A Path to Human-preferred One-step Text-to-image Generative Models

Speaker: Weijian Luo
Final-year PhD student on Statistics and Generative Models
School of Math, Peking University

November 12th, 2024
@Google DeepMind Diffusion Reading Group

• Weijian Luo

I am a final-year PhD student in Statistics and Generative Modeling at School of Maths, **Peking University, China.** I receive my Master Degree on Stats from Peking University and my Bachelor Degree in Math from USTC.

I have a **background of both mathematical statistics and computer sciences**. I am interested in developing large-scale human-preferred text-to-X models. **My series of work have tried to establish the theory and practices of one/few-step text-to-X model**, which includes **Diff-Instruct (NeurIPS23),** faster **SA-Solver** (**NeurIPS23**), **Score-implicit Matching distillation** (**NeurIPS24**), Flow Generator Matching, **Diff-Instruct++ (TMLR under review, first work on HP alignment for one-step models**), **Diff-Instruct* (ICLR24 under review, a leading human-preferred t2i models**), etc. I am also interested in other models such as Consistency Models (ECT, ICLR under review), simulation-free Schrodinger Diffusion Models (ICML24), neural samplers (NeurIPS23), adversarial robustness (NeurIPS23), etc.

Currently, I am invited as a reviewer for Journal of Machine Learning Research (**JMLR**), Nature Communications (**NC**), Transactions in Image Processing (**TIP**), Transactions on Neural Network and Learning Systems (**TNNLS**), Pattern Recognitions (**PR**), NeurIPS, ICLR, ICML, AISTATS, ACM-MM, AAAI.

One-step Diffusion Distillation

What is a Diffusion Model?

The diffusion model is defined with a diffusion process $d\boldsymbol{x}_t = \boldsymbol{F}(\boldsymbol{x}_t, t)\mathrm{d}t + G(t)d\boldsymbol{w}_t,$

The model $s_\phi(x_t, t)$is trained by denoising score-matching along the diffusion process, to match the marginal score functions;

$$\mathcal{L}_{DSM}(\phi) = \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_0 \sim q^{(0)}, \boldsymbol{x}_t | \boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t | \boldsymbol{x}_0)} \| \boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t | \boldsymbol{x}_0) \|_2^2 \mathrm{d}t.$$

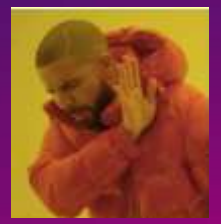
Why Diffusion Distillation?

Usually, sampling from pre-trained DMs requires simulation of solving reversed SDE, which is slow;

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}},$$

Diff-Instruct: A Universal Approach for Transferring Knowledge From Pre-trained Diffusion Models

Weijian Luo[1]*, Tianyang Hu[2]†, Shifeng Zhang[2], Jiacheng Sun[2], Zhenguo Li[2], Zhihua Zhang[1]

Diffusion sampling?  **One-step generator?** 

Diff-Instruct is a **data-free single-step diffusion distillation** method;
It can improve **GAN generators**;
It builds a theory for **text-to-3D generation**, such as DreamFusion and ProlificDreamer;

The diffusion model is defined with a diffusion process $d\boldsymbol{x}_t = \boldsymbol{F}(\boldsymbol{x}_t, t)\mathrm{d}t + G(t)d\boldsymbol{w}_t,$

The model $s_\phi(x_t, t)$is trained by denoising score-matching along the diffusion process, to match the marginal score functions;

$$\mathcal{L}_{DSM}(\phi) = \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_0 \sim q^{(0)}, \boldsymbol{x}_t | \boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t | \boldsymbol{x}_0)} \| \boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t | \boldsymbol{x}_0) \|_2^2 \mathrm{d}t.$$

The problem is: given a pre-trained DM and a one-step generator $\boldsymbol{x} = g_\theta(\boldsymbol{z})$ (p), how can we train the generator to match the distribution of pretrained DM $(q^{(t)})$;
We define the **Integral Kullback-Leibler divergence**

Definition 3.1 (Integral KL divergence). Given a diffusion process (2.1) and a proper weighting function $w(t) > 0$, $t \in [0, T]$, the IKL divergence between two distributions p, q is defined as

$$\mathcal{D}_{IKL}^{[0,T]}(p, q) := \int_{t=0}^{T} w(t)\mathcal{D}_{KL}(p^{(t)}, q^{(t)})\mathrm{d}t := \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_t \sim p^{(t)}} \left[\log \frac{p^{(t)}(\boldsymbol{x}_t)}{q^{(t)}(\boldsymbol{x}_t)}\right] \mathrm{d}t, \qquad (3.1)$$

The goal is to **minimize the IKL divergence between the generator distribution and the DM's distribution**;

We propose a non-trivial parameter-gradient formula to minimize the IKL:

Theorem 3.3. The gradient of the IKL in (3.1) between $p^{(0)}$ and $q^{(0)}$ is

$$\text{Grad}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left[\boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \boldsymbol{s}_{q^{(t)}}(\boldsymbol{x}_t)\right] \frac{\partial \boldsymbol{x}_t}{\partial \theta} \mathrm{d}t. \tag{3.2}$$

The Diff-Instruct algorithm consists of two phases: fine-tuning a new diffusion model and updating the generator;

Algorithm 1: Diff-Instruct Algorithm

Input: pre-trained DM $\boldsymbol{s}_{q^{(t)}}$, generator g_θ, prior distribution p_z, DM $\boldsymbol{s}_\phi$; forward diffusion (2.1).

while *not converge* **do**

update ϕ using SGD with gradient

$$\text{Grad}(\phi) = \frac{\partial}{\partial \phi} \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \|\boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2 \mathrm{d}t.$$

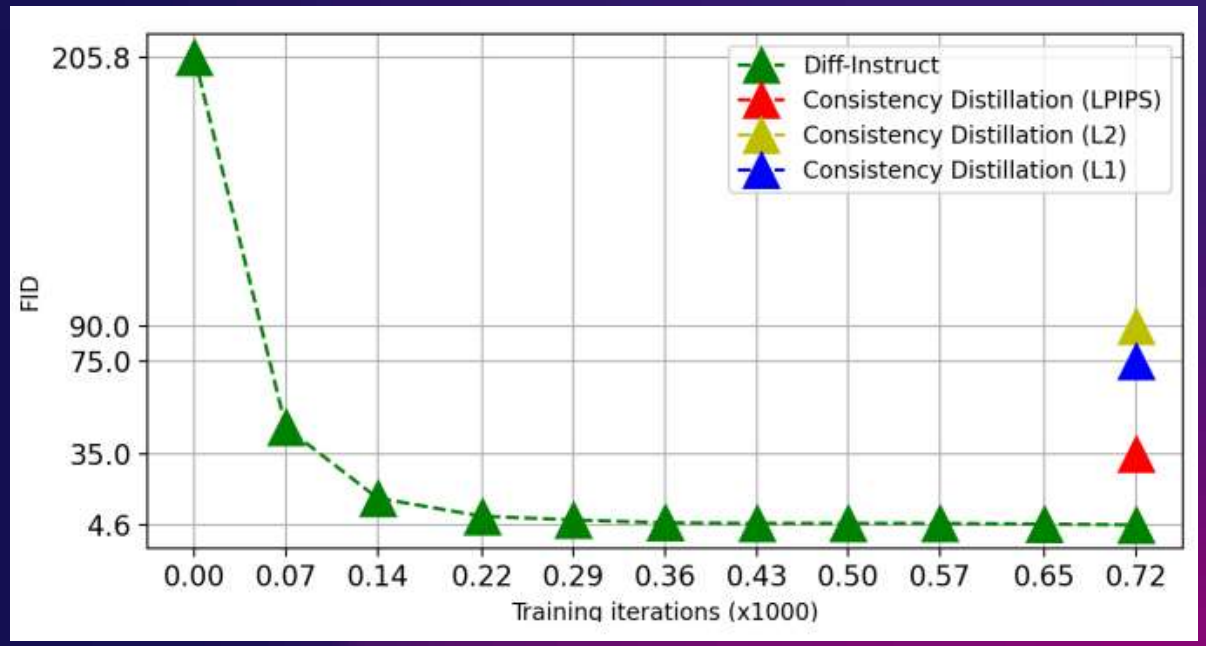
update θ using SGD with the gradient

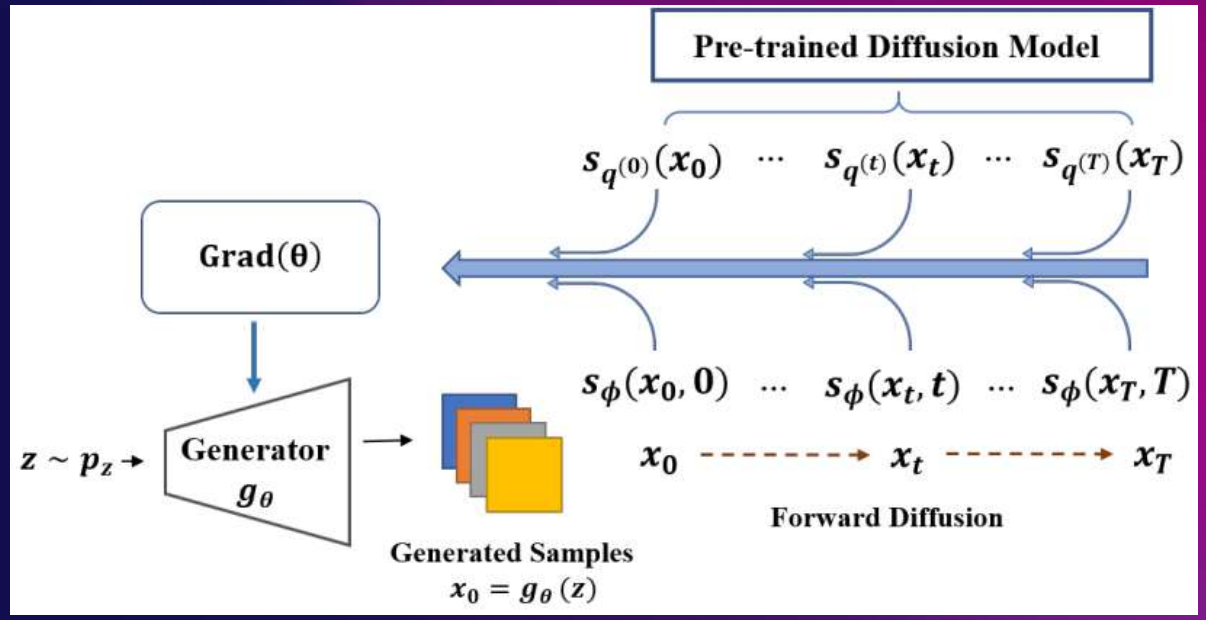
$$\text{Grad}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left[\boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \boldsymbol{s}_{q^{(t)}}(\boldsymbol{x}_t)\right] \frac{\partial \boldsymbol{x}_t}{\partial \theta} \mathrm{d}t.$$

end

return θ, ϕ.

Diff-Instruct outperformed PD and CD on one-step generation on ImageNet64:





Class-conditional ImageNet 64 × 64. †Distillation techniques.

METHOD	NFE (↓)	FID (↓)
Multiple Steps		
ADM [13]	250	**2.07**
SN-DDIM [5]	100	17.53
EDM [34]	79	2.44
EDM-Heun[34]	10	17.25
GGDM [72]	25	18.4
CT [68]	2	11.1
PD† [63]	2	8.95
CD† [68]	2	4.70
Single Steps		
EDM[34]	1	154.78
PD† [63]	1	15.39
CT [68]	1	13.00
CD-L2† [68]	1	12.10
CD-LPIPS† [68]	1	6.20
Diff-Instruct	1	**5.57**

DreamFusion is **a special Case of Diff-Instruct** with Single-point Generator

Theorem 3.3. The gradient of the IKL in (3.1) between $q^{(0)}$ and $p^{(0)}$ is

$$\mathrm{Grad}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left[\boldsymbol{s}_\phi(\boldsymbol{x}_t, t) - \boldsymbol{s}_{p^{(t)}}(\boldsymbol{x}_t)\right] \frac{\partial \boldsymbol{x}_t}{\partial \theta} \mathrm{d}t. \tag{3.2}$$

Corollary 3.4. If the generator's output is a Dirac's Delta distribution with learnable parameters, i.e. $q(\boldsymbol{x}_0) = \delta_{g(\theta)}(\boldsymbol{x}_0)$ [3]. Then the gradient formula (3.2) becomes

$$\mathrm{Grad}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{x}_0 = g(\theta), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left[\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) - \boldsymbol{s}_{p^{(t)}}(\boldsymbol{x}_t)\right] \frac{\partial \boldsymbol{x}_t}{\partial \theta} \mathrm{d}t. \tag{3.3}$$

One-Step Diffusion Distillation through Score Implicit Matching

Weijian Luo,*
Peking University
luoweijian@stu.pku.edu.cn

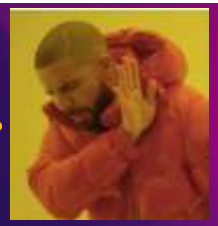
Zemin Huang,
Westlake University
huangzemin@westlake.edu.cn

Zhengyang Geng,
Carnegie Mellon University
zgeng2@cs.cmu.edu

J. Zico Kolter,
Carnegie Mellon University
zkolter@cs.cmu.edu

Guo-jun Qi†
Westlake University
guojunq@gmail.com

https://github.com/maple-research-lab/SIM

IKL? Nonono.  **General Score-based divergence? Yes!!!** 

SIM results in a strong **DiT-based One-step** Text-to-image model;

"Seasoned fisherman portrait, weathered skin etched with deep wrinkles, white beard, piercing gaze beneath a fisherman's hat, softly blurred dock background accentuating rugged features, captured under natural light,

"A Shiba Inu dog wearing a beret and black turtleneck."

The general Score-based divergence:

$$\mathcal{L}(\theta) = \mathcal{D}^{[0,T]}(p_\theta, q) = \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_t\sim\pi_t}\left[\mathbf{d}(\boldsymbol{s}_{p_{\theta,t}}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t))\right]\mathrm{d}t, \tag{3.4}$$

If we take theta gradient, we have an intractable objective:

$$\frac{\partial}{\partial\theta}\mathcal{L}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_t\sim\pi_t}\left[\mathbf{d}'(\boldsymbol{s}_{p_{\theta,t}}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t))\frac{\partial}{\partial\theta}\boldsymbol{s}_{p_{\theta,t}(\boldsymbol{x}_t)}\right]\mathrm{d}t, \tag{3.5}$$

With our gradient trade-off theorem, we can equivalent this intractable gradient to tractable losses:

Theorem 3.1 (Score-divergence gradient Theorem)**.** If distribution $p_{\theta,t}$ satisfies some mild regularity conditions, we have for any score function $\boldsymbol{s}_{q_t}(.)$, the following equation holds for all parameter θ:

$$\mathbb{E}_{\boldsymbol{x}_t\sim p_{\mathrm{sg}[\theta],t}}\left[\mathbf{d}'(\boldsymbol{s}_{p_\theta,t}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t))\frac{\partial}{\partial\theta}\boldsymbol{s}_{p_{\theta,t}(\boldsymbol{x}_t)}\right] \tag{3.6}$$

$$= -\frac{\partial}{\partial\theta}\mathbb{E}_{\substack{\boldsymbol{x}_0\sim p_{\theta,0},\\ \boldsymbol{x}_t|\boldsymbol{x}_0\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\left[\left(\mathbf{d}'(\boldsymbol{s}_{p_{\mathrm{sg}[\theta],t}}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t))\left(\boldsymbol{s}_{p_{\mathrm{sg}[\theta],t}}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right)\right)\right].$$

The tractable loss for distillation:

$$\mathcal{L}_{SIM}(\theta) = \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \Big\{ -\mathbf{d}'(\boldsymbol{y}_t) \Big\}^T \Big\{ \boldsymbol{s}_{p_{\mathrm{sg}[\theta]},t}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0) \Big\} \mathrm{d}t \quad (3.7)$$

The algorithm:

Algorithm 1: Score Implicit Matching for Diffusion Distillation. (Pseudo-code in Appendix A.2)

Input: pre-trained DM $\boldsymbol{s}_{q_t}(.)$, generator g_θ, prior distribution p_z, online DM $\boldsymbol{s}_\psi(.)$; differentiable distance function $\mathbf{d}(.)$, and forward diffusion (2.1).

while *not converge* **do**

with frozen θ, update ψ using SGD with gradient

$$\mathrm{Grad}(\psi) = \frac{\partial}{\partial \psi} \int_{t=0}^{T} \lambda(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \|\boldsymbol{s}_\psi(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2 \mathrm{d}t.$$

with frozen ψ, update θ using SGD with the gradient

$$\mathrm{Grad}(\theta) = \frac{\partial}{\partial \theta} \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \Big\{ -\mathbf{d}'(\boldsymbol{y}_t) \Big\}^T \Big\{ \boldsymbol{s}_\psi(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0) \Big\} \mathrm{d}t,$$

where $\boldsymbol{y}_t := \boldsymbol{s}_\psi(\boldsymbol{x}_t, t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t)$.

end

return θ, ψ.

Table 1: Unconditional sample quality on CIFAR-10. † means method we reproduced.

METHOD	NFE (↓)	FID (↓)
DIFFERENT ARCHITECTURE AS EDM MODEL		
DDPM [16]	1000	3.17
DD-GAN(T=2) [65]	2	4.08
KD [34]	1	9.36
TDPM [77]	1	8.91
DFNO [75]	1	4.12
3-REFLOW (+DISTILL) [32]	1	5.21
STYLEGAN2-ADA [19]	1	2.92
STYLEGAN2-ADA+DI [36]	1	2.71
SAME ARCHITECTURE AS EDM[21] MODEL		
EDM [21]	35	1.97
EDM [21]	15	5.62
PD [51]	2	5.13
CD [58]	2	2.93
GET-ONESTEP [11]	1	6.91
CT [58]	1	8.70
ICT-DEEP [56]	2	2.24
DIFF-INSTRUCT [36]	1	4.53
DMD [70]	1	3.77
CTM [24]	1	1.98
CTM[24]	2	**1.87**
SID ($\alpha = 1.0$) [79]	1	1.92
SID ($\alpha = 1.2$)[79]	1	2.02
DI†	1	3.70
SID† ($\alpha = 1.0$)	1	2.44
SIM (OURS)	1	2.17

Table 2: Class-conditional sample quality on CIFAR10 dataset. † means method we reproduced.

METHOD	NFE (↓)	FID (↓)
DIFFERENT ARCHITECTURE AS EDM MODEL		
BIGGAN [3]	1	14.73
BIGGAN+TUNE[3]	1	8.47
STYLEGAN2 [20]	1	6.96
MULTIHINGE [22]	1	6.40
FQ-GAN [74]	1	5.59
STYLEGAN2-ADA [19]	1	2.42
STYLEGAN2-ADA+DI [36]	1	2.27
STYLEGAN2 + SMART [64]	1	2.06
STYLEGAN-XL [53]	1	1.85
SAME ARCHITECTURE AS EDM[21] MODEL		
EDM [21]	35	1.82
EDM [21]	20	2.54
EDM [21]	10	15.56
EDM [21]	1	314.81
GET-ONESTEP [11]	1	6.25
DIFF-INSTRUCT [36]	1	4.19
DMD (W.O. REG) [70]	1	5.58
DMD (W.O. KL) [70]	1	3.82
DMD [70]	1	2.66
CTM [24]	1	1.73
CTM[24]	2	**1.63**
SID ($\alpha = 1.0$) [79]	1	1.93
SID ($\alpha = 1.2$)[79]	1	1.71
SID† ($\alpha = 1.0$)	1	2.34
SIM (OURS)	1	1.96

Model	Steps	Type	Params	AES Score	Image Reward	Pick Score	User Pref	Distill Cost
SD15-Base [48]	25	UNet	860 M	5.26	0.18	0.217		
SD15-LCM [35]	4	UNet	860 M	5.66	-0.37	0.212		8 A100× 4 Days
SD15-TCD [78]	4	UNet	860 M	5.45	-0.15	0.214		8 A800× 5.8 Days
PeRFlow [69]	4	UNet	860 M	5.64	-0.35	0.208		M GPU× N Days
Hyper-SD15[47]	1	UNet	860 M	5.79	0.29	0.215		32 A100× N Days
SDXL-Base [48]	25	UNet	2.6 B	5.54	0.87	0.229		
SDXL-LCM [35]	4	UNet	2.6 B	5.42	0.48	0.224		8 A100× 4 Days
SDXL-TCD [78]	4	UNet	2.6 B	5.42	0.67	0.226		8 A800× 5.8 Days
SDXL-Lightning [30]	4	UNet	2.6 B	5.63	0.72	0.229		64 A100× N Days
Hyper-SDXL[47]	4	UNet	2.6 B	5.74	0.93	0.232		32 A100× N Days
SDXL-Turbo [54]	1	UNet	2.6 B	5.33	0.78	0.228		M GPU× N Days
SDXL-Lightning [30]	1	UNet	2.6 B	5.34	0.54	0.223		64 A100× N Days
Hyper-SDXL[47]	1	UNet	2.6 B	5.85	1.19	0.231		32 A100× N Days
PixArt-α[6]	30	DiT	610 M	5.97	0.82	0.226		
SIM-DiT-600M	1	DiT	610 M	6.42	0.67	0.223		4 A100× 2 days
PixArt-α^* [6]	30	DiT	610 M	5.93	0.53	0.223	54.88%	
SIM-DiT-600M*	1	DiT	610 M	5.91	0.44	0.223	45.12%	4 A100× 2 days

Table 3: Quantitative comparisons with frontier text-to-image models on COCO-2017 validation dataset. The user preference is the winning rate of our user study on SIM-DiT-600M against 20-step PixelArt-α. $*$ means the results evaluated on the SAM-LLaVA-Caption10M dataset, and SIM-DiT-600M means the SIM generator distilled from PixelArt-α-600M, excluding those in the T5 text encoder. The distillation cost *M GPU× N Days* means the model did not report the cost.

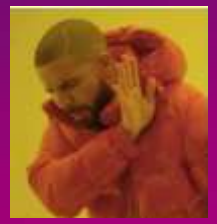
"Drone view of waves crashing against the rugged cliffs along Big Sur's garay point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore. A small island with a lighthouse sits in the distance, and green shrubbery covers the cliff's edge. The steep drop from the road down to the beach is a dramatic feat, with the cliff's edges jutting out over the sea. This is a view that captures the raw beauty of the coast and the rugged landscape of the Pacific Coast Highway."

SCORE-IMPLICIT MATCHING

Preference-alignment for One-step Models

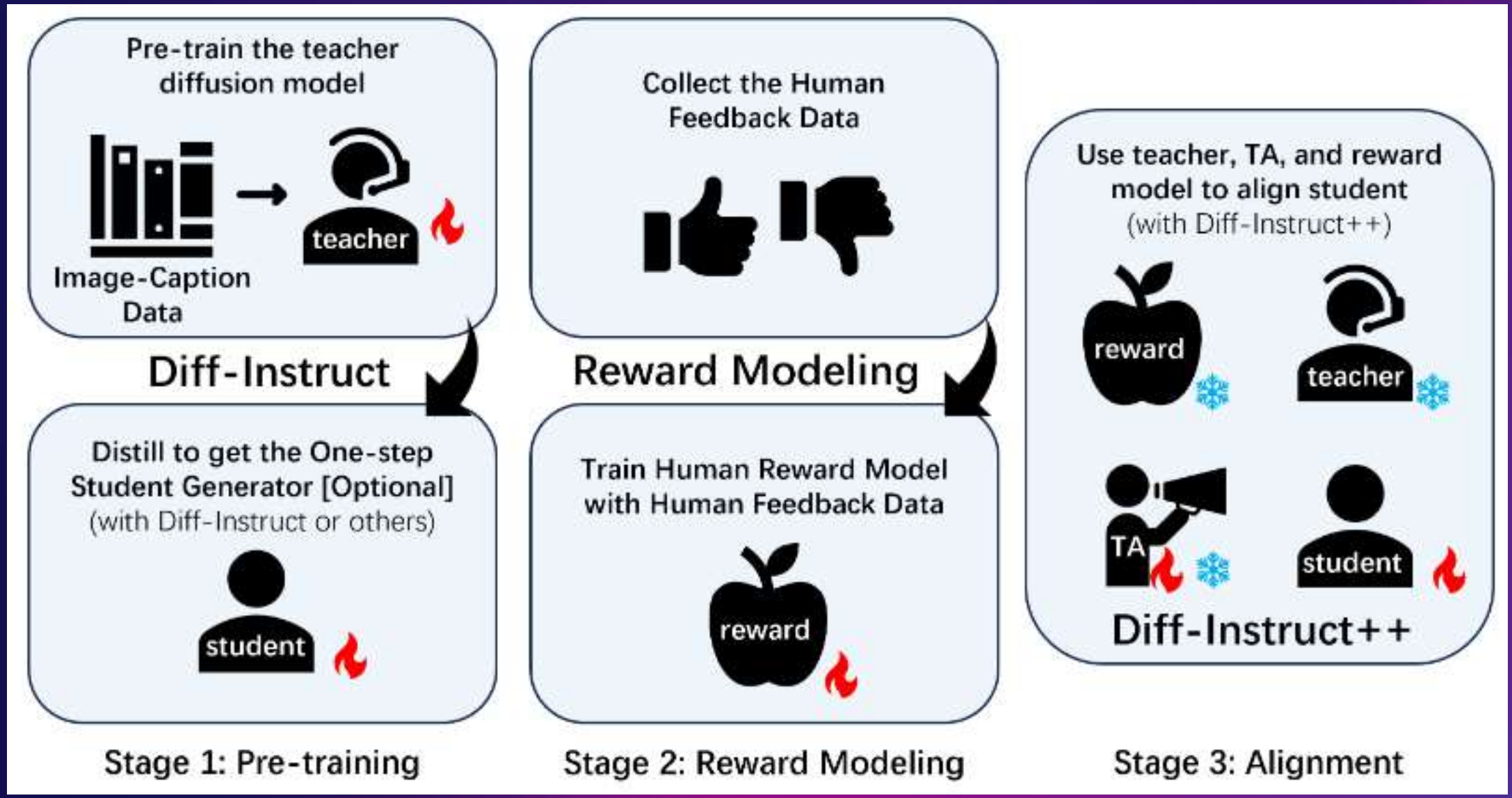
Diff-Instruct++: Training One-step Text-to-image Generator Model to Align with Human Preferences

Weijian Luo*
Peking University
luoweijian@stu.pku.edu.cn.



One-step T2I model? No.
One-step and human-preferred T2I model? Yes!

The first attempt on human-preference alignment for one-step T2I model with RLHF;


Pre-train the teacher diffusion model
Image-Caption Data
teacher
Diff-Instruct
Distill to get the One-step Student Generator [Optional]
(with Diff-Instruct or others)
student
Stage 1: Pre-training
Collect the Human Feedback Data
Reward Modeling
Train Human Reward Model with Human Feedback Data
reward
Stage 2: Reward Modeling
Use teacher, TA, and reward model to align student
(with Diff-Instruct++)
reward
teacher
TA
student
Diff-Instruct++
Stage 3: Alignment

How to do human-preference alignment for one-step model? Online PPO!

$$\boldsymbol{x} = g_\theta(\boldsymbol{z}|\boldsymbol{c}).$$

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\\ \boldsymbol{x}\sim p_\theta(\boldsymbol{x}|\boldsymbol{c})}}\big[-r(\boldsymbol{x},\boldsymbol{c})\big] + \beta\mathcal{D}_{KL}(p_\theta(\boldsymbol{x}|\boldsymbol{c}), p_{ref}(\boldsymbol{x}|\boldsymbol{c}))$$

Theories to overcome gradient issues:

Theorem 3.1. The θ gradient of the objective (3.1) is

$$\mathrm{Grad}(\theta) = \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,\\ \boldsymbol{x}=g_\theta(\boldsymbol{z}|\boldsymbol{c})}}\Big\{-\nabla_{\boldsymbol{x}} r(\boldsymbol{x},\boldsymbol{c}) + \beta\big[\nabla_{\boldsymbol{x}}\log p_\theta(\boldsymbol{x}|\boldsymbol{c}) - \nabla_{\boldsymbol{x}}\log p_{ref}(\boldsymbol{x}|\boldsymbol{c})\big]\Big\}\frac{\partial \boldsymbol{x}}{\partial\theta} \tag{3.2}$$

Classifier-free guidance is **secretly doing Reinforcement Learning using Human-feedback** (RLHF).

$$r(\boldsymbol{x}_0,\boldsymbol{c}) = \int_{t=0}^{T} w(t)\log\frac{p_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c})}{p_{ref}(\boldsymbol{x}_t|t)}\mathrm{d}t. \tag{3.8}$$

Theorem 3.3. Under mild conditions, if we set an implicit reward function as (3.8), the gradient formula (3.7) in Theorem 3.2 with have an explicit expression:

$$\mathrm{Grad}(\theta) = \mathbb{E}_{\substack{\boldsymbol{c},t,\boldsymbol{z}\sim p_z,\boldsymbol{x}_0=g_\theta(\boldsymbol{z}|\boldsymbol{c})\\ \boldsymbol{x}_t|\boldsymbol{x}_0\sim p(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\beta w(t)\Big\{\boldsymbol{s}_\theta(\boldsymbol{x}_t|t,\boldsymbol{c}) - \widetilde{\boldsymbol{s}}^{\beta}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c})\Big\}\frac{\partial \boldsymbol{x}_t}{\partial\theta} \tag{3.9}$$

$$\widetilde{\boldsymbol{s}}^{\beta}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c}) = \boldsymbol{s}_{ref}(\boldsymbol{x}_t|t) + (1+\frac{1}{\beta})\big[\boldsymbol{s}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c}) - \boldsymbol{s}_{ref}(\boldsymbol{x}_t|t)\big]$$

The algorithm:

Algorithm 1: Diff-Instruct++ for aligning generator model with human feedback reward.

Input: prompt dataset $\mathcal{C}$, generator $g_\theta(\boldsymbol{x}_0|\boldsymbol{z},\boldsymbol{c})$, prior distribution p_z, reward model $r(\boldsymbol{x},\boldsymbol{c})$, reward scale α_{rew}, CFG scale α_{cfg}, reference diffusion model $\boldsymbol{s}_{ref}(\boldsymbol{x}_t|c,\boldsymbol{c})$, TA diffusion $\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c})$, forward diffusion $p(\boldsymbol{x}_t|\boldsymbol{x}_0)$ (2.1), TA diffusion updates rounds K_{TA}, time distribution $\pi(t)$, diffusion model weighting $\lambda(t)$, generator IKL loss weighting $w(t)$.

while *not converge* **do**

fix θ, update ψ for K_{TA} rounds by minimizing

$$\mathcal{L}(\psi) = \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,t\sim\pi(t)\\ \boldsymbol{x}_0=g_\theta(\boldsymbol{z}|\boldsymbol{c}),\boldsymbol{x}_t|\boldsymbol{x}_0\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \lambda(t)\|\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2 \mathrm{d}t.$$

update θ using StaD with the gradient

$$\mathrm{Grad}(\theta) = \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,\\ \boldsymbol{x}_0=g_\theta(\boldsymbol{z},\boldsymbol{c})}}[-\alpha_{rew}\nabla_{\boldsymbol{x}_0} r(\boldsymbol{x}_0,\boldsymbol{c})] \tag{3.3}$$

$$+\int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,t\sim\pi(t)\\ \boldsymbol{x}_0=g_\theta(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{x}_t|\boldsymbol{x}_0\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left\{\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c}) - \widetilde{\boldsymbol{s}}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c})\right\}\frac{\partial \boldsymbol{x}_t}{\partial\theta}\mathrm{d}t. \tag{3.4}$$

$$\widetilde{\boldsymbol{s}}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c}) = \boldsymbol{s}_{ref}(\boldsymbol{x}_t|t,\varnothing) + \alpha_{cfg}\left[\boldsymbol{s}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c}) - \boldsymbol{s}_{ref}(\boldsymbol{x}_t|t,\varnothing)\right] \tag{3.5}$$

end

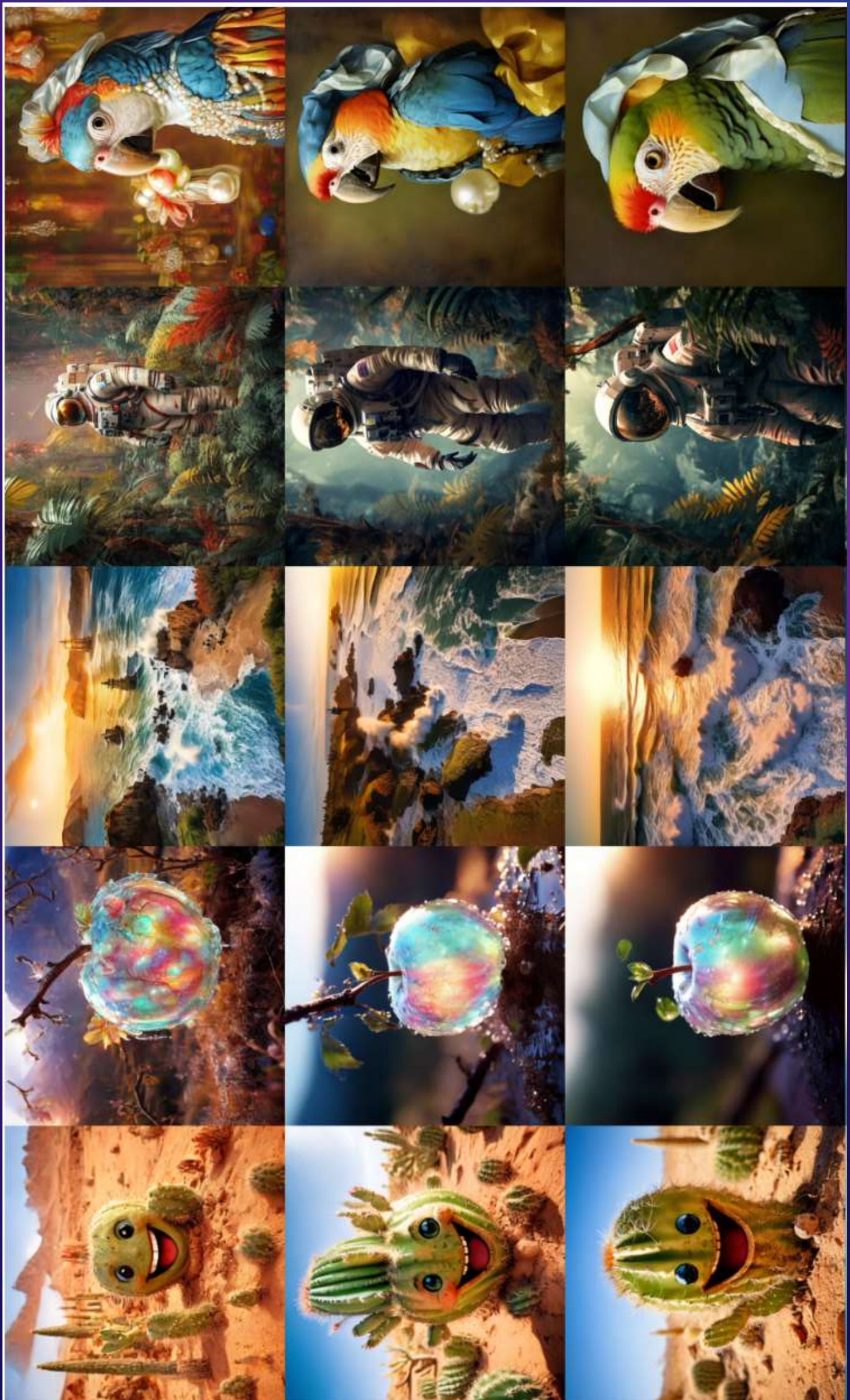
return θ,ψ.

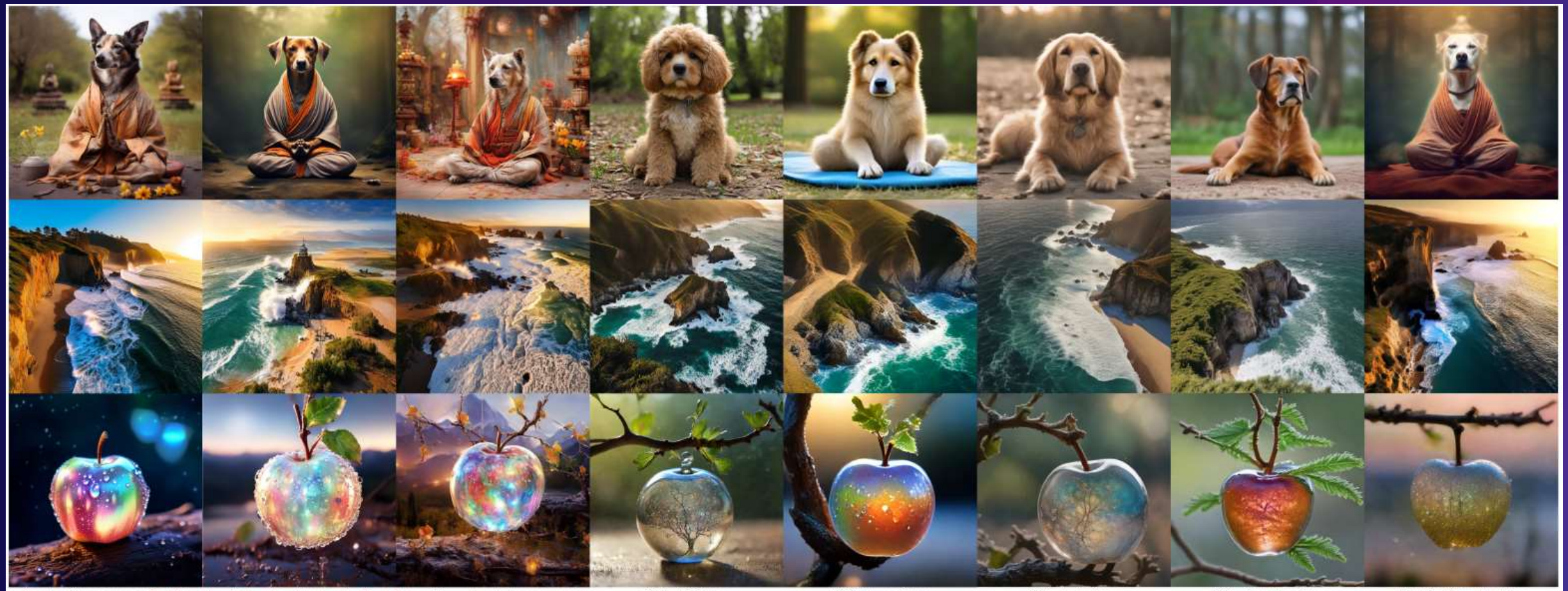
Table 2: Quantitative comparisons of text-to-image models on 1k MSCOCO-2017 validation prompts. DI++ is short for Diff-Instruct++. α_r and α_c are short for α_{rew} and α_{cfg} in Algorithm 1. † means our implementation. Data means the model needs image data for training. Sampling means the model needs to draw samples from reference diffusion models. Reward means the model needs a human reward model for training.

Model	Steps	Type	Params	Image Reward	Aes Score	Pick Score	CLIP Score
SD15-Base(Rombach et al., 2022)	15	UNet	0.86B	0.08	5.25	0.212	30.99
SD15-Base(Rombach et al., 2022)	25	UNet	0.86B	0.22	5.32	0.216	31.13
SD15-DPO(Wallace et al., 2024)	15	UNet	0.86B	0.20	5.29	0.214	31.07
SD15-DPO(Wallace et al., 2024)	25	UNet	0.86B	0.28	5.37	0.218	31.25
SD15-LCM(Luo et al., 2023a)	1	UNet	0.86B	-1.58	5.04	0.194	27.20
SD15-LCM(Luo et al., 2023a)	4	UNet	0.86B	-0.23	5.40	0.214	30.11
SD15-TCD(Zheng et al., 2024)	1	UNet	0.86B	-1.49	5.10	0.196	28.30
SD15-TCD(Zheng et al., 2024)	4	UNet	0.86B	-0.04	5.28	0.212	30.43
PeRFlow(Yan et al., 2024)	4	UNet	0.86B	-0.20	5.51	0.211	29.54
SD15-Hyper(Ren et al., 2024)	1	UNet	0.86B	0.28	5.49	0.214	30.82
SD15-Hyper(Ren et al., 2024)	4	UNet	0.86B	0.42	5.41	0.217	31.03
SD15-InstaFlow(Liu et al., 2023)	1	UNet	0.86B	-0.16	5.03	0.207	30.68
SD15-SiDLSG(Zhou et al., 2024a)	1	UNet	0.86B	-0.18	5.16	0.210	30.04
SDXL-Base(Rombach et al., 2022)	25	UNet	2.6B	0.74	5.57	0.226	**31.83**
SDXL-DMD2-1024(Yin et al., 2024)	1	UNet	2.6B	0.82	5.45	0.224	31.78
SDXL-DMD2-1024(Yin et al., 2024)	4	UNet	2.6B	0.87	5.52	**0.231**	31.50
SDXL-DMD2-512(Yin et al., 2024)	1	UNet	2.6B	0.36	5.03	0.215	31.54
SDXL-DMD2-512(Yin et al., 2024)	4	UNet	2.6B	-0.18	5.17	0.206	29.28
SD15-DMD2-512(Yin et al., 2024)	1	UNet	2.6B	-0.12	5.24	0.211	30.00
SD21-Turbo(Sauer et al., 2023b)	1	UNet	0.86B	0.56	5.47	0.225	31.50
PixelArt-α-512(Chen et al., 2023)	25	DiT	0.6B	0.82	6.01	0.227	31.20
PixelArt-α-512(Chen et al., 2023)	15	DiT	0.6B	0.82	6.03	0.226	31.16
SD15-DI++ (α_r=0, α_c=1.5)	1	UNet	0.86B	0.29	5.26	0.216	30.64
SD15-DI++ (α_r=100, α_c=1.5)	1	UNet	0.86B	0.46	5.44	0.218	30.33
SD15-DI++ (α_r=1000, α_c=1.5)	1	UNet	0.86B	0.82	5.78	0.219	30.30
DiT-DI++ (α_r=0, α_c=4.5)	1	DiT	0.6B	0.74	5.91	0.225	31.04
DiT-DI++ (α_r=1, α_c=4.5)	1	DiT	0.6B	0.85	6.03	0.224	30.76
DiT-DI++ (α_r=10, α_c=4.5)	1	DiT	0.6B	**1.24**	**6.19**	0.225	30.80

Table 1: HPS v2 benchmark. We compare open-sourced models regardless of their base model and architecture.† indicates our implementation. Models with bold formats are our models. Numbers with bold formats are the highest scores.

Model	Animation	Concept-art	Painting	Photo	Average
GLIDE Nichol et al. (2021)	23.34	23.08	23.27	24.50	23.55
LAFITE Zhou et al. (2022)	24.63	24.38	24.43	25.81	24.81
VQ-Diffusion Gu et al. (2022)	24.97	24.70	25.01	25.71	25.10
FuseDream Liu et al. (2021)	25.26	25.15	25.13	25.57	25.28
Latent Diffusion Rombach et al. (2022)	25.73	25.15	25.25	26.97	25.78
CogView2 Ding et al. (2022)	26.50	26.59	26.33	26.44	26.47
DALL · E mini	26.10	25.56	25.56	26.12	25.83
Versatile Diffusion Xu et al. (2023b)	26.59	26.28	26.43	27.05	26.59
VQGAN + CLIP Esser et al. (2021)	26.44	26.53	26.47	26.12	26.39
DALL · E 2 Ramesh et al. (2022)	27.34	26.54	26.68	27.24	26.95
Stable Diffusion v1.4 Rombach et al. (2022)	27.26	26.61	26.66	27.27	26.95
Stable Diffusion v2.0 Rombach et al. (2022)	27.48	26.89	26.86	27.46	27.17
Epic Diffusion	27.57	26.96	27.03	27.49	27.26
DeepFloyd-XL	27.64	26.83	26.86	27.75	27.27
Openjourney	27.85	27.18	27.25	27.53	27.45
MajicMix Realistic	27.88	27.19	27.22	27.64	27.48
ChilloutMix	27.92	27.29	27.32	27.61	27.54
Deliberate	28.13	27.46	27.45	27.62	27.67
Realistic Vision	28.22	27.53	27.56	27.75	27.77
SDXL-base(Podell et al., 2023)	28.42	27.63	27.60	27.29	27.73
SDXL-Refiner(Podell et al., 2023)	28.45	27.66	27.67	27.46	27.80
Dreamlike Photoreal 2.0	28.24	27.60	27.59	27.99	27.86
SD15-15Step(Rombach et al., 2022)	26.76	26.37	26.41	27.12	26.66
SD15-25Step(Rombach et al., 2022)	27.04	26.57	26.61	27.30	26.88
SD15-DPO-15Step(Wallace et al., 2024)	27.11	26.75	26.70	27.30	26.97
SD15-DPO-25Step(Wallace et al., 2024)	27.54	26.97	26.99	27.49	27.25
SD15-LCM-1Step(Luo et al., 2023a)	23.35	23.41	23.53	23.81	23.52
SD15-LCM-4Step(Luo et al., 2023a)	26.42	25.79	25.95	26.91	26.27
SD15-TCD-1Step(Zheng et al., 2024)	23.37	23.16	23.26	23.88	23.42
SD15-TCD-4Step(Zheng et al., 2024)	26.67	26.25	26.26	27.19	26.59
SD15-Hyper-1Step(Ren et al., 2024)	27.76	27.36	27.41	27.63	27.54
SD15-Hyper-4Step(Ren et al., 2024)	28.04	27.39	27.42	27.89	27.69
SD15-Instaflow-1Step(Liu et al., 2023)	26.07	25.80	25.89	26.32	26.02
SD15-PeReflow-1Step(Yan et al., 2024)	25.70	25.45	25.57	25.96	25.67
SD15-BOOT-1Step(Gu et al., 2023)	25.29	24.40	24.61	25.16	24.86
SD21-SwiftBrush-1Step(Nguyen & Tran, 2023)	26.91	26.32	26.37	27.21	26.70
SD15-SiDLSG-1Step(Zhou et al., 2024a)	27.39	26.65	26.58	27.30	26.98
SD21-SiDLSG-1Step(Zhou et al., 2024a)	27.42	26.81	26.79	27.31	27.08
SD21-TURBO-1Step(Sauer et al., 2023b)	27.48	26.86	27.46	26.89	27.71
SDXL-DMD2-1Step-1024(Yin et al., 2024)	27.67	27.02	27.01	26.94	27.16
SDXL-DMD2-4Step-1024(Yin et al., 2024)	**28.97**	27.99	27.90	28.28	28.29
SDXL-DMD2-1Step-512(Yin et al., 2024)	27.70	27.07	27.02	26.94	27.18
SDXL-DMD2-4Step-512(Yin et al., 2024)	27.22	26.65	26.62	26.57	26.76
SD15-DMD2-1Step-512(Yin et al., 2024)	26.31	25.75	25.78	26.59	26.11
PixelArt-α-25Step-512(Chen et al., 2023)	28.77	27.92	27.96	28.37	28.25
PixelArt-α-15Step-512(Chen et al., 2023)	28.68	27.85	27.87	28.29	28.17
SD15-DI++-1Step($\alpha_r = 100, \alpha_c = 1.5$)	28.42	27.84	28.01	28.19	28.12
DiT-DI++-1Step($\alpha_r = 10, \alpha_c = 4.5$)	28.91	**28.25**	**28.28**	**28.50**	**28.48**


CFG + (large) Reward
CFG + Reward
CFG


Random Placed: one be PixelArt-alpha-14step, the other tow be DI++-1step with weak and strong reward
SDXL-TCD 4step
SDXL-Hyper 1step
SDXL-Turbo 1step
SDXL-Lightining 1step
LCM-PixelArt-alpha 2step

DIFF-INSTRUCT*: TOWARDS HUMAN-PREFERRED ONE-STEP TEXT-TO-IMAGE GENERATIVE MODELS

Weijian Luo*
Peking University
luoweijian@stu.pku.edu.cn

Colin Zhang
Xiaohongshu Inc
martin@xiaohongshu.com

Debing Zhang
Xiaohongshu Inc
dengyang@xiaohongshu.com

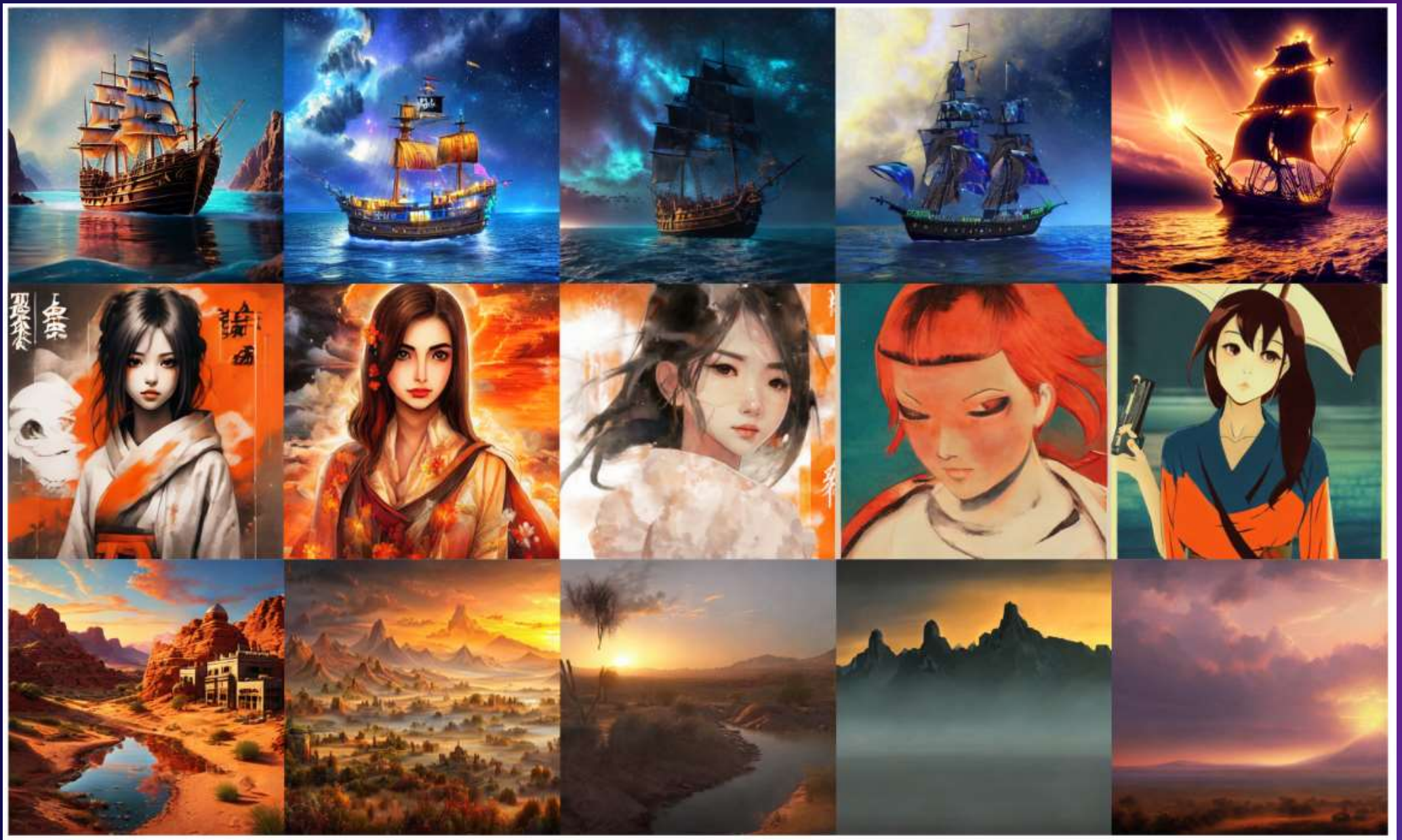
Zhengyang Geng
Carnegie Mellon University
zgeng2@cs.cmu.edu

RLHF (PPO) with KL? Weak.

RLHF with score-based divergence? Sick!

An DiT-based one-step T2I model with a leading HPSv2.0 score of **28.70**;

Figure 1: None cherry-picked generated images from one-step 0.6B DiT-DI* model with an record-breaking HPSv2.0 score of 28.70. After being trained with Diff-Instruct*, the images show better layouts, rich colors, vivid details, and aesthetic appearance, making them favored in terms of human preferences. Refer to the Appendix B.1 for the prompts used in comparison.


0.6B DiT-DI*
1-Step
0.86B SD1.5-DI*
1-Step
2.6B SDXL-DMD2
1-Step
0.86B SD1.5-SiDIsg
1-Step
0.86B SD1.5-DPO
25-Step

Problem formulation:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}_0 \sim p_\theta(\boldsymbol{x}_0|\boldsymbol{c})}\left[r(\boldsymbol{x}_0, \boldsymbol{c})\right], \quad \text{s.t. } \boldsymbol{D}(p_\theta(\cdot|\boldsymbol{c}), p_{ref}(\cdot|\boldsymbol{c})) \leq \delta \tag{3.2}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}_0 \sim p_\theta(\boldsymbol{x}_0|\boldsymbol{c})}\left[-\alpha r(\boldsymbol{x}_0, \boldsymbol{c})\right] + \boldsymbol{D}(p_\theta, p_{ref}) \tag{3.3}$$

If we take the Score-based divergence as the proximal regularization:

$$\mathbf{D}^{[0,T]}(p_\theta, p_{ref}) := \int_{t=0}^{T} w(t)\mathbb{E}_{\boldsymbol{x}_t \sim \pi_t}\left\{\mathbf{d}(\boldsymbol{s}_{p_{\theta,t}}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t))\right\}\mathrm{d}t, \tag{3.4}$$

We can practically minimize the learning objective with:

$$\begin{aligned}\mathcal{L}_{DI*}(\theta) =& \mathbb{E}_{\substack{\boldsymbol{z}\sim p_z,\\ \boldsymbol{x}_0 = g_\theta(\boldsymbol{z})}}\Bigg[-\alpha r(\boldsymbol{x}_0, \boldsymbol{c}) \\ &+ \int_{t=0}^{T} w(t)\mathbb{E}_{\substack{\boldsymbol{x}_t|\boldsymbol{x}_0\\ \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\left\{-\mathbf{d}'(\boldsymbol{y}_t)\right\}^T\left\{\boldsymbol{s}_{p_{\mathrm{sg}[\theta],t}}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right\}\mathrm{d}t\Bigg]\end{aligned} \tag{3.6}$$

with $\boldsymbol{y}_t := \boldsymbol{s}_{p_{\mathrm{sg}[\theta],t}}(\boldsymbol{x}_t) - \boldsymbol{s}_{q_t}(\boldsymbol{x}_t)$.

We can **decouple** the CFG-reward and explicit human reward, leading to **better performances and stable training**:

Theorem 3.2. Under mild conditions, if we set an implicit reward function as (3.9), the loss (3.8)

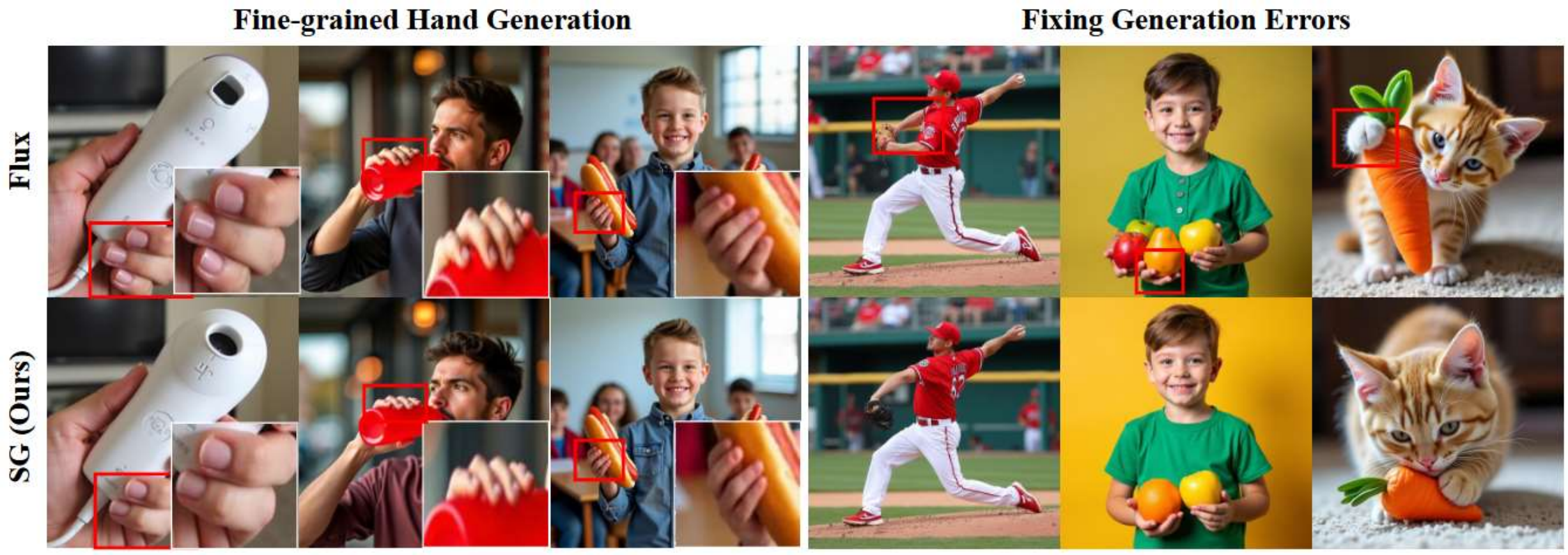
$$\mathcal{L}_{cfg}(\theta) = \int_{t=0}^{T} \mathbb{E}_{\substack{\boldsymbol{z}\sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}|\boldsymbol{c}) \\ \boldsymbol{x}_t|\boldsymbol{x}_0 \sim p(\boldsymbol{x}_t|\boldsymbol{x}_0)}} w(t) \left\{ \boldsymbol{s}_{ref}(\mathrm{sg}[\boldsymbol{x}_t]|t, \boldsymbol{c}) - \boldsymbol{s}_{ref}(\mathrm{sg}[\boldsymbol{x}_t]|t, \varnothing) \right\}^{T} \boldsymbol{x}_t \mathrm{d}t \tag{3.8}$$

has the same gradient as the negative implicit reward function (3.9)

$$-r(\boldsymbol{x}_0, \boldsymbol{c}) = -\int_{t=0}^{T} \mathbb{E}_{\boldsymbol{x}_t \sim p_{\theta,t}} w(t) \log \frac{p_{ref}(\boldsymbol{x}_t|t, \boldsymbol{c})}{p_{ref}(\boldsymbol{x}_t|t)} \mathrm{d}t. \tag{3.9}$$

Remark: the decouple of explicit and implicit rewards allows us to introduce flexible guidances into the Diff-Instruct++ and Diff-Instruct*.

For example, we can also introduce the self-guidance (SG, coming to public soon) to improve hands generations without expensive data curation and retraining.



Methods	Hand Generation		Face Generation		Text-Image Alignment	Generation Quality		
	FID-H ↓	Hand-Conf ↑	FID-F ↓	FaceScore ↑	CLIP Score ↑	FID ↓	Pick Score↑	ImageReward↑
Flux.1[23]	67.0163	0.9580	34.1976	4.5448	0.3080	29.0906	22.9802	1.0971
+Self Guidance	**66.4360**	**0.9616**	**34.0561**	**4.6634**	**0.3084**	**28.8046**	**22.9960**	**1.1046**

Table 2. **Quantitative results of SG Performances in Specific Generative Domains compared with Flux.1[23].** Bolded values highlight the best performance for each metric.

The algorithm

Algorithm 1: Diff-Instruct* for training human-preferred one-step text-to-image generators.

Input: prompt dataset $\mathcal{C}$, generator $g_\theta(\boldsymbol{x}_0|\boldsymbol{z},\boldsymbol{c})$, prior distribution p_z, reward model $r(\boldsymbol{x},\boldsymbol{c})$, reward model scale α_{rew}, CFG reward scale α_{cfg}, reference diffusion model $\boldsymbol{s}_{ref}(\boldsymbol{x}_t|c,\boldsymbol{c})$, assistant diffusion $\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c})$, forward diffusion $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ (2.1), assistant diffusion updates rounds K_{TA}, time distribution $\pi(t)$, diffusion model weighting $\lambda(t)$, generator loss time weighting $w(t)$.

while *not converge* **do**

freeze θ, update ψ for K_{TA} rounds using SGD by minimizing

$$\mathcal{L}(\psi) = \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,t\sim\pi(t)\\ \boldsymbol{x}_0=g_\theta(\boldsymbol{z}|\boldsymbol{c}),\boldsymbol{x}_t|\boldsymbol{x}_0\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \lambda(t)\|\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2 \mathrm{d}t.$$

freeze ψ, update θ using SGD by minimizing loss

$$\begin{aligned}\mathcal{L}_{DI*}(\theta) =& \mathbb{E}_{\substack{\boldsymbol{c}\sim\mathcal{C},\boldsymbol{z}\sim p_z,\\ \boldsymbol{x}_0=g_\theta(\boldsymbol{z},\boldsymbol{c})}}\Bigg\{ -\alpha_{rew}\cdot r(\boldsymbol{x}_0,\boldsymbol{c}) + \mathbb{E}_{\substack{t\sim\pi(t),\\ \boldsymbol{x}_t|\boldsymbol{x}_0\sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\Bigg[\\ & - w(t)\big\{\mathbf{d}'(\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c}) - \boldsymbol{s}_{ref}(\boldsymbol{x}_t|t,\boldsymbol{c}))\big\}^T\big\{\boldsymbol{s}_\psi(\boldsymbol{x}_t|t,\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\big\} \\ & + \alpha_{cfg}\cdot w(t)\big\{\boldsymbol{s}_{ref}(\mathrm{sg}[\boldsymbol{x}_t]|t,\boldsymbol{c}) - \boldsymbol{s}_{ref}(\mathrm{sg}[\boldsymbol{x}_t]|t,\varnothing)\big\}^T\boldsymbol{x}_t\Bigg]\Bigg\} \end{aligned} \tag{3.7}$$

end

return θ,ψ.

Score-based regularization is better than KL for alignment:

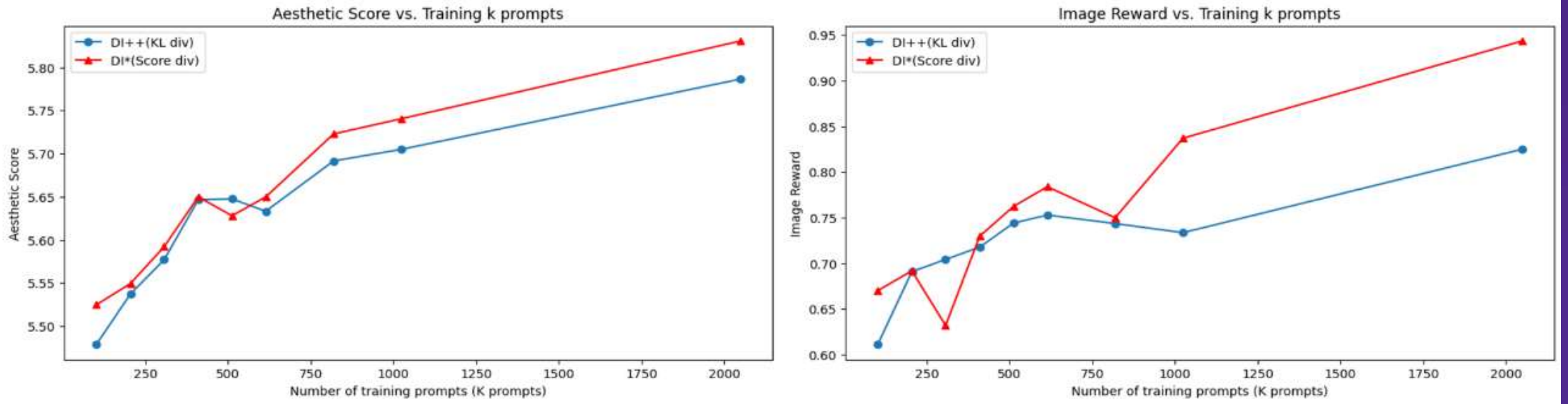


Figure 3: Comparison of Aesthetic Scores and Image Reward of Score-based divergence (DI*)

Model	Steps	Type	Params	Image Reward	Aes Score	Pick Score	CLIP Score	Additional Requirements
SD15-Base(Rombach et al., 2022)	15	UNet	0.86B	0.08	5.25	0.212	30.99	Image-Text
SD15-Base(Rombach et al., 2022)	25	UNet	0.86B	0.22	5.32	0.216	31.13	Image-Text
SD15-DPO(Wallace et al., 2024)	15	UNet	0.86B	0.20	5.29	0.214	31.07	Data, Sampling
SD15-DPO(Wallace et al., 2024)	25	UNet	0.86B	0.28	5.37	0.218	31.25	Data, Sampling
SD15-LCM(Luo et al., 2023a)	1	UNet	0.86B	-1.58	5.04	0.194	27.20	Data, Sampling
SD15-LCM(Luo et al., 2023a)	4	UNet	0.86B	-0.23	5.40	0.214	30.11	Data, Sampling
SD15-TCD(Zheng et al., 2024)	1	UNet	0.86B	-1.49	5.10	0.196	28.30	Data, Sampling
SD15-TCD(Zheng et al., 2024)	4	UNet	0.86B	-0.04	5.28	0.212	30.43	Data, Sampling
PeRFlow(Yan et al., 2024)	4	UNet	0.86B	-0.20	5.51	0.211	29.54	Data, Sampling
SD15-Hyper(Ren et al., 2024)	1	UNet	0.86B	0.28	5.49	0.214	30.82	Data, Sampling
SD15-Hyper(Ren et al., 2024)	4	UNet	0.86B	0.42	5.41	0.217	31.03	Reward, Seg-Model
SD15-InstaFlow(Liu et al., 2023)	1	UNet	0.86B	-0.16	5.03	0.207	30.68	Data, Sampling
SD15-SiDLSG(Zhou et al., 2024a)	1	UNet	0.86B	-0.18	5.16	0.210	30.04	-
SDXL-Base(Rombach et al., 2022)	25	UNet	2.6B	0.74	5.57	0.226	**31.83**	Image-Text
SDXL-DMD2-1024(Yin et al., 2024)	1	UNet	2.6B	0.82	5.45	0.224	31.78	Image-Text
SDXL-DMD2-1024(Yin et al., 2024)	4	UNet	2.6B	0.87	5.52	**0.231**	31.50	Image-Text
SDXL-DMD2-512(Yin et al., 2024)	1	UNet	2.6B	0.36	5.03	0.215	31.54	Image-Text
SDXL-DMD2-512(Yin et al., 2024)	4	UNet	2.6B	-0.18	5.17	0.206	29.28	Image-Text
SD15-DMD2-512(Yin et al., 2024)	1	UNet	2.6B	-0.12	5.24	0.211	30.00	Image-Text
SD21-Turbo(Sauer et al., 2023b)	1	UNet	0.86B	0.56	5.47	0.225	31.50	Image-Text
PixelArt-α-512(Chen et al., 2023)	25	DiT	0.6B	0.82	6.01	0.227	31.20	Image-Text
PixelArt-α-512(Chen et al., 2023)	15	DiT	0.6B	0.82	6.03	0.226	31.16	Image-Text
SD15-DI++(Anonymous, 2024)†	1	UNet	0.86B	0.82	5.78	0.219	30.30	Reward
DiT-DI++(Anonymous, 2024)†	1	DiT	0.6B	1.24	6.19	0.225	30.80	Reward
SD15-DI*($\alpha_r = 0, \alpha_c = 1.5$)	1	UNet	0.86B	0.34	5.27	0.217	30.83	Reward
SD15-DI*($\alpha_r = 100, \alpha_c = 1.5$)	1	UNet	0.86B	0.62	5.44	0.218	30.76	Reward
SD15-DI*($\alpha_r = 1000, \alpha_c = 4.5$)	1	UNet	0.86B	0.73	5.56	0.219	30.71	Reward
SD15-DI*($\alpha_r = 1000, \alpha_c = 1.5$)	1	UNet	0.86B	0.94	5.83	0.220	30.49	Reward
DiT-DI*($\alpha_r = 1, \alpha_c = 4.5$)	1	DiT	0.6B	0.98	6.02	0.225	31.00	Reward
DiT-DI*($\alpha_r = 10, \alpha_c = 4.5$)	1	DiT	0.6B	**1.31**	**6.30**	0.225	30.84	Reward

SD15-15Step(Rombach et al., 2022)	26.76	26.37	26.41	27.12	26.66
SD15-25Step(Rombach et al., 2022)	27.04	26.57	26.61	27.30	26.88
SD15-DPO-15Step(Wallace et al., 2024)	27.11	26.75	26.70	27.30	26.97
SD15-DPO-25Step(Wallace et al., 2024)	27.54	26.97	26.99	27.49	27.25
SD15-LCM-1Step(Luo et al., 2023a)	23.35	23.41	23.53	23.81	23.52
SD15-LCM-4Step(Luo et al., 2023a)	26.42	25.79	25.95	26.91	26.27
SD15-TCD-1Step(Zheng et al., 2024)	23.37	23.16	23.26	23.88	23.42
SD15-TCD-4Step(Zheng et al., 2024)	26.67	26.25	26.26	27.19	26.59
SD15-Hyper-1Step(Ren et al., 2024)	27.76	27.36	27.41	27.63	27.54
SD15-Hyper-4Step(Ren et al., 2024)	28.04	27.39	27.42	27.89	27.69
SD15-InstaFlow-1Step(Liu et al., 2023)	26.07	25.80	25.89	26.32	26.02
SD15-PeReFlow-1Step(Yan et al., 2024)	25.70	25.45	25.57	25.96	25.67
SD15-BOOT-1Step(Gu et al., 2023)	25.29	24.40	24.61	25.16	24.86
SD21-SwiftBrush-1Step(Nguyen & Tran, 2023)	26.91	26.32	26.37	27.21	26.70
SD15-SiDLSG-1Step(Zhou et al., 2024a)	27.39	26.65	26.58	27.30	26.98
SD21-SiDLSG-1Step(Zhou et al., 2024a)	27.42	26.81	26.79	27.31	27.08
SD21-TURBO-1Step(Sauer et al., 2023b)	27.48	26.86	27.46	26.89	27.71
SDXL-DMD2-1Step-1024(Yin et al., 2024)	27.67	27.02	27.01	26.94	27.16
SDXL-DMD2-4Step-1024(Yin et al., 2024)	28.97	27.99	27.90	28.28	28.29
SDXL-DMD2-1Step-512(Yin et al., 2024)	27.70	27.07	27.02	26.94	27.18
SDXL-DMD2-4Step-512(Yin et al., 2024)	27.22	26.65	26.62	26.57	26.76
SD15-DMD2-1Step-512(Yin et al., 2024)	26.31	25.75	25.78	26.59	26.11
PixelArt-α-25Step-512(Chen et al., 2023)	28.77	27.92	27.96	28.37	28.25
PixelArt-α-15Step-512(Chen et al., 2023)	28.68	27.85	27.87	28.29	28.17
SD15-DI++-1Step(Anonymous, 2024)†	28.42	27.84	28.01	28.19	28.12
SD15-DI*-1Step($\alpha_r = 1000, \alpha_c = 1.5$)	28.56	28.05	28.17	28.31	28.27
DiT-DI*-1Step($\alpha_r = 10, \alpha_c = 4.5$)(ours)	28.78	28.31	28.48	28.37	28.48
DiT-DI*-1Step($\alpha_r = 1, \alpha_c = 4.5$)(ours)	**29.13**	**28.51**	**28.51**	**28.63**	**28.70**

Broader Few-step Models

FLOW GENERATOR MATCHING

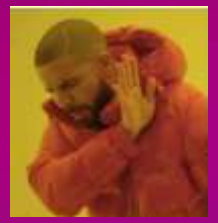
Zemin Huang
Zhejiang University, Westlake University
huangzem@zju.edu.cn

Zhengyang Geng
Carnegie Mellon University
zgeng2@cs.cmu.edu

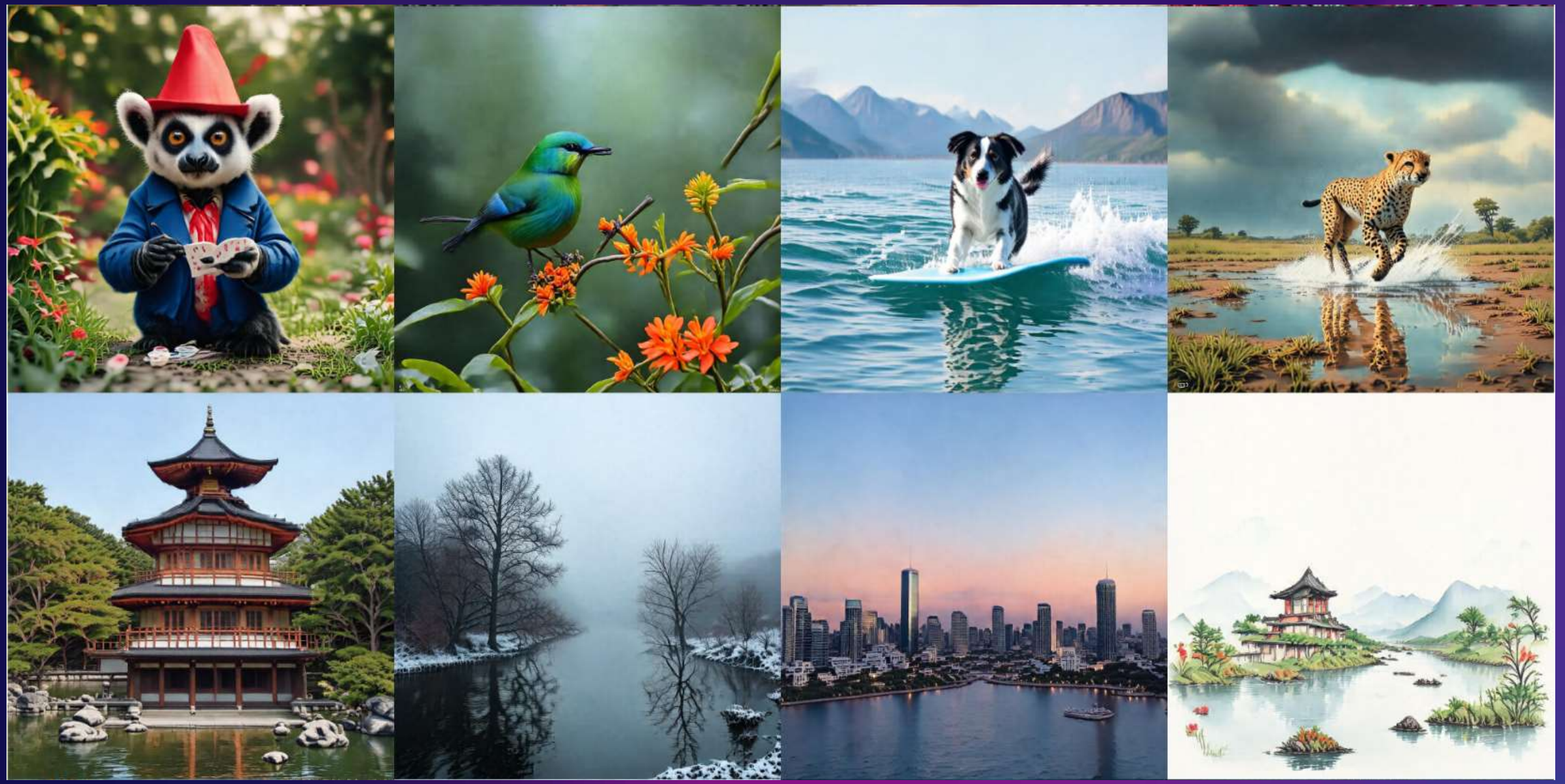
Weijian Luo*
Peking University
luoweijian@stu.pku.edu.cn

Guo-jun Qi
Westlake University
guojunq@gmail.com

Diffusion distillation? No.



Flow distillation?Yes!



Practical Instance of Flow Matching Models. In this paper, we especially consider a widely used flow matching model, the rectified flow (ReFlow) (Liu et al., 2022; Albergo & Vanden-Eijnden, 2022) as a specific instance. Our theory and algorithms for the general flow-matching model share the same concepts as the ones based on ReFlow. The ReFlow defines the conditional vector field as

$$\boldsymbol{u}_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{1-t}. \tag{3.5}$$

This results in a simple training objective as

$$\mathcal{L}_{ReFlow}(\theta) = \mathbb{E}_{\substack{t,\boldsymbol{x}_0\sim q_0(\boldsymbol{x}_0),\boldsymbol{x}_1\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\\ \boldsymbol{x}_t=(1-t)\boldsymbol{x}_0+t\boldsymbol{x}_1}} \|\boldsymbol{v}_\theta(\boldsymbol{x}_t,t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|_2^2 \tag{3.6}$$

Directly minimizing flow-matching objective is intractable

$$\mathcal{L}_{FM}(\theta) := \mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}} \|\boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t) - \boldsymbol{u}_t(\boldsymbol{x}_t)\|^2 \tag{4.1}$$

$$= \mathbb{E}_{\substack{t,\boldsymbol{z}\sim p_z(\boldsymbol{z}),\boldsymbol{x}_0=g_\theta(\boldsymbol{z}),\\ \boldsymbol{x}_t\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \|\boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t) - \boldsymbol{u}_t(\boldsymbol{x}_t)\|^2 \tag{4.2}$$

We can get tractable alternative loss:

$$\frac{\partial}{\partial\theta}\mathcal{L}_{FM}(\theta) = \frac{\partial}{\partial\theta}\mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}}\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\|_2^2$$
$$= \mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}}\left\{\frac{\partial}{\partial\boldsymbol{x}_t}\{\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\|_2^2\}\frac{\partial\boldsymbol{x}_t(\theta)}{\partial\theta} - 2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\}^T\frac{\partial}{\partial\theta}\boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\right\}$$
$$= \mathrm{Grad}_1(\theta) + \mathrm{Grad}_1(\theta). \tag{4.3}$$

Where $\mathrm{Grad}_1(\theta)$ and $\mathrm{Grad}_2(\theta)$ are defined with

$$\mathrm{Grad}_1(\theta) = \mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}}\left\{\frac{\partial}{\partial\boldsymbol{x}_t}\{\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\|_2^2\}\frac{\partial\boldsymbol{x}_t(\theta)}{\partial\theta}\right\}, \tag{4.4}$$

$$\mathrm{Grad}_2(\theta) = \mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}}\left\{-2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\}^T\frac{\partial}{\partial\theta}\boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\right\}. \tag{4.5}$$

$$\mathcal{L}_1(\theta) = \mathbb{E}_{t,\boldsymbol{x}_t\sim p_{\theta,t}}\left\{\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\|_2^2\right\}$$
$$= \mathbb{E}_{\substack{t,\boldsymbol{z}\sim p_z,\boldsymbol{x}_0=g_\theta(\boldsymbol{z}),\\ \boldsymbol{x}_t\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\left\{\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\|_2^2\right\} \tag{4.6}$$

We can get tractable alternative loss:

Theorem 4.1 (Flow Product Identity). Let $\mathbf{f}(\cdot, \theta)$ be a vector-valued function, using the notations in Section 4.1, under mild conditions, the identity holds:

$$\mathbb{E}_{\boldsymbol{x}_t \sim p_{\theta,t}} \mathbf{f}(\boldsymbol{x}_t, \theta)^T \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t) = \mathbb{E}_{\substack{\boldsymbol{x}_0 \sim p_{\theta,0}, \\ \boldsymbol{x}_t | \boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t | \boldsymbol{x}_0)}} \mathbf{f}(\boldsymbol{x}_t, \theta)^T \boldsymbol{u}_t(\boldsymbol{x}_t | \boldsymbol{x}_0) \tag{4.7}$$

Theorem 4.2. If distribution $p_{\theta,t}$ satisfies some wild regularity conditions, then we have for all θ-parameter free vector-valued function $\boldsymbol{u}_t(\cdot)$, the equation holds for all parameter θ:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}_t \sim p_{\theta,t}} \left\{ -2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t)\}^T \frac{\partial}{\partial \theta} \boldsymbol{v}_{\theta,t}(\boldsymbol{x}_t) \right\} \\ &= \frac{\partial}{\partial \theta} \mathbb{E}_{\substack{\boldsymbol{x}_0 \sim p_{\theta,0}, \\ \boldsymbol{x}_t | \boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t | \boldsymbol{x}_0)}} \left\{ 2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\}^T \{\boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t) - \boldsymbol{u}_t(\boldsymbol{x}_t | \boldsymbol{x}_0)\} \right\} \end{aligned} \tag{4.8}$$

$$\mathcal{L}_2(\theta) = \mathbb{E}_{\substack{t, \boldsymbol{z} \sim p_z, \boldsymbol{x}_0 = g_\theta(\boldsymbol{z}), \\ \boldsymbol{x}_t | \boldsymbol{x}_0 \sim q_t(\boldsymbol{x}_t | \boldsymbol{x}_0)}} \left\{ 2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\}^T \{\boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t) - \boldsymbol{u}_t(\boldsymbol{x}_t | \boldsymbol{x}_0)\} \right\}. \tag{4.9}$$

Algorithm 1: Flow Generator Matching Algorithm for training one-step Generators.

Input: pre-trained flow matching model $\boldsymbol{u}_t(\cdot)$, one-step generator g_θ, prior distribution p_z, online flow model $\boldsymbol{v}_\psi(\cdot)$, time $t \in \mathcal{U}[0,1]$, and conditional transition $q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$.

while *not converge* **do**

freeze θ, update ψ using SGD by minimizing the flow matching loss

$$\mathcal{L}_{FM}(\psi) = \mathbb{E}_{\substack{t,\boldsymbol{z}\sim p_z,\boldsymbol{x}_0=g_\theta(\boldsymbol{z}),\\ \boldsymbol{x}_t|\boldsymbol{x}_0\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \|\boldsymbol{v}_\psi(\boldsymbol{x}_t,t) - \boldsymbol{u}_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2.$$

freeze ψ, update θ using SGD with by minimizing the FGM loss (4.10):

$$\mathcal{L}_{FGM}(\theta) = \mathcal{L}_1(\theta) + \mathcal{L}_2(\theta)$$

$$\mathcal{L}_1(\theta) = \mathbb{E}_{\substack{t,\boldsymbol{z}\sim p_z,\boldsymbol{x}_0=g_\theta(\boldsymbol{z}),\\ \boldsymbol{x}_t\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left\{ \|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\|_2^2 \right\} \tag{4.11}$$

$$\mathcal{L}_2(\theta) = \mathbb{E}_{\substack{t,\boldsymbol{z}\sim p_z,\boldsymbol{x}_0=g_\theta(\boldsymbol{z}),\\ \boldsymbol{x}_t|\boldsymbol{x}_0\sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \left\{ 2\{\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t)\}^T \{\boldsymbol{v}_{\mathrm{sg}[\theta],t}(\boldsymbol{x}_t) - \boldsymbol{u}_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\} \right\} \tag{4.12}$$

end

return θ, ψ.

Table 1: Unconditional sample quality on CIFAR-10. † means method we reproduced.

FAMILY	METHOD	NFE (↓)	FID (↓)
DIFFUSION & GAN	DDPM (HO ET AL., 2020)	1000	3.17
	DD-GAN(T=2) (XIAO ET AL., 2021)	2	4.08
	KD LUHMAN & LUHMAN (2021)	1	9.36
	TDPM (ZHENG ET AL., 2023)	1	8.91
	DFNO (ZHENG ET AL., 2022)	1	4.12
	STYLEGAN2-ADA (KARRAS ET AL., 2020A)	1	2.92
	STYLEGAN2-ADA+DI (LUO ET AL., 2023A)	1	2.71
	EDM (KARRAS ET AL., 2022)	35	1.97
	EDM (KARRAS ET AL., 2022)	15	5.62
	PD (SALIMANS & HO, 2022)	2	5.13
	CD (SONG ET AL., 2023)	2	2.93
	GET (GENG ET AL., 2024A)	1	6.91
	CT (SONG ET AL., 2023)	1	8.70
	ICT-DEEP (SONG & DHARIWAL, 2023)	2	2.24
	DIFF-INSTRUCT (LUO ET AL., 2023A)	1	4.53
	DMD (YIN ET AL., 2024B)	1	3.77
	CTM (KIM ET AL., 2023)	1	1.98
	CTM(KIM ET AL., 2023)	2	**1.87**
	SID ($\alpha = 1.0$) (ZHOU ET AL., 2024)	1	1.92
	SID ($\alpha = 1.2$)(ZHOU ET AL., 2024)	1	2.02
	DI†	1	3.70
FLOW-BASED	1-REFLOW (+DISTILL) (LIU ET AL., 2022)	1	6.18
	2-REFLOW (+DISTILL) (LIU ET AL., 2022)	1	4.85
	3-REFLOW (+DISTILL) (LIU ET AL., 2022)	1	5.21
	CFM(YANG ET AL., 2024)	2	5.34
	FLOW	100	**2.93**
	FLOW	50	3.67
	FGM (OURS)	1	3.08

Table 2: Class-conditional sample quality on CIFAR10 dataset. † means method we reproduced.

FAMILY	METHOD	NFE (↓)	FID (↓)
DIFFUSION & GAN	BIGGAN (BROCK ET AL., 2019)	1	14.73
	BIGGAN+TUNE(BROCK ET AL., 2019)	1	8.47
	STYLEGAN2 (KARRAS ET AL., 2020B)	1	6.96
	MULTIHINGE (KAVALEROV ET AL., 2021)	1	6.40
	FQ-GAN (ZHAO ET AL., 2020)	1	5.59
	STYLEGAN2-ADA (KARRAS ET AL., 2020A)	1	2.42
	STYLEGAN2-ADA+DI (LUO ET AL., 2023A)	1	2.27
	STYLEGAN2 + SMART (XIA ET AL., 2023)	1	2.06
	STYLEGAN-XL (SAUER ET AL., 2022)	1	1.85
	STYLESAN-XL (TAKIDA ET AL., 2023)	1	**1.36**
	EDM (KARRAS ET AL., 2022)	35	1.82
	EDM (KARRAS ET AL., 2022)	20	2.54
	EDM (KARRAS ET AL., 2022)	10	15.56
	EDM (KARRAS ET AL., 2022)	1	314.81
	GET (GENG ET AL., 2024A)	1	6.25
	DIFF-INSTRUCT (LUO ET AL., 2023A)	1	4.19
	DMD (W.O. REG) (YIN ET AL., 2024B)	1	5.58
	DMD (W.O. KL) (YIN ET AL., 2024B)	1	3.82
	DMD (YIN ET AL., 2024B)	1	2.66
	CTM (KIM ET AL., 2023)	1	1.73
	CTM(KIM ET AL., 2023)	2	1.63
	GDD (ZHENG & YANG, 2024)	1	1.58
	GDD-I (ZHENG & YANG, 2024)	1	1.44
	SID ($\alpha = 1.0$) (ZHOU ET AL., 2024)	1	1.93
	SID ($\alpha = 1.2$)(ZHOU ET AL., 2024)	1	1.71
FLOW-BASED	**FLOW**	100	2.87
	FLOW	50	3.66
	FGM (OURS)	1	**2.58**


SD3 (28 steps)
Hyper-SD3 (4 steps)
Flash-SD3 (4 steps)
Ours (1 step)

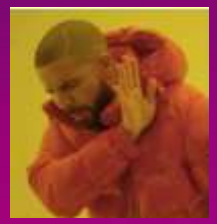
Model	Overall	Objects		Counting	Colors	Position	Color Attribution	NFEs
		Single	Two					
minDALL-E(Zeqiang et al., 2023)	0.23	0.73	0.11	0.12	0.37	0.02	0.01	-
SD v1.5(Rombach et al., 2022)	0.43	0.97	0.38	0.35	0.76	0.04	0.06	50
PixArt-alpha(Chen et al., 2023)	0.48	*0.98*	0.50	0.44	0.80	0.08	0.07	40
SD v2.1(Rombach et al., 2022)	0.50	*0.98*	0.51	0.44	0.85	0.07	0.17	50
DALL-E 2	0.52	0.94	0.66	0.49	0.77	0.10	0.19	-
SDXL(Podell et al., 2023)	0.55	*0.98*	0.74	0.39	0.85	0.15	0.23	50
SDXL Turbo (Sauer et al., 2023)	0.55	**1.00**	0.72	0.49	0.80	0.10	0.18	1
IF-XL	0.61	0.97	0.74	**0.66**	0.81	0.13	0.35	100
DALL-E 3(James Betker et al., 2023)	0.67	0.96	0.87	0.47	0.83	**0.43**	*0.45*	-
SD3†(Esser et al., 2024),	**0.70**	0.99	**0.88**	0.60	0.85	0.30	**0.59**	28
Hyper-SD3†(Ren et al., 2024)	0.63	**1.00**	0.74	0.56	*0.84*	0.22	0.42	4
Flash-SD3†(Chadebec et al., 2024)	0.67	0.99	0.77	*0.59*	**0.86**	*0.28*	0.54	4
Ours	*0.65*	**1.00**	*0.82*	0.58	0.83	0.20	0.46	1

Table 3: **GenEval metrics**. Our distilled model closely matches the performance of the teacher model SD3 (depth=24) on GenEval (Ghosh et al., 2024). Same as Esser et al. (2024) we highlight the **best**, second best, and *third best* entries. (†indicates that the metrics were evaluated by us.)

CONSISTENCY MODELS MADE EASY

Zhengyang Geng[1] **Ashwini Pokle**[1] **William Luo**[2] **Justin Lin**[1] **J. Zico Kolter**[1]

Consistency training? Expensive!



Easy consistency tuning (ECT)? Cheeeeaaap!

Consistency model

Consistency Models. CMs are built upon the PF-ODE in Eq. (3), which establishes a bijective mapping between data distribution and noise distribution. CMs learn a *consistency function* $f(\mathbf{x}_t, t)$ that maps the noisy image $\mathbf{x}_t$ back to the clean image $\mathbf{x}_0$

$$f(\mathbf{x}_t, t) = \mathbf{x}_0. \tag{4}$$

During training, CMs first discretize the PF-ODE into $N-1$ subintervals with boundaries given by $t_{\min} = t_1 < t_2 < \ldots < t_N = T$. The model is trained on the following CM loss, which minimizes a metric between adjacent points on the sampling trajectory

$$\arg\min_{\theta} \mathbb{E}\left[w(t_i) d(f_\theta(\mathbf{x}_{t_{i+1}}, t_{i+1}), f_{\theta^-}(\tilde{\mathbf{x}}_{t_i}, t_i))\right]. \tag{6}$$

$\rho = 0.7$. Further, the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ can either be estimated from a pretrained diffusion model, which results in Consistency Distillation (CD), or can be estimated with an unbiased score estimator

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = \mathbb{E}\left[\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) \middle| \mathbf{x}_t\right] = \mathbb{E}\left[-\frac{\mathbf{x}_t - \mathbf{x}_0}{t^2} \middle| \mathbf{x}_t\right], \tag{7}$$

which results in consistency training (CT).

Consistency model made easy

As stated in Sec. 2, CMs learn a *consistency function* $f(\mathbf{x}_t, t)$ that maps the noisy image $\mathbf{x}_t$ back to the clean image $\mathbf{x}_0$: $f(\mathbf{x}_t, t) = \mathbf{x}_0$. Instead, by taking the time derivative of both sides, given by the differential form

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_0 = 0. \tag{9}$$

$$\boxed{f(\mathbf{x}_t, t) = \mathbf{x}_0 \Leftrightarrow \frac{\mathrm{d}f}{\mathrm{d}t} = 0, f(\mathbf{x}_0, 0) = \mathbf{x}_0.} \tag{10}$$

Finite Difference Approximation. To learn the consistency condition, we discretize the differential form $\frac{\mathrm{d}f}{\mathrm{d}t} = 0$ using a finite-difference approximation:

$$0 = \frac{\mathrm{d}f}{\mathrm{d}t} \approx \frac{f_\theta(\mathbf{x}_t) - f_\theta(\mathbf{x}_r)}{t - r} \tag{11}$$

Consistency model made easy

Algorithm 1 Easy Consistency Tuning (ECT)

Input: Dataset $\mathcal{D}$, a pretrained diffusion model ϕ, mapping function $p(r \mid t, \text{Iters})$, weighting function $w(t)$.
Init: $\theta \leftarrow \theta_\phi$, $\text{Iters} = 0$.
repeat
  Sample $\mathbf{x}_0 \sim \mathcal{D}$, $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$, $t \sim p(t)$, $r \sim p(r \mid t, \text{Iters})$
  Compute $\mathbf{x}_t = \mathbf{x}_0 + t \cdot \boldsymbol{\epsilon}$, $\mathbf{x}_r = \mathbf{x}_0 + r \cdot \boldsymbol{\epsilon}$, $\Delta t = t - r$
  $L(\theta) = w(t) \cdot \mathbf{d}(f_\theta(\mathbf{x}_t), f_{\text{sg}(\theta)}(\mathbf{x}_r))$ ▷ sg is stop-gradient operator
  $\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$
  $\text{Iters} = \text{Iters} + 1$
until $\Delta t \to \mathrm{d}t$ **return** θ ▷ ECM

We refer to $p(r|t)$ as the mapping function. Since we need to shrink $\Delta t \to \mathrm{d}t$ as the training progresses, we augment the mapping function to depend on training iterations, $p(r|t, \text{iters})$, to control $\Delta t = (t - r) \to \mathrm{d}t$. We parametrize the mapping function $p(r|t, \text{iters})$ as

$$\frac{r}{t} = 1 - \frac{1}{q^a} n(t) = 1 - \frac{1}{q^{\lfloor \text{iters}/d \rfloor}} n(t), \tag{15}$$

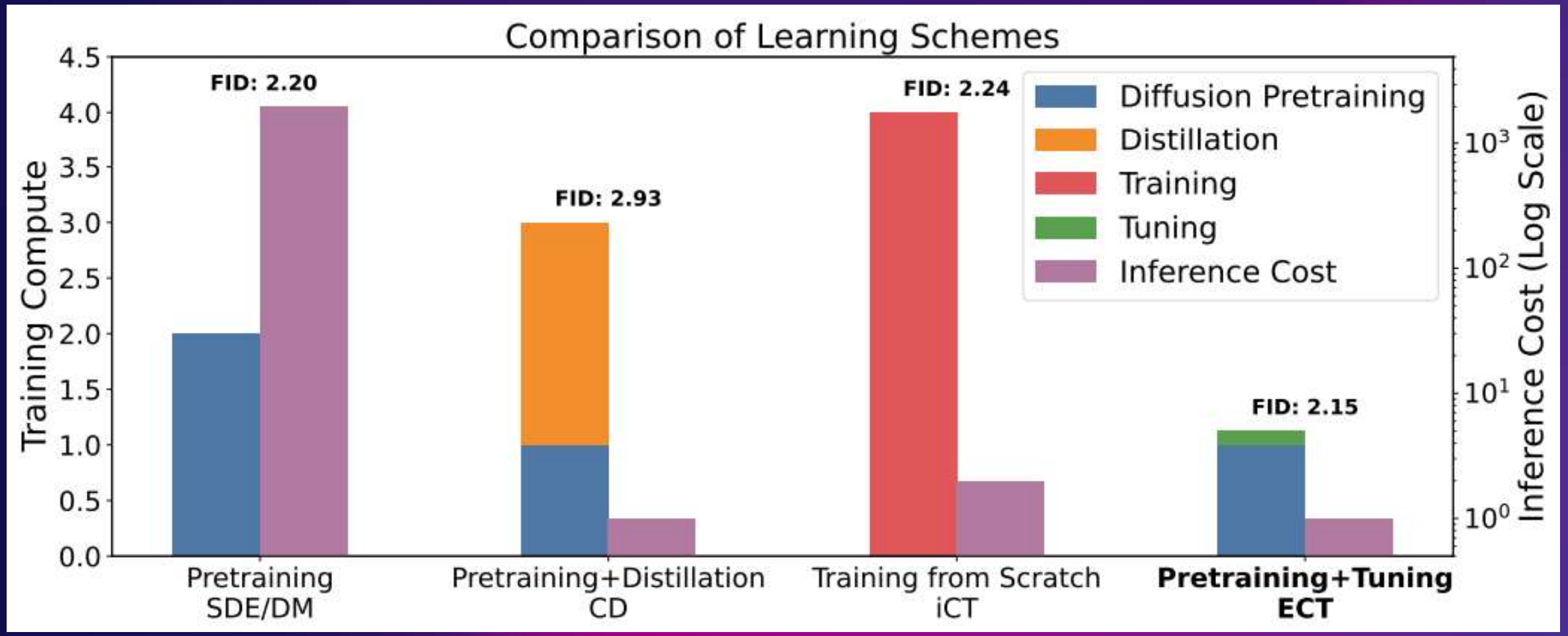
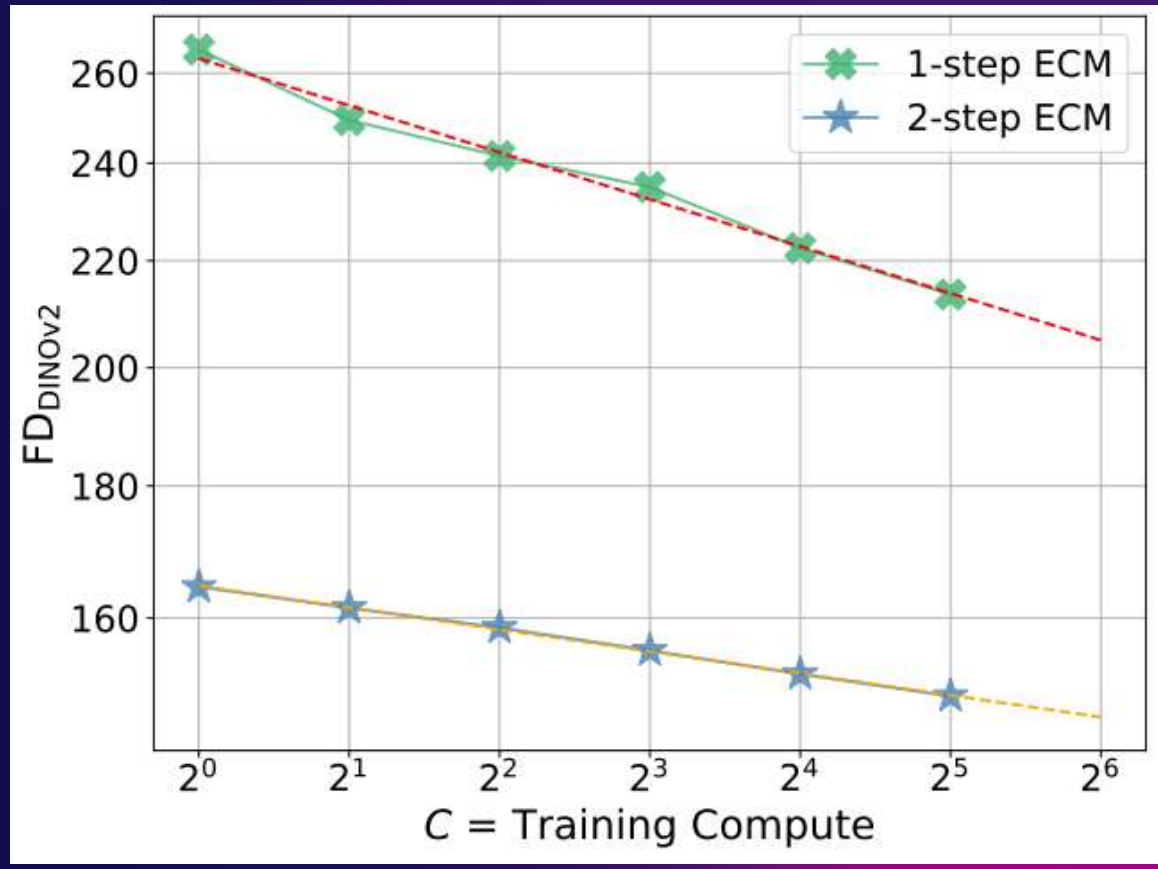

Comparison of Learning Schemes
Training Compute
Inference Cost (Log Scale)
FID: 2.20
FID: 2.93
FID: 2.24
FID: 2.15
Diffusion Pretraining
Distillation
Training
Tuning
Inference Cost
Pretraining
SDE/DM
Pretraining+Distillation
CD
Training from Scratch
iCT
Pretraining+Tuning
ECT

Table 1: Generative performance on unconditional CIFAR-10 and class-conditional ImageNet 64×64. We use a budget of 12.8M training images (batch size 128 and 100k iterations) for ECMs. ⋆ stands for a budget of 102.4M training images (batch size 1024 and 100k iterations) on ImageNet 64×64.

CIFAR-10		
Method	FID↓	NFE↓
Diffusion Models		
Score SDE (Song et al., 2020)	2.38	2000
Score SDE-deep (Song et al., 2020)	2.20	2000
EDM (Karras et al., 2022)	2.01	35
EDM (DPM-Solver-v3) (Zheng et al., 2024)	2.51	10
Diffusion Distillation		
PD (Salimans and Ho, 2022)	8.34	1
GET (Geng et al., 2024)	5.49	1
Diff-Instruct (Luo et al., 2024)	4.53	1
TRACT (Berthelot et al., 2023)	3.32	2
CD (LPIPS) (Song et al., 2023)	3.55	1
CD (LPIPS) (Song et al., 2023)	2.93	2
Consistency Models		
iCT (Song and Dhariwal, 2023)	2.83	1
	2.46	2
iCT-deep (Song and Dhariwal, 2023)	2.51	1
	2.24	2
ECT		
ECM (100k iters)	4.54	1
ECM (200k iters)	3.86	1
ECM (400k iters)	3.60	1
ECM (100k iters)	2.20	2
ECM (200k iters)	2.15	2
ECM (400k iters)	2.11	2

ImageNet 64×64		
Method	FID↓	NFE↓
Diffusion Models		
ADM (Karras et al., 2022)	2.07	250
EDM (Karras et al., 2022)	2.22	79
EDM2-XL (Karras et al., 2023)	1.33	63
Diffusion Distillation		
BOOT (Gu et al., 2023)	16.3	1
DFNO (LPIPS) (Zheng et al., 2023)	7.83	1
Diff-Instruct (Luo et al., 2024)	5.57	1
TRACT (Berthelot et al., 2023)	4.97	2
PD (LPIPS) (Salimans and Ho, 2022)	5.74	2
CD (LPIPS) (Song et al., 2023)	4.70	2
Consistency Models		
iCT (Song and Dhariwal, 2023)	3.20	2
iCT-deep (Song and Dhariwal, 2023)	2.77	2
ECT		
ECM-S (100k iters)	3.18	2
ECM-M (100k iters)	2.35	2
ECM-L (100k iters)	2.14	2
ECM-XL (100k iters)	1.96	2
ECM-S⋆	4.05	1
ECM-S⋆	2.79	2
ECM-XL⋆	2.49	1
ECM-XL⋆	1.67	2

The Scaling Law of Consistency Models



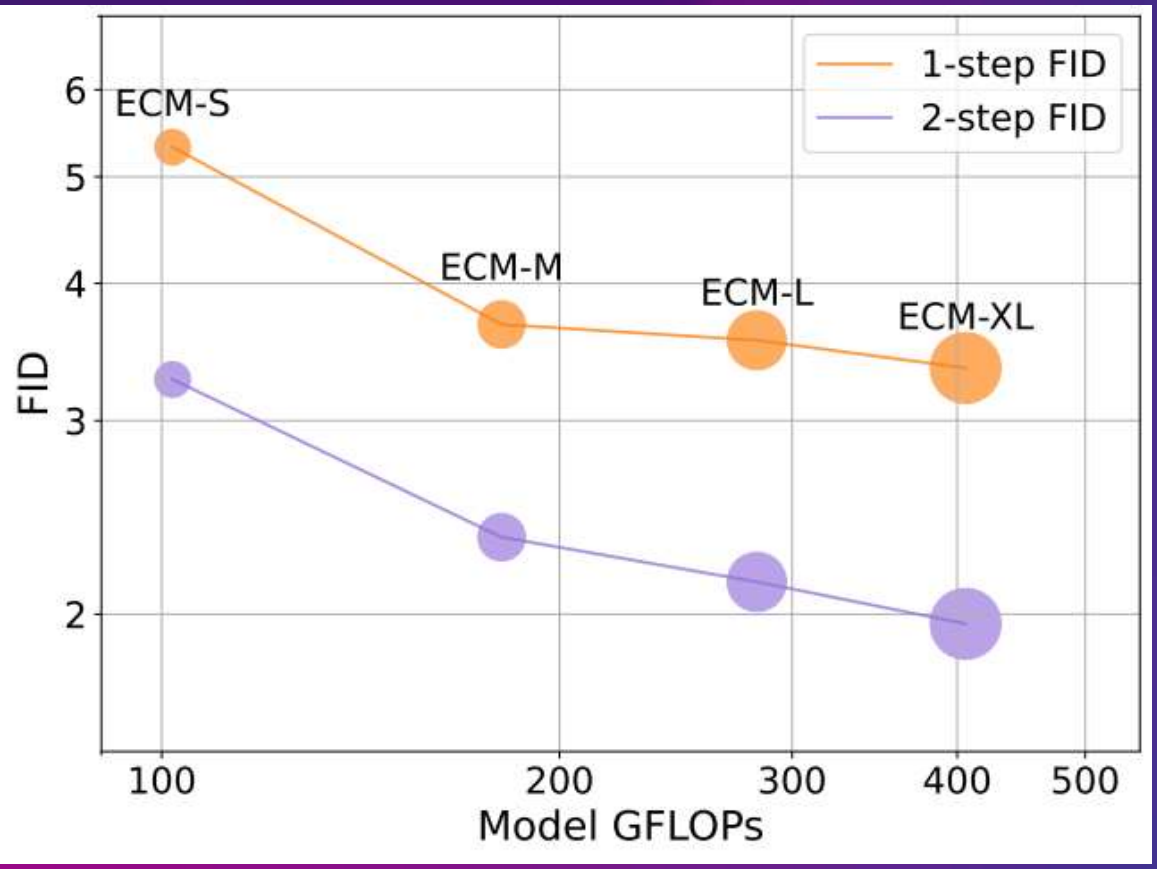


Figure 3: Scaling up training compute and model sizes results in improved sample quality on ImageNet 64×64. Each triplet (left-to-right) has 2-step samples from ECM-S trained with 12.8M images, ECM-S trained with 102.4M images, and ECM-XL trained with 102.4M images.

Thank You for Listening!
Q&A Session