0.775	$0.56_{77\%\downarrow}$

When combined with ControlNet, our inference time Shows a significant advantage compared to DeepCache

	Clipscore↑	FID↓	s/image↓
ContrlNet	0.769	13.78	3.20
ContrlNet w/ DeepCache	0.765	14.18	1.89 (1.69x)
ContrlNet w/ Ours	0.767	14.65	**1.52 (2.10x)**

Thank you for your attention!

Senmao Li[1], Taihang Hu[1], Joost van de Weijer[2], Fahad Shahbaz Khan[3,4], Tao Liu[1] Linxuan Li[1], Shiqi Yang[5], Yaxing Wang[1*], Ming-Ming Cheng[1], Jian Yang[1]

[1]VCIP, CS, Nankai University, [2]Computer Vision Center, Universitat Autònoma de Barcelona
[3]Mohamed bin Zayed University of AI, [4]Linkoping University, [5]Independent Researcher, Tokyo

{senmaonk, hutaihang00, ltolcy0, linxuanli520, shiqi.yang147.jp}@gmail.com

joost@cvc.uab.es, fahad.khan@liu.se, {yaxing, cmm, csjyang}@nankai.edu.cn

∗ The corresponding author

Code: https://github.com/hutaiHang/Faster-Diffusion