Simple and Fast Distillation of Diffusion Models

Zhenyu Zhou[1,2], Defang Chen[3], Can Wang[1,2], Chun Chen[1,2], Siwei Lyu[3]

[1] Zhejiang University, State Key Laboratory of blockchain and Data Security

[2] Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security

[3] University at Buffalo, State University of New York

Reporter : Zhenyu Zhou

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

Forward [1]:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \tag{1}$$

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

Forward [1]:
$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \tag{1}$$

Backward (PFODE) [1]:
$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t \tag{2}$$

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

Forward [1]: $$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \tag{1}$$

Backward (PFODE) [1]: $$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t \tag{2}$$

EDM formulation [2]: $$\mathrm{d}\mathbf{x} = -t\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\mathrm{d}t \tag{3}$$

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

Forward [1]: $$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \tag{1}$$

Backward (PFODE) [1]: $$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t \tag{2}$$

EDM formulation [2]: $$\mathrm{d}\mathbf{x} = -t\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\mathrm{d}t \tag{3}$$

Diffusion Models: $$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \approx \mathbf{s}_\theta(\mathbf{x}, t) = -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)}{t} = \frac{\mathbf{D}_\theta(\mathbf{x}, t) - \mathbf{x}}{t^2} \tag{4}$$

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Simplest Perspective on the Sampling of Diffusion Models**

Forward [1]: $$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \tag{1}$$

Backward (PFODE) [1]: $$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t \tag{2}$$

EDM formulation [2]: $$\mathrm{d}\mathbf{x} = -t\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\mathrm{d}t \tag{3}$$

Diffusion Models: $$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \approx \mathbf{s}_\theta(\mathbf{x}, t) = -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)}{t} = \frac{\mathbf{D}_\theta(\mathbf{x}, t) - \mathbf{x}}{t^2} \tag{4}$$

Simplified formulation: $$\mathrm{d}\mathbf{x} = \boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\mathrm{d}t \tag{5}$$

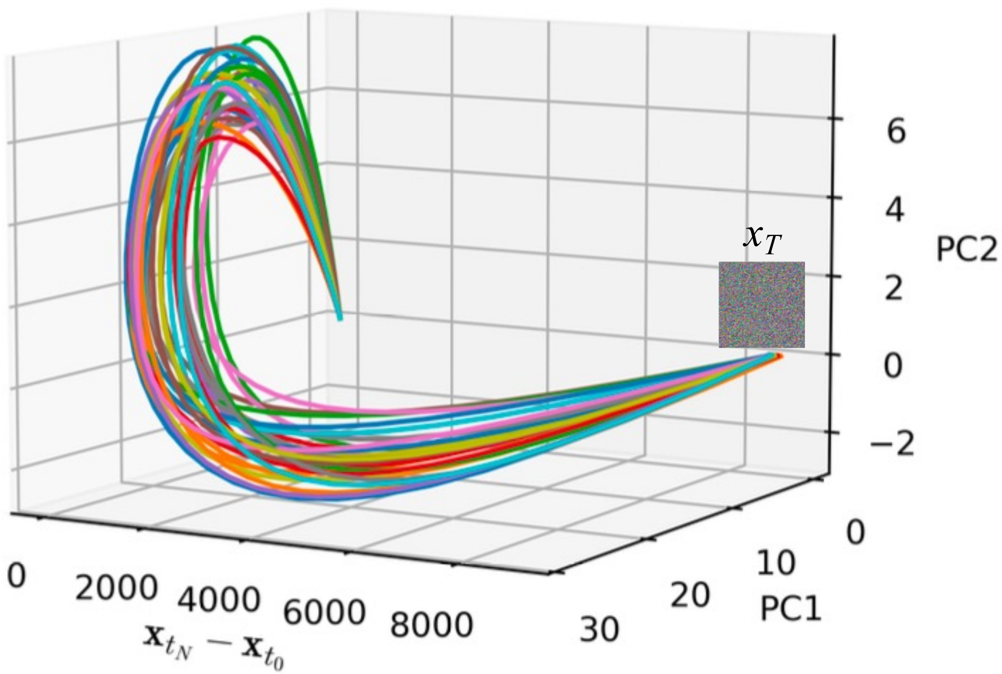
Sampling trajectory: $$\{\mathbf{x}_n\}_{n=0}^{N} \tag{6}$$

[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Karras T, Aittala M, Aila T, et al. Elucidating the design space of diffusion-based generative models[J]. Advances in Neural Information Processing Systems, 2022, 35: 26565-26577.

Preliminary

- **The Trajectory Regularity [3]**

$$\mathrm{d}\mathbf{x} = \boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\mathrm{d}t$$

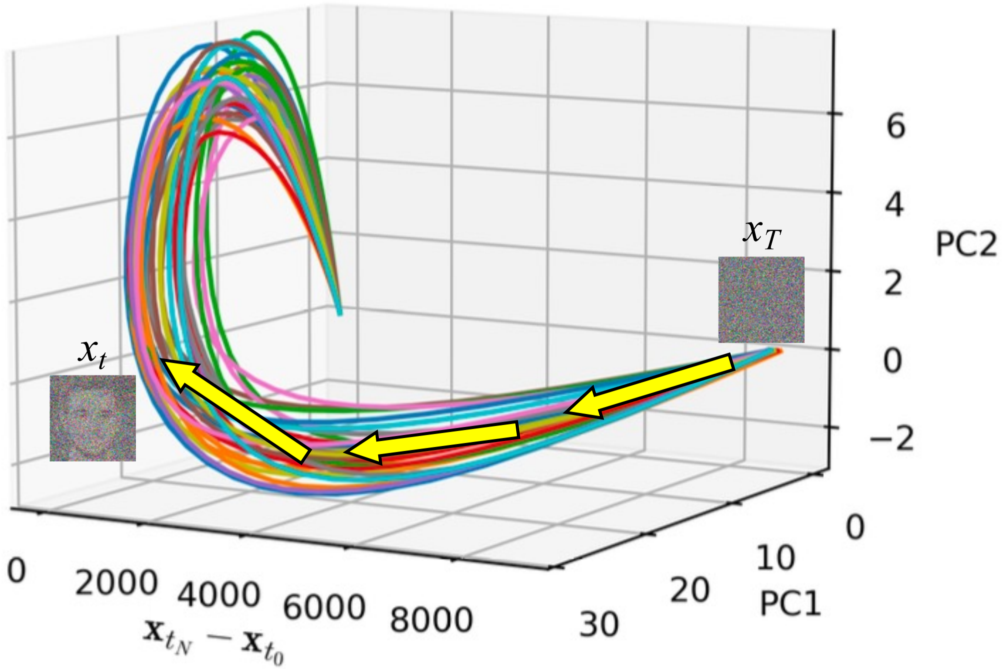


[3] Chen D, Zhou Z, Wang C, et al. On the Trajectory Regularity of ODE-based Diffusion Sampling[J]. arXiv preprint arXiv:2405.11326, 2024.

Preliminary

- **The Trajectory Regularity [3]**

$$\mathrm{d}\mathbf{x} = \boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\mathrm{d}t$$

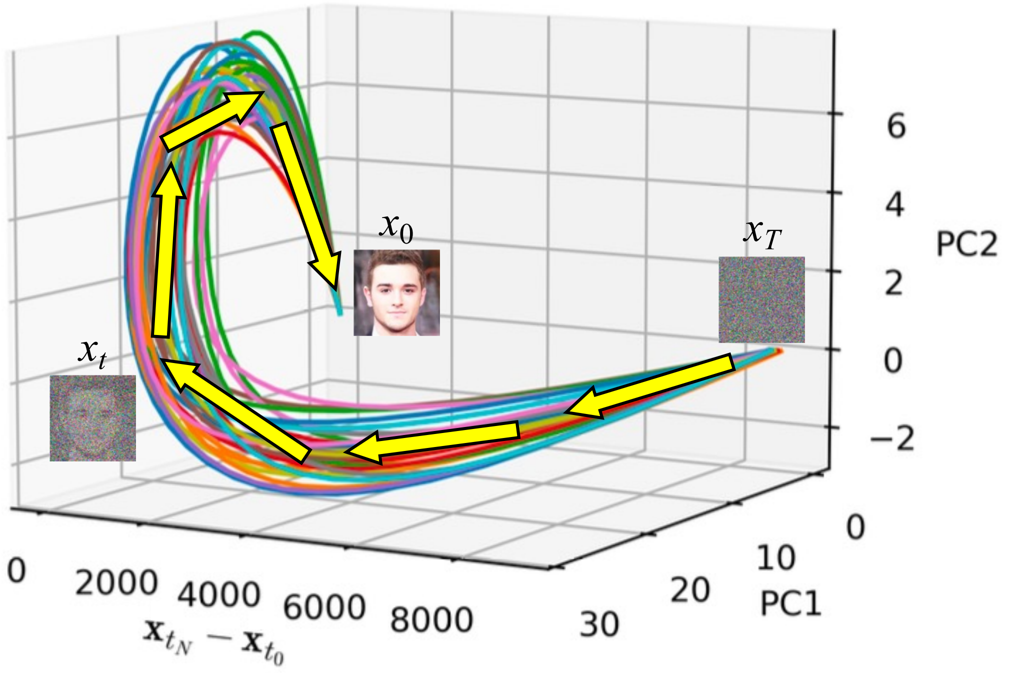


[3] Chen D, Zhou Z, Wang C, et al. On the Trajectory Regularity of ODE-based Diffusion Sampling[J]. arXiv preprint arXiv:2405.11326, 2024.

Preliminary

- **The Trajectory Regularity [3]**

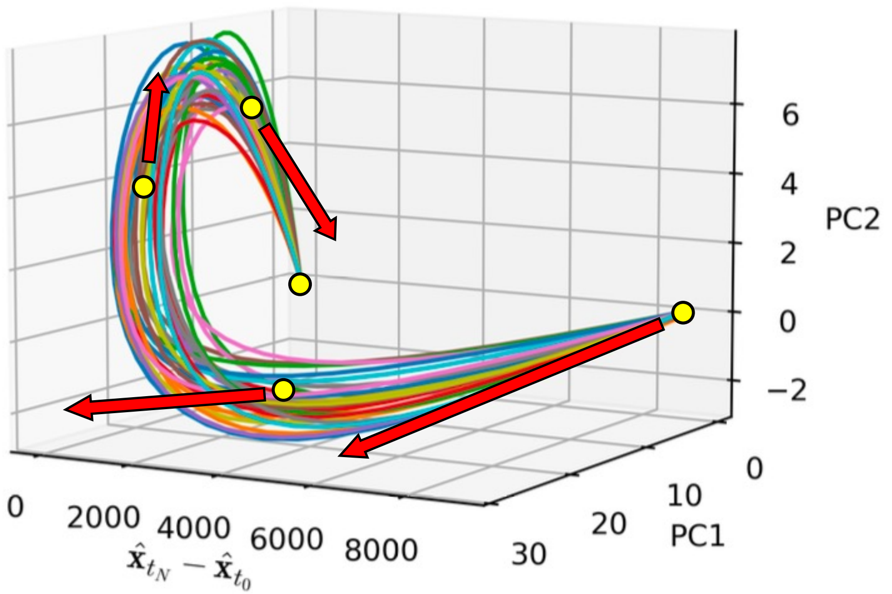
$$\mathrm{d}\mathbf{x} = \boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\mathrm{d}t$$



[3] Chen D, Zhou Z, Wang C, et al. On the Trajectory Regularity of ODE-based Diffusion Sampling[J]. arXiv preprint arXiv:2405.11326, 2024.

Motivation

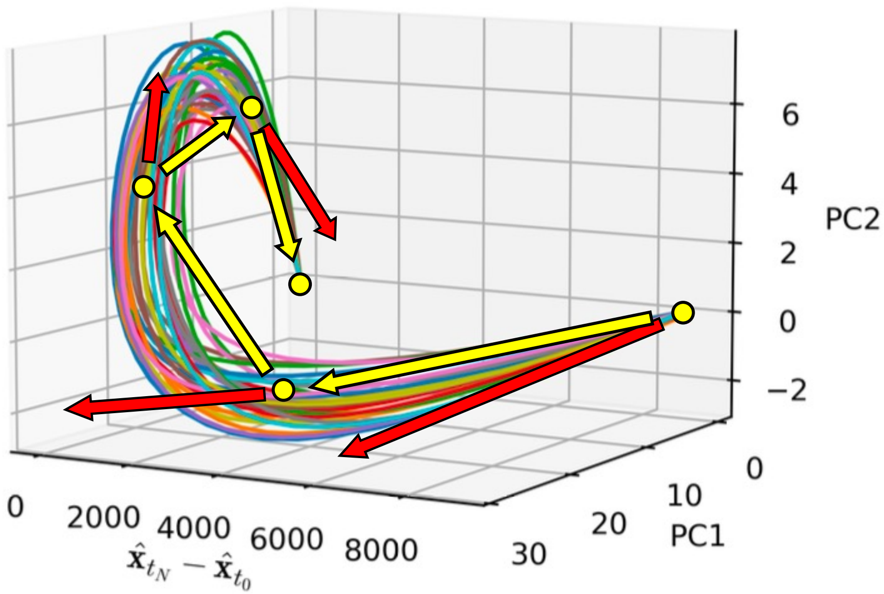
Progressive Distillation [4]



[4] Salimans T, Ho J. Progressive distillation for fast sampling of diffusion models[J]. arXiv preprint arXiv:2202.00512, 2022.

Motivation

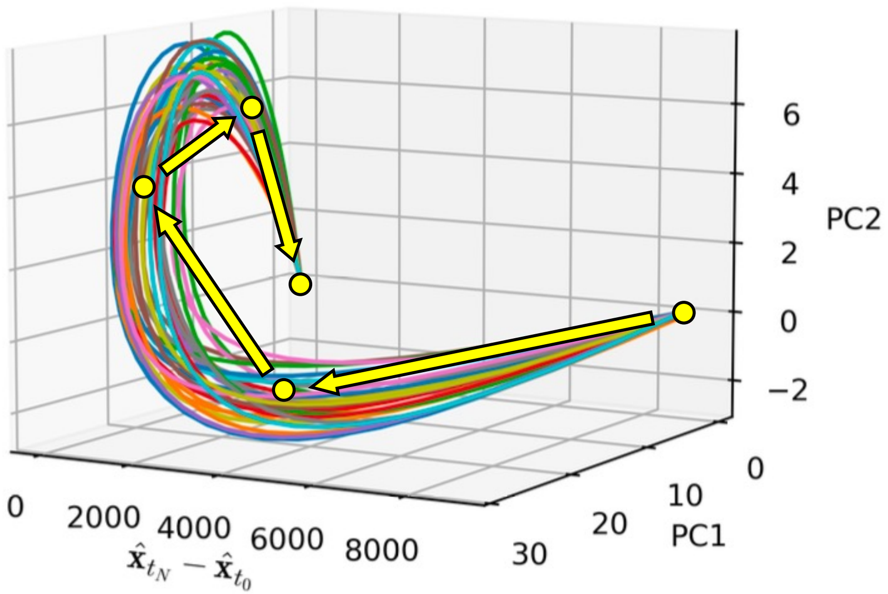
Progressive Distillation [4]



[4] Salimans T, Ho J. Progressive distillation for fast sampling of diffusion models[J]. arXiv preprint arXiv:2202.00512, 2022.

Motivation

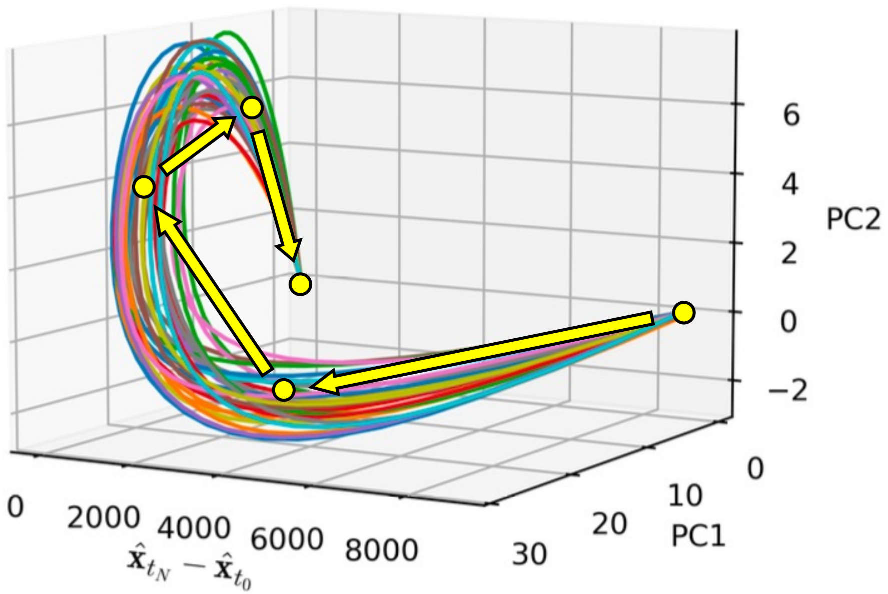
Progressive Distillation [4]



[4] Salimans T, Ho J. Progressive distillation for fast sampling of diffusion models[J]. arXiv preprint arXiv:2202.00512, 2022.

Motivation

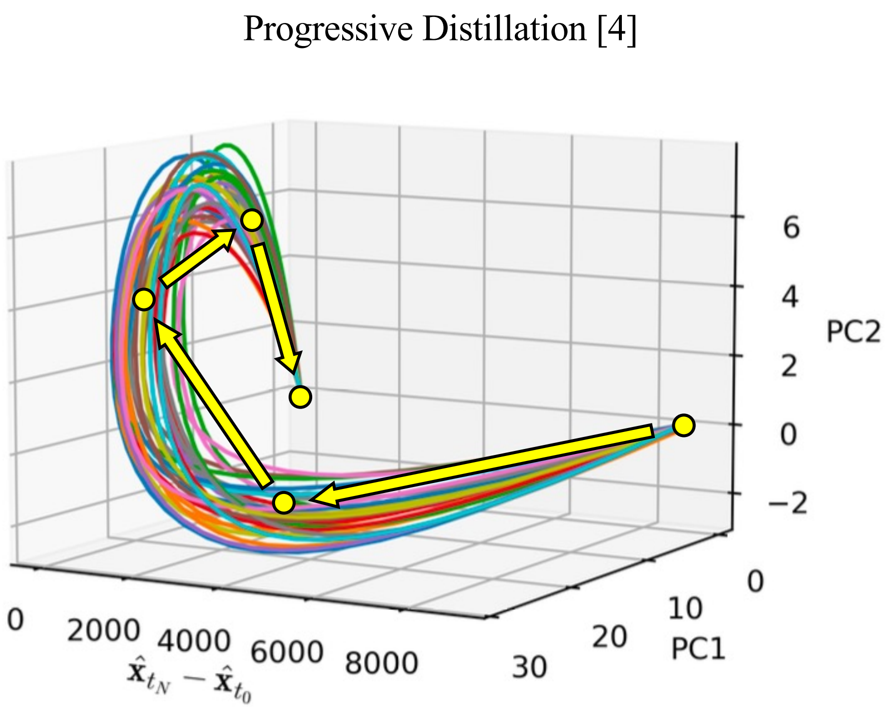
Progressive Distillation [4]



- **Problem: Time-consuming**
  - ➢ The mismatch between fine-tuning and sampling steps:
    Wasted training efforts
  - ➢ The complex optimization objectives:
    LPIPS loss, adversarial training, regularization…

[4] Salimans T, Ho J. Progressive distillation for fast sampling of diffusion models[J]. arXiv preprint arXiv:2202.00512, 2022.

Motivation



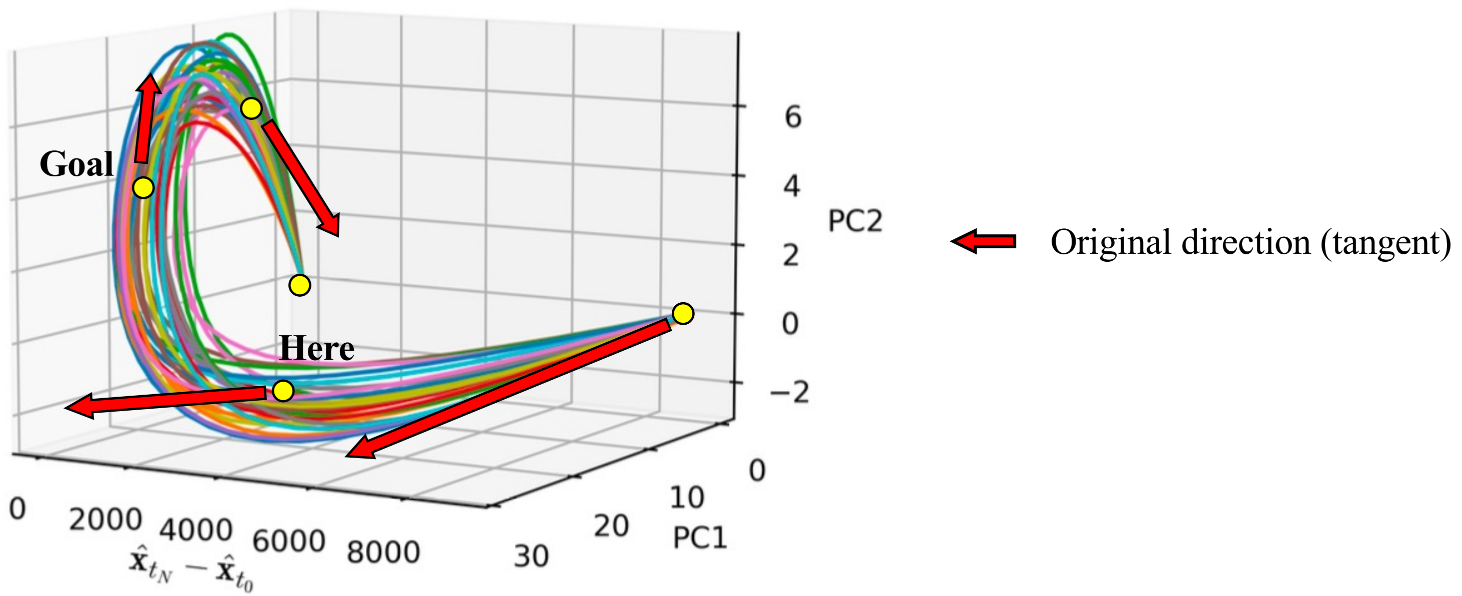
- **Problem: Time-consuming**
  - ➢ The mismatch between fine-tuning and sampling steps:
    Wasted training efforts
  - ➢ The complex optimization objectives:
    LPIPS loss, adversarial training, regularization…

- **Our Goal**
  - ➢ Simple distillation: simplified pipeline
  - ➢ Fast distillation: accelerated training

[4] Salimans T, Ho J. Progressive distillation for fast sampling of diffusion models[J]. arXiv preprint arXiv:2202.00512, 2022.

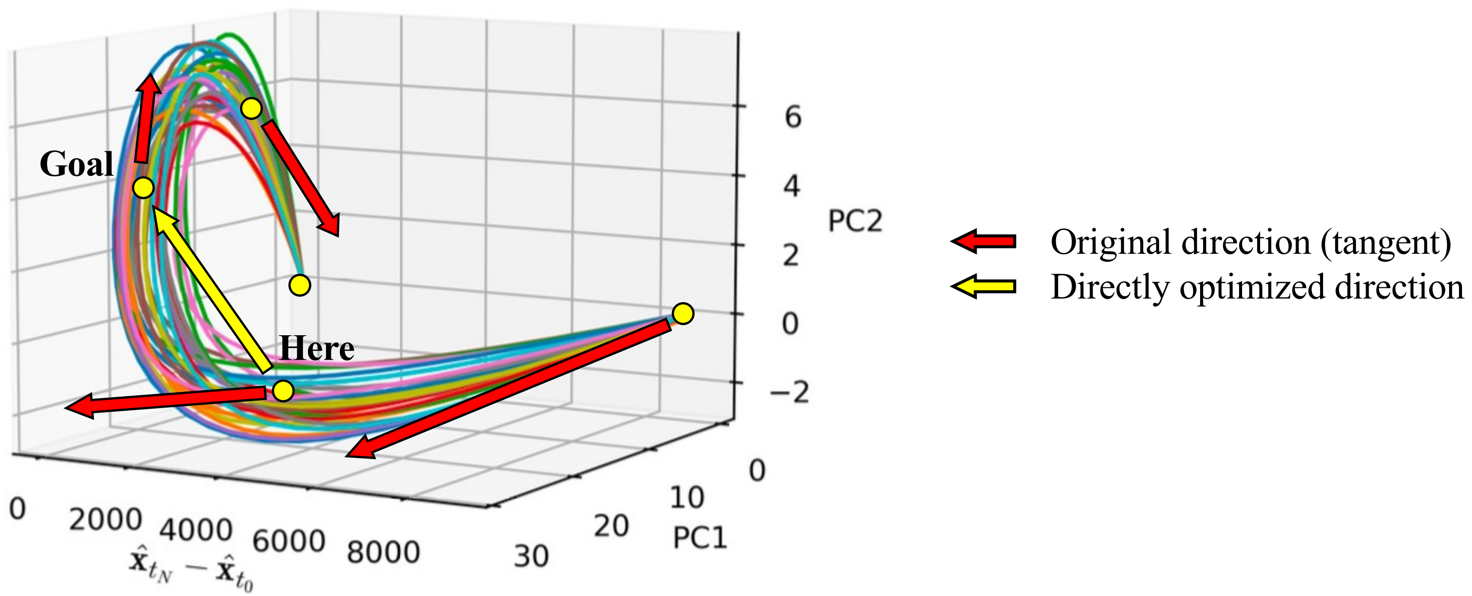
Motivation

- **The Mutual Enhancement of Fine-tuning at Different Timestamps**



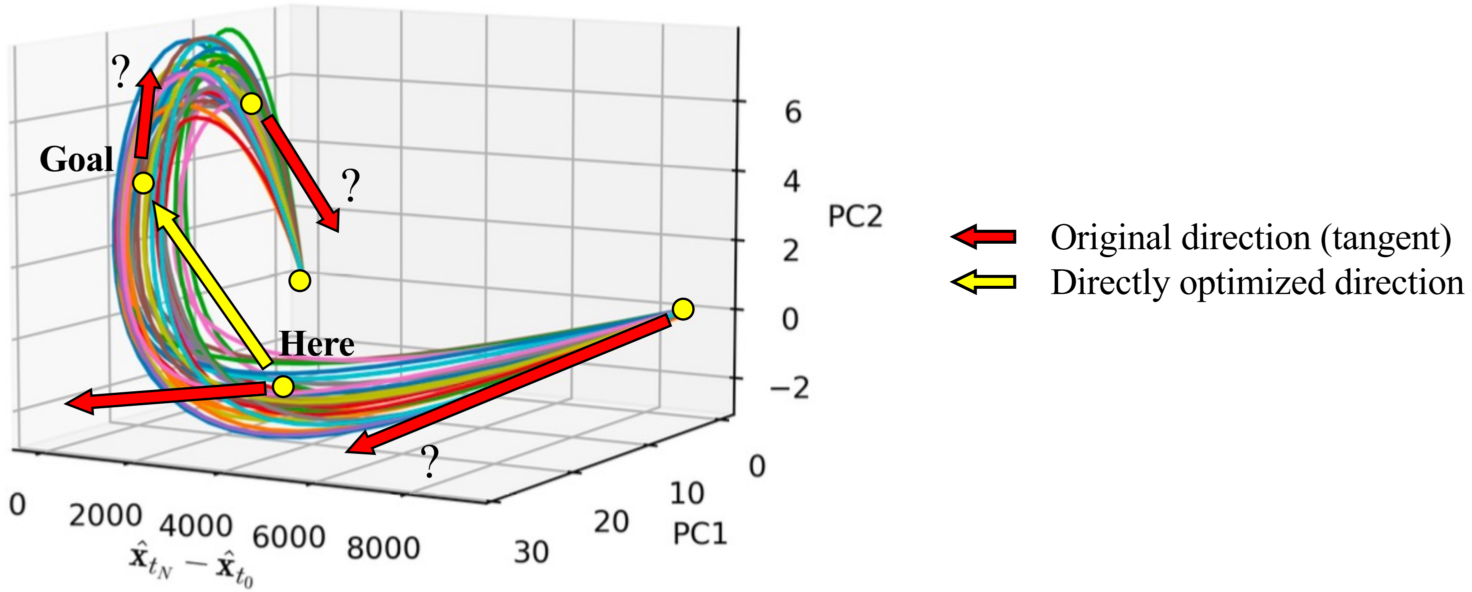
Motivation

- **The Mutual Enhancement of Fine-tuning at Different Timestamps**



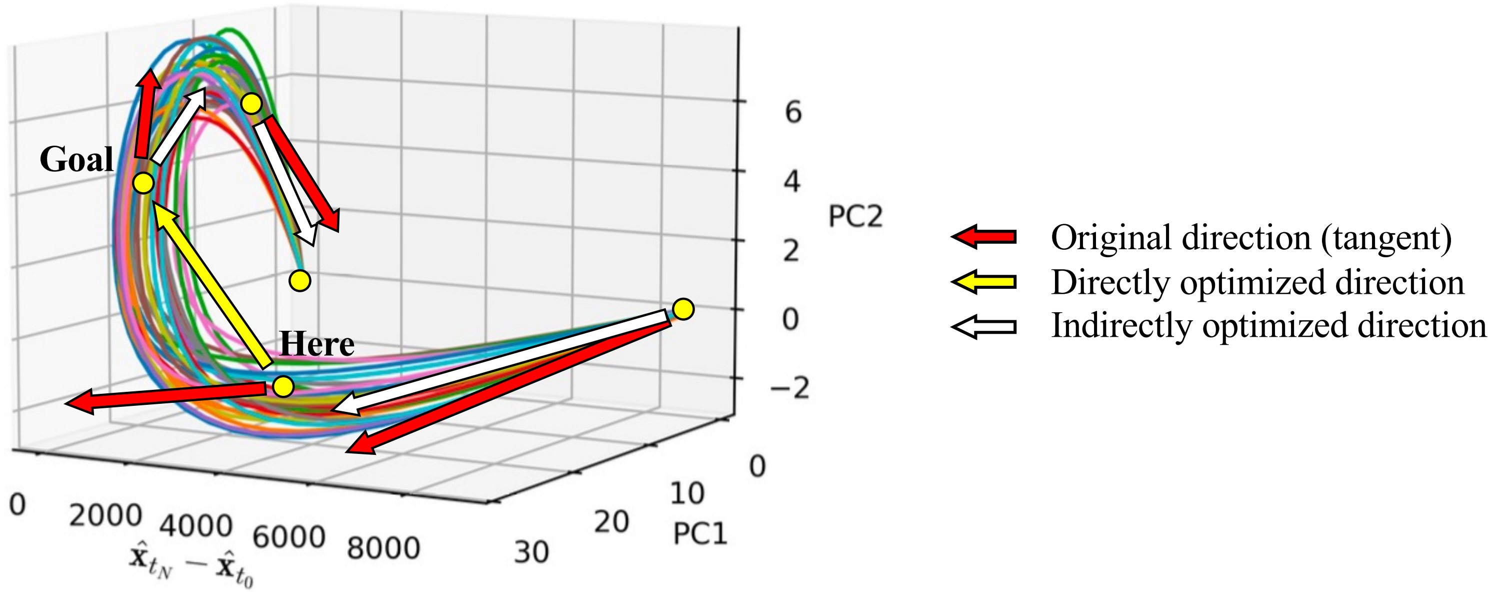
Motivation

- **The Mutual Enhancement of Fine-tuning at Different Timestamps**



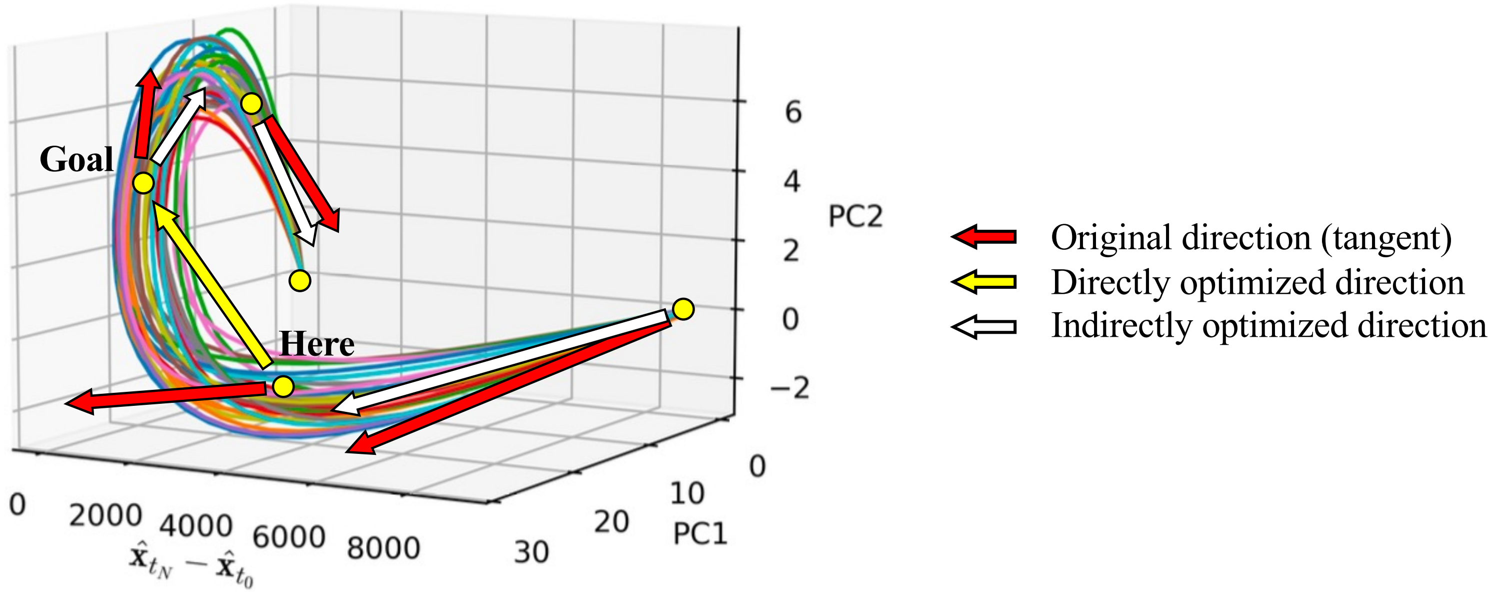
Motivation

- **The Mutual Enhancement of Fine-tuning at Different Timestamps**



Motivation

- **The Mutual Enhancement of Fine-tuning at Different Timestamps**



No need to fine-tune on a fine-grained timestamps

Method Overview

- **Ingredient 1: From Local Optimization to Global Optimization**

Algorithm 1 Trajectory Distillation

repeat
  Sample $\mathbf{x}_0$ from the dataset
  Sample $n \sim \mathcal{U}(0, N-1)$
  Sample $\mathbf{x}_{n+1} \sim \mathcal{N}(\mathbf{x}_0; t_{n+1}^2 \mathbf{I})$
  $\mathbf{x}_n^{\psi} \leftarrow \text{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
  $\tilde{\mathbf{x}}_n \leftarrow \text{Solver}(\mathbf{x}_{n+1}, t_{n+1}, t_n, K; \theta)$
  $\mathcal{L}(\psi) \leftarrow d(\mathbf{x}_n^{\psi}, \tilde{\mathbf{x}}_n)$
  $\psi \leftarrow \psi - \eta \nabla_{\psi} \mathcal{L}(\psi)$
until convergence

Algorithm 2 SFD (ours)

repeat
  Sample $\mathbf{x}_N = \tilde{\mathbf{x}}_N \sim \mathcal{N}(\mathbf{0}; t_N^2 \mathbf{I})$
  for $n = N-1$ **to** 0 **do**
    $\mathbf{x}_n^{\psi} \leftarrow \text{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
    $\tilde{\mathbf{x}}_n \leftarrow \text{Solver}(\tilde{\mathbf{x}}_{n+1}, t_{n+1}, t_n, K; \theta)$
    $\psi \leftarrow \psi - \eta \nabla_{\psi} d(\mathbf{x}_n^{\psi}, \tilde{\mathbf{x}}_n)$
    $\mathbf{x}_n \leftarrow \text{detach}(\mathbf{x}_n^{\psi})$
  end for
until convergence

  - Enable the teacher to sample more efficiently by multi-step solvers
  - Enable the student to fix accumulated errors

- **Ingredient 2: Efficient Solver for Teacher Sampling**

  - DPM++(3M) > DPM(2S) > Heun > DDIM (Euler)

Method Overview

- **Ingredient 1: From Local Optimization to Global Optimization**

Algorithm 1 Trajectory Distillation

> **repeat**
> Sample $\mathbf{x}_0$ from the dataset
> Sample $n \sim \mathcal{U}(0, N-1)$
> Sample $\mathbf{x}_{n+1} \sim \mathcal{N}(\mathbf{x}_0; t_{n+1}^2 \mathbf{I})$
> $\mathbf{x}_n^{\psi} \leftarrow \text{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
> $\tilde{\mathbf{x}}_n \leftarrow \text{Solver}(\mathbf{x}_{n+1}, t_{n+1}, t_n, K; \theta)$
> $\mathcal{L}(\psi) \leftarrow d(\mathbf{x}_n^{\psi}, \tilde{\mathbf{x}}_n)$
> $\psi \leftarrow \psi - \eta \nabla_{\psi} \mathcal{L}(\psi)$
> **until** convergence

Algorithm 2 SFD (ours)

> **repeat**
> Sample $\mathbf{x}_N = \tilde{\mathbf{x}}_N \sim \mathcal{N}(\mathbf{0}; t_N^2 \mathbf{I})$
> **for** $n = N-1$ **to** 0 **do**
> $\mathbf{x}_n^{\psi} \leftarrow \text{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
> $\tilde{\mathbf{x}}_n \leftarrow \text{Solver}(\tilde{\mathbf{x}}_{n+1}, t_{n+1}, t_n, K; \theta)$
> $\psi \leftarrow \psi - \eta \nabla_{\psi} d(\mathbf{x}_n^{\psi}, \tilde{\mathbf{x}}_n)$
> $\mathbf{x}_n \leftarrow \text{detach}(\mathbf{x}_n^{\psi})$
> **end for**
> **until** convergence

  - Enable the teacher to sample more efficiently by multi-step solvers
  - Enable the student to fix accumulated errors

- **Ingredient 2: Efficient Solver for Teacher Sampling**
  - DPM++(3M) > DPM(2S) > Heun > DDIM (Euler)

- **Ingredient 3: Minimum and Maximum Timestamps**
  - Use analytical first step (AFS) to save one sampling step

- **Ingredient 4: Loss Metric:** L1 > Pseudo-Huber > LPIPS > L2

Method	Teacher	$t_{\min}$	AFS	Loss	FID
Vanilla	Heun	0.002	N/A	L2	46.84
Vanilla	DPM(2S)	0.002	N/A	L2	16.69
SFD	Heun	0.002	False	L2	20.88
SFD	DPM(2S)	0.002	False	L2	12.50
SFD	DPM++(3M)	0.002	False	L2	11.65
SFD	DPM++(3M)	0.006	False	L2	10.93
SFD	DPM++(3M)	0.002	True	L2	7.17
SFD	DPM++(3M)	0.006	True	L2	5.67
SFD	DPM++(3M)	0.006	True	LPIPS	5.10
SFD	DPM++(3M)	0.006	True	PH	4.90
SFD	DPM++(3M)	0.006	True	L1	4.57

Method Overview

- **Ingredient 1: From Local Optimization to Global Optimization**

Algorithm 1 Trajectory Distillation

repeat
  Sample $\mathbf{x}_0$ from the dataset
  Sample $n \sim \mathcal{U}(0, N-1)$
  Sample $\mathbf{x}_{n+1} \sim \mathcal{N}(\mathbf{x}_0; t_{n+1}^2 \mathbf{I})$
  $\mathbf{x}_n^\psi \leftarrow \mathrm{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
  $\tilde{\mathbf{x}}_n \leftarrow \mathrm{Solver}(\mathbf{x}_{n+1}, t_{n+1}, t_n, K; \theta)$
  $\mathcal{L}(\psi) \leftarrow d(\mathbf{x}_n^\psi, \tilde{\mathbf{x}}_n)$
  $\psi \leftarrow \psi - \eta \nabla_\psi \mathcal{L}(\psi)$
until convergence

Algorithm 2 SFD (ours)

repeat
  Sample $\mathbf{x}_N = \tilde{\mathbf{x}}_N \sim \mathcal{N}(\mathbf{0}; t_N^2 \mathbf{I})$
  for $n = N-1$ **to** 0 **do**
    $\mathbf{x}_n^\psi \leftarrow \mathrm{Euler}(\mathbf{x}_{n+1}, t_{n+1}, t_n, 1; \psi)$
    $\tilde{\mathbf{x}}_n \leftarrow \mathrm{Solver}(\tilde{\mathbf{x}}_{n+1}, t_{n+1}, t_n, K; \theta)$
    $\psi \leftarrow \psi - \eta \nabla_\psi d(\mathbf{x}_n^\psi, \tilde{\mathbf{x}}_n)$
    $\mathbf{x}_n \leftarrow \mathrm{detach}(\mathbf{x}_n^\psi)$
  end for
until convergence

  - Enable the teacher to sample more efficiently by multi-step solvers
  - Enable the student to fix accumulated errors

- **Ingredient 2: Efficient Solver for Teacher Sampling**
  - DPM++(3M) > DPM(2S) > Heun > DDIM (Euler)
- **Ingredient 3: Minimum and Maximum Timestamps**
  - Use analytical first step (AFS) to save one sampling step
- **Ingredient 4: Loss Metric:** L1 > Pseudo-Huber > LPIPS > L2

Method	Teacher	$t_{\min}$	AFS	Loss	FID
Vanilla	Heun	0.002	N/A	L2	46.84
Vanilla	DPM(2S)	0.002	N/A	L2	16.69
SFD	Heun	0.002	False	L2	20.88
SFD	DPM(2S)	0.002	False	L2	12.50
SFD	DPM++(3M)	0.002	False	L2	11.65
SFD	DPM++(3M)	0.006	False	L2	10.93
SFD	DPM++(3M)	0.002	True	L2	7.17
SFD	DPM++(3M)	0.006	True	L2	5.67
SFD	DPM++(3M)	0.006	True	LPIPS	5.10
SFD	DPM++(3M)	0.006	True	PH	4.90
SFD	DPM++(3M)	0.006	True	L1	4.57

Consistent settings can be applied to various datasets

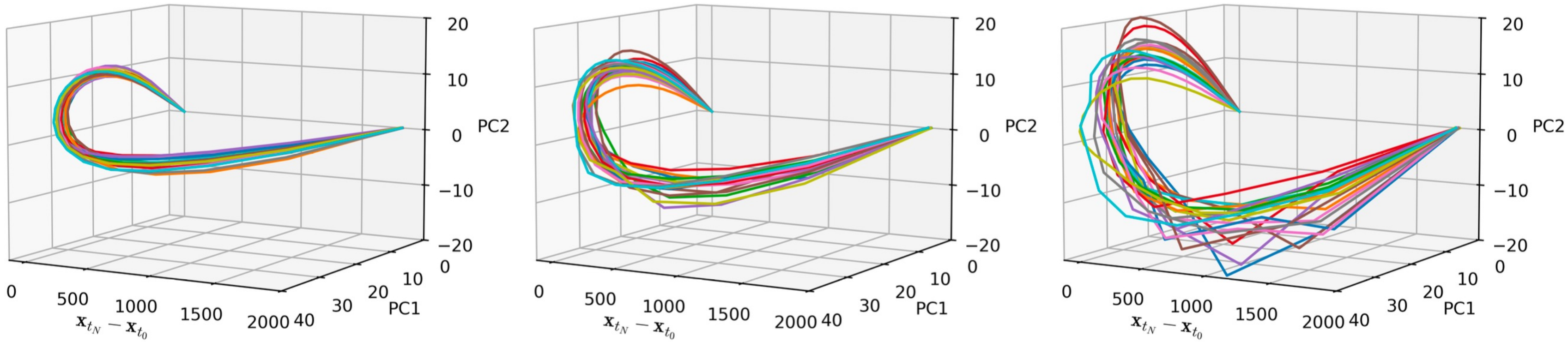
Extensions

- **Enable NFE-variable Sampling using One Model**
  - Add step-condition as a new input

Extensions

- **Enable NFE-variable Sampling using One Model**
  - Add step-condition as a new input

- **Distillation under Classifier-free Guidance**
  - Classifier-free guidance: $\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}, t, c) = \omega \boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c) + (1-\omega)\boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c=\varnothing)$
  - Distill with guidance scale of 1 and sampling with any guidance scale

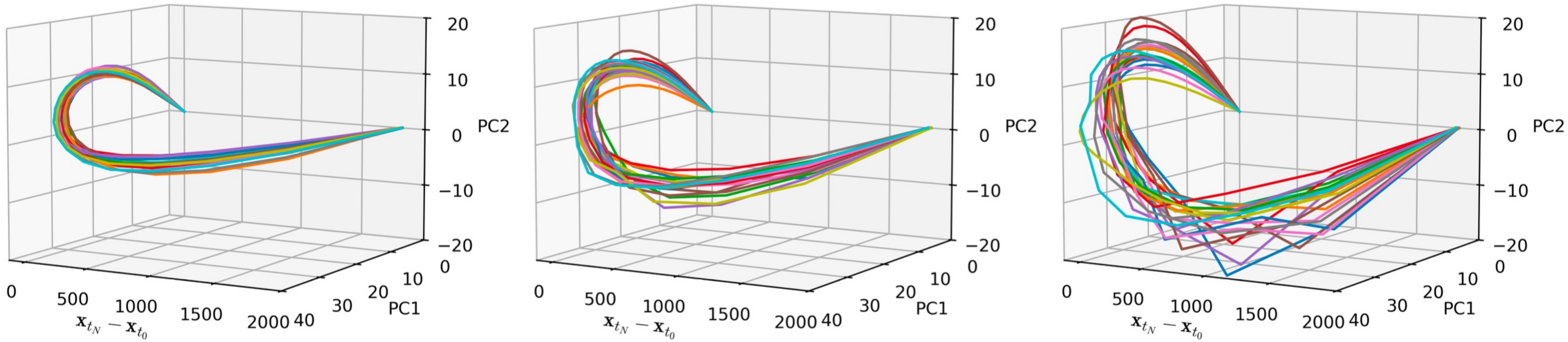


(a) Guidance scale = 1.0. (b) Guidance scale = 4.0. (c) Guidance scale = 7.5.

Extensions

- **Enable NFE-variable Sampling using One Model**
  - ➢ Add step-condition as a new input

- **Distillation under Classifier-free Guidance**
  - ➢ Classifier-free guidance: $\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}, t, c) = \omega \boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c) + (1-\omega)\boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c=\varnothing)$
  - ➢ Distill with guidance scale of 1 and sampling with any guidance scale



(a) Guidance scale = 1.0. (b) Guidance scale = 4.0. (c) Guidance scale = 7.5.

- **Second-stage Distillation for One-NFE Sampling**

Experiments: Main Results

Table 2: Results on CIFAR10 32 $\times$ 32.

Method	NFE	FID	Training time (A100 hours)
Solver-based Methods			
DDIM [44]	10	15.69	0
	50	2.91	0
DPM++(3M) [28]	5	24.97	0
	10	3.00	0
AMED-Plugin [58]	5	6.61	$\sim$ 0.08
	10	2.48	$\sim$ 0.11
GITS [4]	5	8.38	$<$ 0.01
	10	2.49	$\sim$ 0.01
Diffusion Distillation			
PD [41]	1	9.12	$\sim$ 195
	2	4.51	$\sim$ 171
Guided PD [32]	1	8.34	$\sim$ 146
	2	4.48	$\sim$ 128
	4	3.18	$\sim$ 119
CD [46]	1	3.55	$\sim$ 1156
	2	2.93	$\sim$ 1156
CTM [15]	1	1.98	$\sim$ 83
CTM [15] w/o GAN loss	1	$>$ 5	$\sim$ 60
SFD (ours) (second-stage)	1	5.83	4.88
SFD (ours)	2	4.53	0.64
	3	3.58	0.92
	4	3.24	1.17
	5	3.06	1.42
SFD-v (ours)	2	4.28	4.26
	3	3.50	
	4	3.18	
	5	2.95	

Table 3: Results on ImageNet 64 $\times$ 64.

Method	NFE	FID	Training time (A100 hours)
Solver-based Methods			
DDIM [44]	10	16.72	0
	50	4.09	0
DPM++(3M) [28]	5	25.49	0
	10	5.67	0
AMED-Plugin [58]	5	13.83	$\sim$ 0.18
	10	5.01	$\sim$ 0.32
GITS [4]	5	10.79	$<$ 0.02
	10	4.48	$\sim$ 0.02
Diffusion Distillation			
PD [41]	1	15.39	$<$ 5533
	2	8.95	$<$ 4611
Guided PD [32]	1	22.74	$<$ 5533
	2	9.75	$<$ 4611
	4	4.14	$<$ 4150
CD [46]	1	6.20	$<$ 7867
	2	4.70	$<$ 7867
CTM [15]	1	2.06	$<$ 902
	2	1.90	$<$ 902
SFD (ours) (second-stage)	1	12.89	6.86
SFD	2	10.25	3.34
	3	6.35	4.63
	4	4.99	5.98
	5	4.33	7.11
SFD-v (ours)	2	9.47	23.62
	3	5.78	
	4	4.72	
	5	4.21	

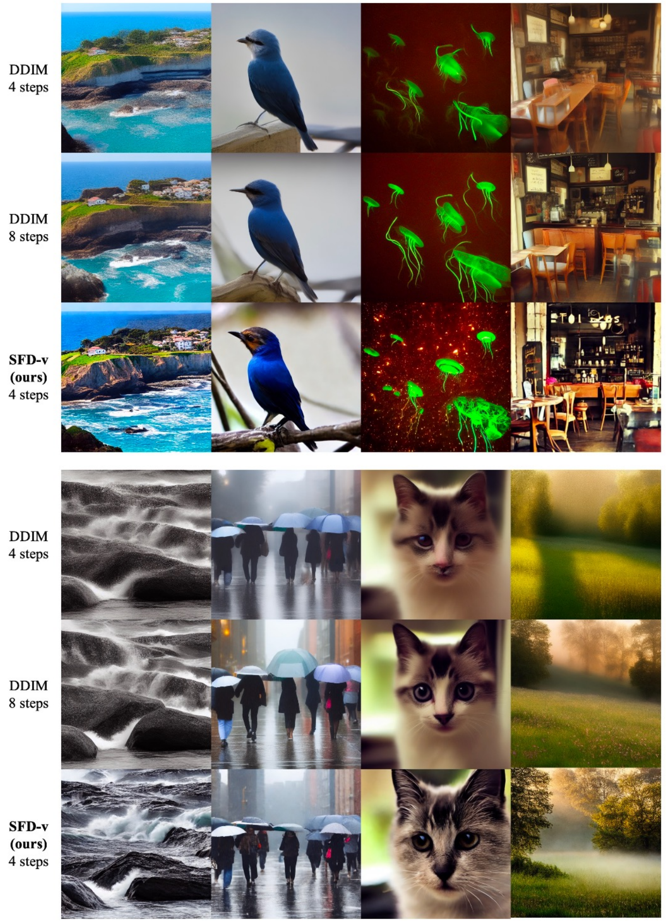
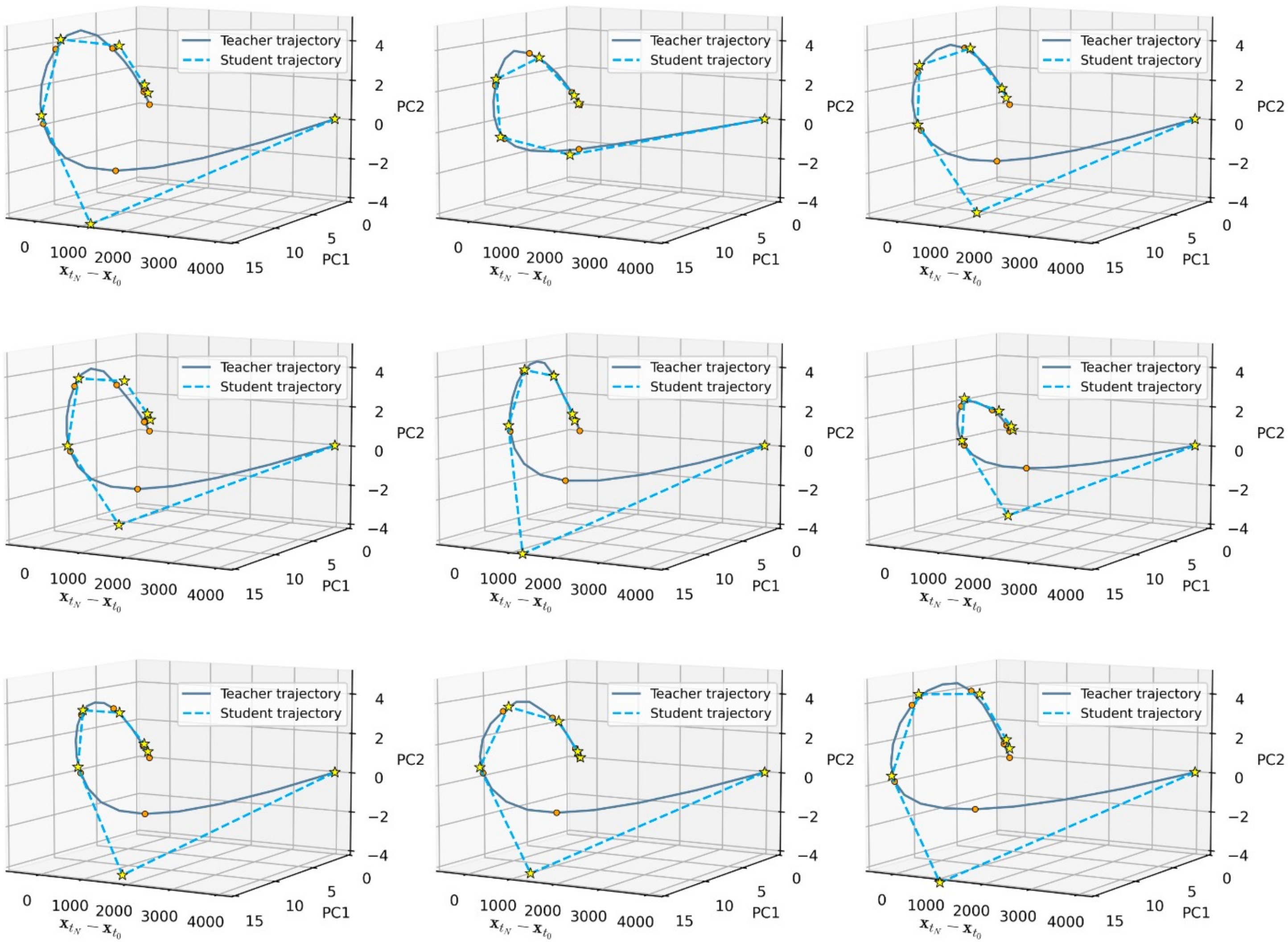


Figure 12: Qualitative results generated by Stable Diffusion v1.5 [41].

Experiments: Visualization



Thanks for watching !