Fast ODE-based Sampling for Diffusion Models in Around 5 Steps

Supplementary Material

A. Related Works

Ever since the birth of diffusion models [15, 44], their generation speed has become a major drawback compared to other generative models [12, 19]. To address this issue, efforts have been taken to accelerate the sampling of diffusion models, which fall into two main streams.

One is designing faster solvers. In early works [30, 43], the authors speed up the generation from 1000 to less than 50 NFE by reducing the number of time steps with systematic or quadratic sampling. Analytic-DPM [3], provides an analytic form of optimal variance in sampling process and improves the results. More recently, with the knowledge of interpreting the diffusion process as a PF-ODE [45], there is a class of ODE solvers based on numerical methods that accelerate the sampling process to around 10 NFE. The authors in EDM [18] achieve several improvements on training and sampling of diffusion models and propose to use Heun's second method. PDNM [23] uses linear multi-step method to solve the PF-ODE with Runge-Kutta algorithms for the warming start. The authors in [51] recommend to use lower order linear multi-step method for warming start and propose iPNDM. Given the semi-linear structure of the PF-ODE, DPM-Solver [25] and DEIS [51] are proposed by approximate the integral involved in the analytic solution of PF-ODE with Taylor expansion and polynomial extrapolation respectively. DPM-Solver is further extend to both single-step and multi-step methods in DPM-Solver++ [26]. UniPC [52] gives a unified predictor-corrector solver and improved results compared to DPM-Solver++.

Besides training-free fast solvers above, there are also solvers requiring additional training. [48] proposes a series of reparameterization for a generalized family of DDPM with a KID [5] loss. More related to our method, in GENIE [10], the authors apply the second truncated Taylor method [32] to the PF-ODE and distill a new model to predict the higher-order gradient term. Different from this, in AMED-Solver, we train a network that only predict the intermediate time steps, instead of a high-dimensional output.

Another mainstream is training-based distillation methods, which attempt to build a direct mapping from noise distribution to implicit data distribution. This idea is first introduced in [27] as an offline method where one needs to preconstruct a dataset generated by the original model. Rectified flow [24] also introduces an offline distillation based on optimal transport. For online distillation methods, one can progressively distill a diffusion model from more than 1k steps to 1 step [4, 39], or utilize the consistency property of PF-ODE trajectory to tune the denoising output [8, 13, 46].

B. Experimental Details

Datasets. We employ AMED-Solver and AMED-Plugin on a wide range of datasets and settings. We report results on settings including unconditional generation in both pixel and latent space, conditional generation with or without guidance. Datasets are chosen with image resolutions ranging from 32 to 512, including CIFAR10 32×32 [21], FFHQ 64×64 [17], ImageNet 64×64 [37] and LSUN Bedroom 256×256 [50]. We give quantitative and qualitative results generated by stable-diffusion [34] with resolution of 512. To further evaluate the effectiveness of our methods, in Appendix C, we also include more results on ImageNet 256×256 [37] with classifier guidance and latent-space LSUN Bedroom 256×256 [50].

Models. The pre-trained models we use throughout our experiments are pixel-space models from [18], [46] as well as [9], and latent-space models from [34]. Our code architecture is mainly based on the implementation in [18].

Fast ODE solvers. To give fair comparison, we reimplement several representative fast ODE solvers including DDIM [43], DPM-Solver-2 [25], multi-step DPM-Solver++ [26], UniPC [52] and improved PNDM (iPNDM) [23, 51]. Through the implementation, we obtain better or on par FID results compared with original papers. It is worth mentioning that during the implementation, we find iPNDM achieves very impressive results and outperforms other ODE solvers in many cases.

Time schedule. We notice that different ODE solvers have different preference on time schedule. We mainly use the polynomial time schedule with $\rho = 7$, which is the default setting in [18], except for DPM-Solver++ and UniPC where we use logSNR time schedule recommended in original papers [26, 52] for better results. As illustrated in Sec. 3.4, the logSNR schedule is the limit case of polynomial schedule as ρ approaches $+\infty$. Besides, for AMED-Solver on CIFAR10 32×32 [21], FFHQ 64×64 [17] and ImageNet 64×64 [37], we use uniform time schedule which is widely used in papers with a DDPM [15] backbone. This uniform time schedule is transferred from its original range $[0.001, 1]$ to $[t_1, t_N]$ in our setting following the EDM [18] implementation. For experiments on ImageNet 256×256 with classifier guidance and latent-space LSUN Bedroom, the uniform time schedule gives best results. The reason may lie in their different training process.

Training. Since there are merely 9k parameters in the AMED predictor g_ϕ, its training does not cause much computational cost. The training process spends its main time on generating student and teacher trajectories. We train g_ϕ

for 10k images, which takes 2-8 minutes on CIFAR10 and 1-3 hours on LSUN Bedroom using a single NVIDIA A100 GPU. For the distance metric in Eq. (14), we use L2 norm in all experiments. For the generation of teacher trajectories for AMED-Solver, we use DPM-Solver-2 [25] or EDM [18] with doubled NFE ($M = 1$). For AMED-Plugin, we use the same solver that generates student trajectories with $M = 1$ for DPM-Solver-2 and $M = 2$ else.

Sampling. Due to designation, our AMED-Solver or AMED-Plugin naturally create solvers with even NFE. Therefore once AFS is used, the total NFE become odd. With the goal of designing fast ODE solvers in extremely small NFE, we mainly test our method on NFE $\in \{3, 5, 7, 9\}$ where AFS is applied. There are also results on NFE $\in \{4, 6, 8, 10\}$ without using AFS.

Evaluation. We measure the sample quality via Frechet Inception Distance (FID) [14], which is a well-known quantitative evaluation metric for image quality that aligns well with human perception. For all the experiments involved, we calculate FID with 50k samples using the implementation in [18]. For Stable-Diffusion, we follow [33] and evaluate the FID value by 30k samples generated by 30k fixed prompts sampled from the MS-COCO [22] validation set.

C. Additional Results

C.1. Fast Degradation of Single-step Solvers

In pilot experiments, we find that the fast degradation of single-step solvers can be alleviated by appropriate choose of the intermediate time steps. As shown in Tab. 6, the performance of DPM-Solver-2 is very sensitive to the choice of its hyperparameter r. We apply AMED-Plugin to learn the appropriate r and find that it achieves similar results with the best searched r while adding little training overhead and negligible sampling overhead.

r	NFE			
	4	6	8	10
0.1	28.75	17.61	11.00	5.30
0.2	24.44	21.94	8.23	4.17
0.3	37.31	30.27	6.82	**3.89**
0.4	75.06	43.32	7.34	4.20
0.5 (default)	146.0	60.00	10.30	5.01
0.6	241.2	79.27	15.62	6.38
AMED-Plugin	**24.44**	**17.10**	**6.60**	4.73

Table 6. Performance of DPM-Solver-2 on CIFAR10 is sensitive to the choice of r. Applying AMED-Plugin on DPM-Solver-2 efficiently help to learn the appropriate r.

C.2. Ablation Study on Intermediate Steps

As illustrated in Sec. 3.4, for the teacher sampling trajectory, M intermediate time steps are injected in every sampling step. We use a smooth interpolation, meaning that the teacher time schedule given by the original schedule Γ combined with injected time steps, is equivalent to the time schedule obtained by simply setting the total number of time steps to $(M + 1)(N - 1) + 1$ in Eq. (10) under the same ρ. In this way, we can easily extract samples on teacher trajectories at Γ to get reference samples $\{\mathbf{y}_{t_n}\}_{n=1}^{N}$.

Here we take unconditional generation on CIFAR10 using iPNDM with AMED-Plugin as an example and provide an ablation study on the choose of M. The student and teacher solvers are set to be the same. The results are shown in Tab. 7.

M	NFE			
	3	5	7	9
1	25.98	7.53	4.01	2.77
2	**10.81**	**6.61**	**3.65**	2.63
3	11.20	7.00	4.15	2.63
4	10.92	7.40	3.69	**2.61**
5	12.87	7.70	3.65	2.75

Table 7. The sensitivity of M on CIFAR10 with AMED-Plugin on iPNDM. Additional time scaling factors are trained.

C.3. Ablation Study on Bottleneck Feature Input and Time Scaling Factor

As the U-Net bottleneck input to g_ϕ varies for different samples, the learned parameters are sample-wise, meaning that different trajectories have different time schedules. Since sampling trajectories from different starting points share similar geometric shapes, the effectiveness of inputting bottleneck might be limited. We also notice that during training, the standard deviation of learned parameters in one batch is small. Therefore, we should test if the U-Net bottleneck feature input is necessary. In Tab. 8 we replace U-Net bottleneck feature with zero matrix (w/o bottleneck) and get shared parameters across all sampling trajectories. The results show the effectiveness of the bottleneck feature input. Besides, the use of time scaling factors provides improved results. When applying our AMED-Plugin on DDIM [43], iPNDM [51] and DPM-Solver++ [26] on datasets with small resolution (32×32 and 64×64), we optionally train this time scaling factors through g_ϕ to expand the solution space.

Method	NFE			
	3	5	7	9
iPNDM [23, 51]	47.98	13.59	5.08	3.17
AMED-Plugin (w/ bottleneck)	15.87	7.29	3.92	2.82
AMED-Plugin (w/o bottleneck) †	11.12	7.31	3.80	2.64
AMED-Plugin (w/ bottleneck) †	**10.81**	**6.61**	**3.65**	**2.63**

Table 8. Ablation study of the bottleneck feature input on CIFAR10 with AMED-Plugin on iPNDM. †: additional time scaling factors are trained.

C.4. Ablation Study on AFS

The trick of analytical first step (AFS) is first introduced in [10] to reduce one NFE, where the authors replace the U-Net output in the first sampling step with the direction of x_T. In Tab. 10 and Tab. 11, we provide extended results of Tab. 2a and Tab. 2b as well as the ablations between AFS and our proposed AMED-Plugin. We find that the use of AFS provides consistent improvement on datasets with resolutions of 32 and 64. The results show that in most cases they can be considered as two independent components that can together boost the performance of various ODE solvers. However, for datasets with large resolutions, applying AFS usually causes a large degradation (see Tab. 9).

Method	AFS	NFE			
		3	5	7	9
DPM-Solver++(3M) [26]	✗	111.9	23.15	8.87	6.45
	✓	127.5	25.04	10.51	7.32
UniPC [52]	✗	112.3	23.34	8.73	6.61
	✓	127.3	25.78	10.50	7.20
iPNDM [23, 51]	✗	80.99	26.65	13.80	8.38
	✓	95.61	34.61	21.96	10.06
DPM-Solver-2 [25]	✗	210.6	80.60	23.25	9.61
	✓	241.8	88.79	22.59	9.07
AMED-Solver (**ours**)	✓	**58.51**	**13.20**	**7.10**	**5.65**

Table 9. Ablation study of AFS on pixel-space LSUN Bedroom 256×256.

C.5. More Quantitative Results

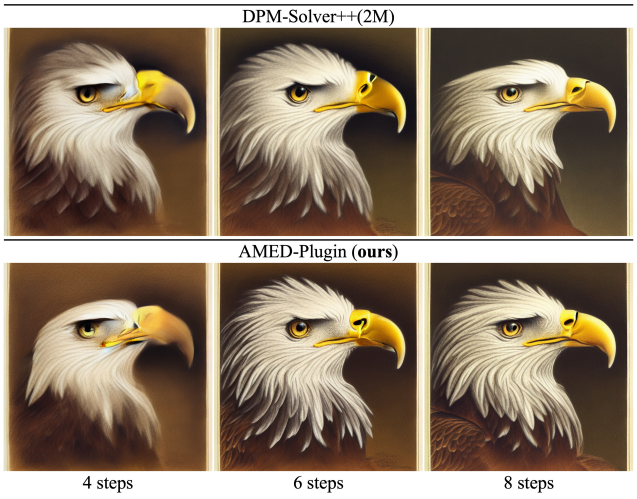
In this section, we provide additional quantitative results on more datasets including latent-space LSUN Bedroom 256×256 [34, 50] and ImageNet 256×256 [37] with classifier guidance [9]. The results are shown in Tab. 12 and Tab. 13.

C.6. More Qualitative Results

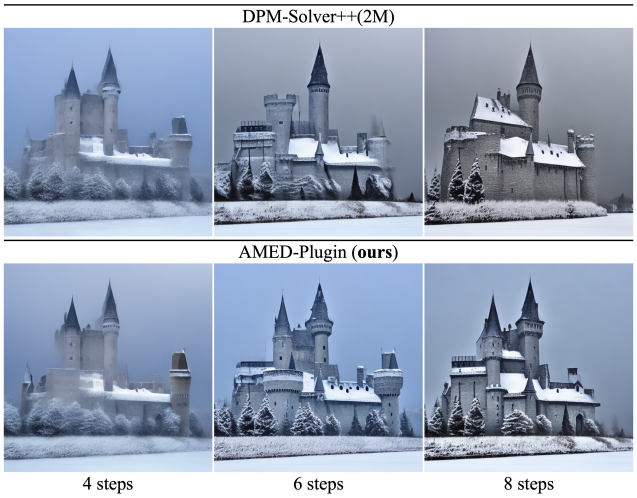
We give more qualitative results generated by stable-diffusion-v1 [34] with a default classifier-free guidance scale 7.5 in Fig. 8. Results on various datasets with NFE of 3 and 5 are provided from Fig. 10 to Fig. 15.

D. Theoretical Analysis

In Sec. 3.1, we experimentally showed that the sampling trajectory of diffusion models generated by an ODE solver almost lies in a two-dimensional subspace embedded in the ambient space. This is the core condition for the mean value theorem to approximately hold in the vector-valued function case. However, the sampling trajectory would not necessarily lie in a plane. In this section, we analyze to what extent will this affects our AMED-Solver, where we set the two-dimensional subspace to be the place spanned by the first two principal components.



(a) Text prompt: "*A portrait of an eagle*".



(b) Text prompt: "*Under a gray sky, a castle in ice and snow*".

(c) Text prompt: "*A motorcycle racer is bending*".

Figure 8. Synthesized images by Stable-Diffusion v1.4 [34] with default classifier-free guidance scale 7.5.

Method	AFS	**AMED**	NFE							
			3	4	5	6	7	8	9	10
Multi-step solvers										
UniPC [52]	✗	✗	109.6	45.20	23.98	11.14	5.83	3.99	3.21	2.89
	✓	✗	54.36	20.55	9.01	5.75	4.11	3.26	2.93	2.65
DPM-Solver++(3M) [26]†	✗	✗	110.0	46.52	24.97	11.99	6.74	4.54	3.42	3.00
	✓	✗	55.74	22.40	9.94	5.97	4.29	3.37	2.99	2.71
	✗	✓	-	21.62	-	6.82	-	4.41	-	2.76
	✓	✓	25.95	-	7.68	-	4.51	-	3.03	-
iPNDM [23, 51]†	✗	✗	47.98	24.82	13.59	7.05	5.08	3.69	3.17	2.77
	✓	✗	24.54	13.92	7.76	**5.07**	4.04	**3.22**	2.83	2.56
	✗	✓	-	**10.43**	-	6.67	-	3.34	-	**2.48**
	✓	✓	**10.81**	-	**6.61**	-	**3.65**	-	**2.63**	-
Single-step solvers										
DDIM [43]†	✗	✗	93.36	66.76	49.66	35.62	27.93	22.32	18.43	15.69
	✓	✗	67.26	49.96	35.78	28.00	22.37	18.48	15.69	13.47
	✗	✓	-	37.72	-	25.15	-	17.03	-	11.33
	✓	✓	38.23	-	24.44	-	15.72	-	10.93	-
DPM-Solver-2 [25]	✗	✗	-	146.0	-	60.00	-	10.30	-	5.01
	✓	✗	155.7	-	57.28	-	10.20	-	4.98	-
	✗	✓	-	24.44	-	17.10	-	6.60	-	4.73
	✓	✓	38.48	-	28.14	-	7.46	-	4.73	-
AMED-Solver (**ours**)	✗	✓	-	**17.18**	-	**7.04**	-	**5.56**	-	**4.14**
	✓	✓	**18.49**	-	**7.59**	-	**4.36**	-	**3.67**	-

Table 10. Unconditional generation on CIFAR10 32×32. †: additional time scaling factors $\{a_n\}_{n=1}^{N-1}$ are trained.

Method	AFS	**AMED**	NFE							
			3	4	5	6	7	8	9	10
Multi-step solvers										
UniPC [52]	✗	✗	91.38	55.63	24.36	14.30	9.57	7.52	6.34	5.53
	✓	✗	64.54	29.59	16.17	11.03	8.51	6.98	6.04	5.26
DPM-Solver++(3M) [26]†	✗	✗	91.52	56.34	25.49	15.06	10.14	7.84	6.48	5.67
	✓	✗	65.20	30.56	16.87	11.38	8.68	7.12	6.25	5.58
	✗	✓	-	53.28	-	13.68	-	7.98	-	5.57
	✓	✓	76.51	-	15.21	-	8.36	-	6.04	-
iPNDM [23, 51]	✗	✗	58.53	33.79	18.99	12.92	9.17	7.20	5.91	5.11
	✓	✗	34.81	**21.32**	15.53	**10.27**	8.64	**6.60**	5.64	**4.97**
	✗	✓	-	23.55	-	12.05	-	7.03	-	5.01
	✓	✓	**28.06**	-	**13.83**	-	**7.81**	-	**5.60**	-
Single-step solvers										
DDIM [43]†	✗	✗	82.96	58.43	43.81	34.03	27.46	22.59	19.27	16.72
	✓	✗	62.42	46.06	35.48	28.50	23.31	19.82	17.14	15.02
	✗	✓	-	40.85	-	32.46	-	20.72	-	15.52
	✓	✓	46.10	-	33.54	-	21.94	-	15.56	-
DPM-Solver-2 [25]	✗	✗	-	129.8	-	44.83	-	12.42	-	6.84
	✓	✗	140.2	-	42.41	-	12.03	-	6.64	-
	✗	✓	-	40.99	-	31.19	-	11.24	-	6.94
	✓	✓	70.64	-	29.96	-	11.54	-	6.91	-
AMED-Solver (**ours**)	✗	✓	-	**32.69**	-	**10.63**	-	**7.71**	-	**6.06**
	✓	✓	**38.10**	-	**10.74**	-	**6.66**	-	**5.44**	-

Table 11. Conditional generation on ImageNet 64×64. †: additional time scaling factors $\{a_n\}_{n=1}^{N-1}$ are trained.

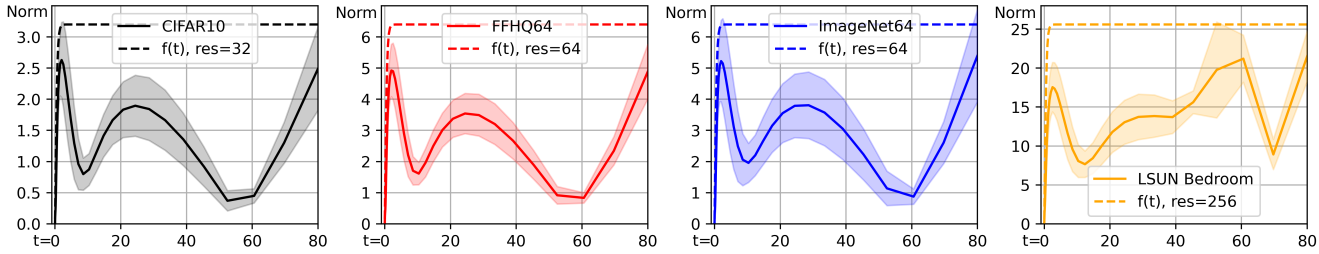


Figure 9. Following the experiment in Sec. 3.1, we calculate $\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2$ and find that we can bound it by a proper setting of Eq. (16).

Method	NFE			
	4	6	8	10
DPM-Solver++(3M) [26]	48.55	10.01	4.61	3.62
AMED-Plugin (**ours**)	**15.67**	**8.92**	**4.19**	**3.52**

Table 12. Unconditional generation on latent-space LSUN Bedroom. AMED-Plugin is applied on DPM-Solver++.

Method	NFE			
	4	6	8	10
Guidance scale = 8.0				
DPM-Solver++(3M) [26]	60.01	25.51	**11.98**	**7.95**
AMED-Plugin (**ours**)	**39.84**	**21.79**	13.94	9.05
Guidance scale = 4.0				
DPM-Solver++(3M) [26]	27.15	10.25	7.10	6.15
AMED-Plugin (**ours**)	**24.19**	**8.86**	**6.54**	**5.72**
Guidance scale = 2.0				
DPM-Solver++(3M) [26]	**23.06**	10.17	7.04	5.92
AMED-Plugin (**ours**)	28.81	**9.56**	**6.42**	**5.46**

Table 13. Conditional generation on ImageNet256 with classifier guidance. AMED-Plugin is applied on DPM-Solver++.

Notations. Denote d as the dimension of the ambient space. Let $\{\mathbf{x}_\tau\}_{\tau=\epsilon}^T$ to be the solution of the PF-ODE Eq. (6). Let $\{\tilde{\mathbf{x}}_\tau\}_{\tau=\epsilon}^T$ to be the trajectory obtained by projecting $\{\mathbf{x}_\tau\}_{\tau=\epsilon}^T$ to the two-dimensional subspace spanned by its first two principal components. Given $\epsilon \le s < m < t \le T$ and a constant c, one step of the AMED-Solver is given by

$$\mathbf{x}_s^A = \mathbf{x}_t + c(s-t)\boldsymbol{\epsilon}_\theta(\mathbf{x}_m, m). \tag{15}$$

Define the scaled logistic function to be

$$f(\tau) = a\left(\frac{1}{1+e^{-b\tau}} - \frac{1}{2}\right), \tau \in \mathbb{R}; a, b \in \mathbb{R}^+. \tag{16}$$

Finally, define a SDE

$$\mathrm{d}\mathbf{z}_\tau = g(\tau)\mathrm{d}\mathbf{w}_\tau, \tau \in [s, t] \tag{17}$$

with initial value $\mathbf{0}$ at t where $g(\tau)$ is a real-valued function and $\mathbf{w}_\tau \in \mathbb{R}^d$ is the standard Wiener process.

We start by the following assumptions:

Assumption 1. *Assume that there exists $a, b > 0$ s.t. $\|\mathbf{x}_\tau - \tilde{\mathbf{x}}_\tau\|_2 \le f(\tau), \tau \in [\epsilon, T]$.*

For the choice of a and b, in Fig. 9, we calculate $\|\mathbf{x}_\tau - \tilde{\mathbf{x}}_\tau\|_2$ following the experiment settings in Sec. 3.1. We experimentally find that $\|\mathbf{x}_\tau - \tilde{\mathbf{x}}_\tau\|_2$ can be roughly upper bounded by setting $a = \sqrt{3d}/15$ and $b = 3$.

Assumption 2. *Assume that there exists an integrable function $\boldsymbol{\omega} : \mathbb{R}^{d+1} \to \mathbb{R}^d$ that generate $\{\tilde{\mathbf{x}}_\tau\}_{\tau=\epsilon}^T$ by*

$$\tilde{\mathbf{x}}_s = \tilde{\mathbf{x}}_t + \int_t^s \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau, \tau) + \boldsymbol{\omega}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau. \tag{18}$$

with initial value $\tilde{\mathbf{x}}_T$. In this way, we can decompose the integral in Eq. (7) into two components that parallel and perpendicular to the plane where $\{\tilde{\mathbf{x}}_\tau\}_{\tau=\epsilon}^T$ lies, i.e.,

$$\begin{aligned}\int_t^s \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau, \tau)\mathrm{d}\tau = &\int_t^s \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau, \tau)\boldsymbol{\omega}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau \\ &- \int_t^s \boldsymbol{\omega}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau.\end{aligned} \tag{19}$$

Assumption 3. *Decompose $\boldsymbol{\epsilon}_\theta(\mathbf{x}_m, m)$ in Eq. (15) into two components as in Assumption 2 that parallel and perpendicular to the plane where $\{\tilde{\mathbf{x}}_\tau\}_{\tau=\epsilon}^T$ lies:*

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_m, m) = \boldsymbol{\epsilon}_\theta^{\parallel}(\mathbf{x}_m, m) + \boldsymbol{\epsilon}_\theta^{\perp}(\mathbf{x}_m, m). \tag{20}$$

Assume that the parallel component is optimally learned and for the perpendicular component, we have

$$\left\|c(s-t)\boldsymbol{\epsilon}_\theta^{\perp}(\mathbf{x}_m, m)\right\|_2 \le \left\|\int_t^s \boldsymbol{\omega}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau\right\|_2. \tag{21}$$

Assumption 4. *There exists such a $g(\tau)$ s.t. with high probability that*

$$\left\|\int_t^s \boldsymbol{\omega}(\tilde{\mathbf{x}}_\tau, \tau)\mathrm{d}\tau\right\|_2 \le \|\mathbf{z}_s\|_2. \tag{22}$$

Lemma 1. *Under assumption 1 and 4, let $g(\tau) = f(\tau)/\sqrt{d}$, then $\mathbf{z}_s$ concentrates at a thin shell with radius*

$$r(s,t) = \frac{a}{\sqrt{b}}\sqrt{\frac{1}{1+e^x}\Big|_{bs}^{bt} + \frac{b}{4}(t-s)}. \tag{23}$$

Proof. Since the SDE Eq. (17) has zero drift coefficient, its perturbation kernel $p(\mathbf{z}_s|\mathbf{z}_t = \mathbf{0})$ is a Gaussian with zero mean [41]. The covariance $\mathbf{P}(s,t)$ is given by

$$\mathbf{P}(s,t) = \int_s^t g^2(\tau)\mathrm{d}\tau\mathbf{I} \tag{24}$$

$$= \frac{a^2}{d}\int_s^t \left(\frac{1}{1+e^{-b\tau}} - \frac{1}{2}\right)^2 \mathrm{d}\tau\mathbf{I} \tag{25}$$

$$= \underbrace{\frac{a^2}{bd}\left(\frac{1}{1+e^x}\Big|_{bs}^{bt} + \frac{b}{4}(t-s)\right)}_{\sigma^2(s,t)}\mathbf{I}. \tag{26}$$

By the well-known concentration of measure [47], there exists a constant $c > 0$ s.t. for any $h \geq 0$, we have

$$\mathbb{P}\left(\left|\|\mathbf{z}_s\|_2 - |\sigma(s,t)|\sqrt{d}\right| \geq h\right) \leq 2e^{-ch^2} \tag{27}$$

which complete the proof. □

Proposition 1. *Given $\epsilon \leq s < t \leq T$, under the assumptions and Lemma 1 above, with high probability we have*

$$\left\|\mathbf{x}_s - \mathbf{x}_s^A\right\|_2 \leq f(s) + f(t) + r(s,t). \tag{28}$$

Proof. Under assumptions and Lemma 1 above, we have

$$\left\|\mathbf{x}_s - \mathbf{x}_s^A\right\|_2 \tag{29}$$

$$\leq \|\mathbf{x}_s - \tilde{\mathbf{x}}_s\|_2 + \left\|\tilde{\mathbf{x}}_s - \mathbf{x}_s^A\right\|_2 \tag{30}$$

$$\leq f(s) + \Big\|\mathbf{x}_t - \tilde{\mathbf{x}}_t + c(s-t)\boldsymbol{\epsilon}_\theta(\mathbf{x}_m, m) \tag{31}$$

$$- \int_t^s \boldsymbol{\epsilon}_\theta(\mathbf{x}_\tau,\tau) + \boldsymbol{\omega}(\mathbf{x}_\tau,\tau)\mathrm{d}\tau\Big\|_2 \tag{32}$$

$$\leq f(s) + f(t) + \Big\|c(s-t)\boldsymbol{\epsilon}_\theta^\perp(\mathbf{x}_m, m) \tag{33}$$

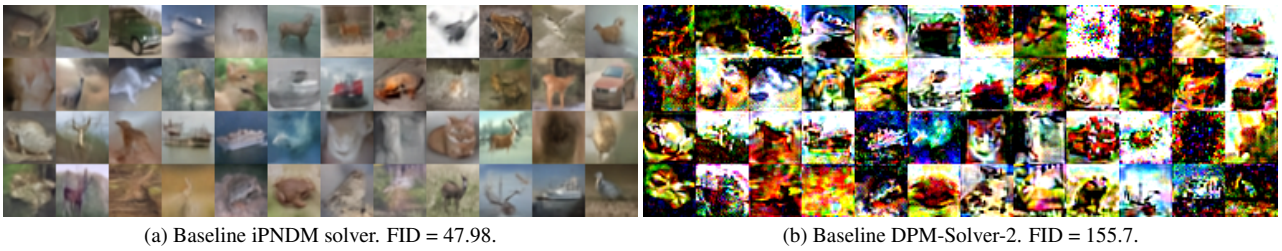
$$+ c(s-t)\boldsymbol{\epsilon}_\theta^\parallel(\mathbf{x}_m, m) - \int_t^s (\boldsymbol{\epsilon}_\theta + \boldsymbol{\omega})(\mathbf{x}_\tau,\tau)\mathrm{d}\tau\Big\|_2 \tag{34}$$

$$\leq f(s) + f(t) + \left\|c(s-t)\boldsymbol{\epsilon}_\theta^\perp(\mathbf{x}_m, m)\right\|_2 \tag{35}$$

$$\leq f(s) + f(t) + \left\|\int_t^s \boldsymbol{\omega}(\mathbf{x}_\tau,\tau)\mathrm{d}\tau\right\|_2 \tag{36}$$

$$\leq f(s) + f(t) + r(s,t) \tag{37}$$

with high probability. □



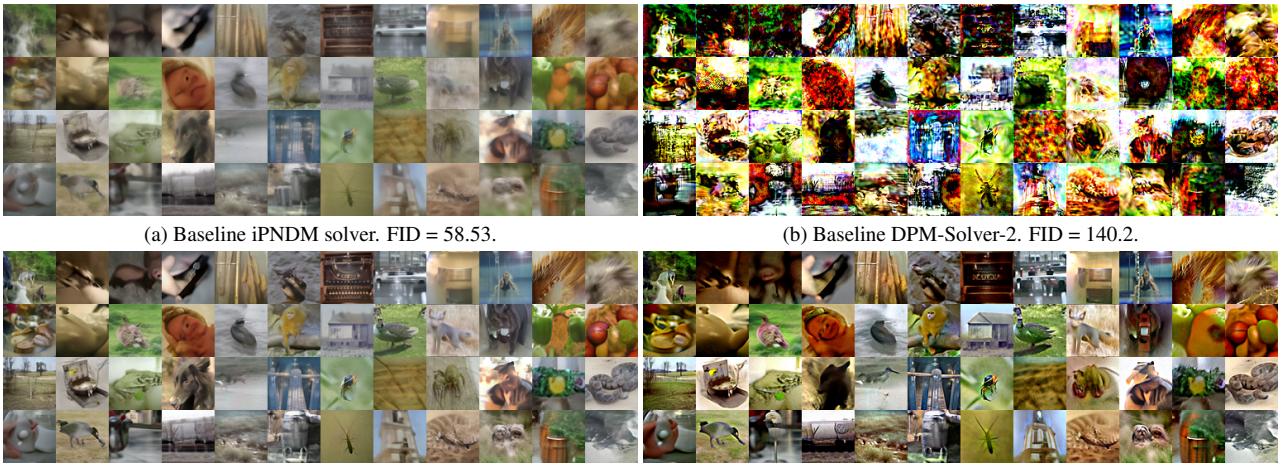
(a) Baseline iPNDM solver. FID = 47.98.

(b) Baseline DPM-Solver-2. FID = 155.7.

(c) AMED-Plugin applied on iPNDM solver. FID = 10.81.

(d) AMED-Solver. FID = 18.49.

Figure 10. Uncurated samples on CIFAR10 32×32 with 3 NFE.



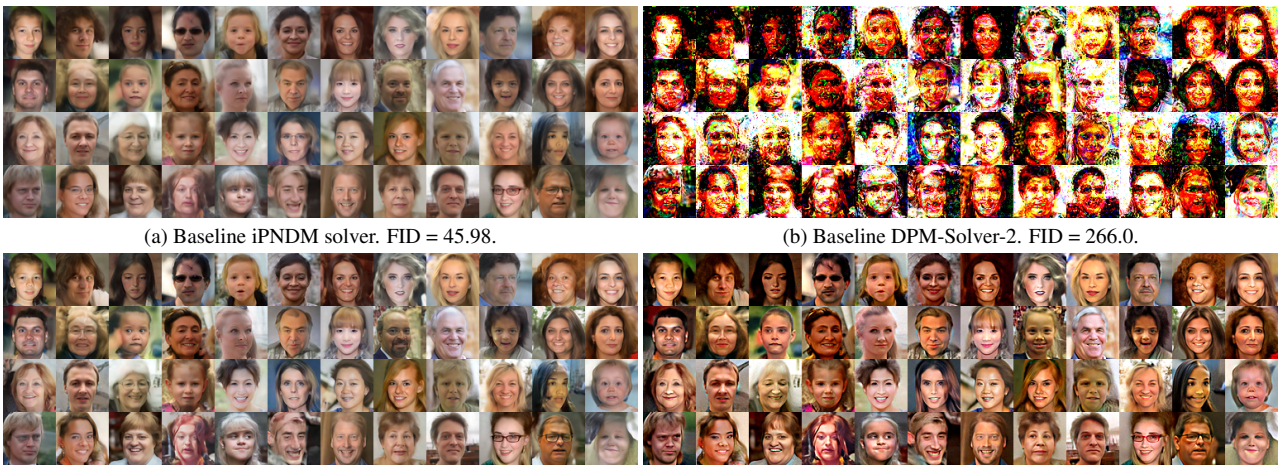
(a) Baseline iPNDM solver. FID = 58.53.

(b) Baseline DPM-Solver-2. FID = 140.2.

(c) AMED-Plugin applied on iPNDM solver. FID = 28.06.

(d) AMED-Solver. FID = 38.10.

Figure 11. Uncurated samples on Imagenet 64×64 with 3 NFE.



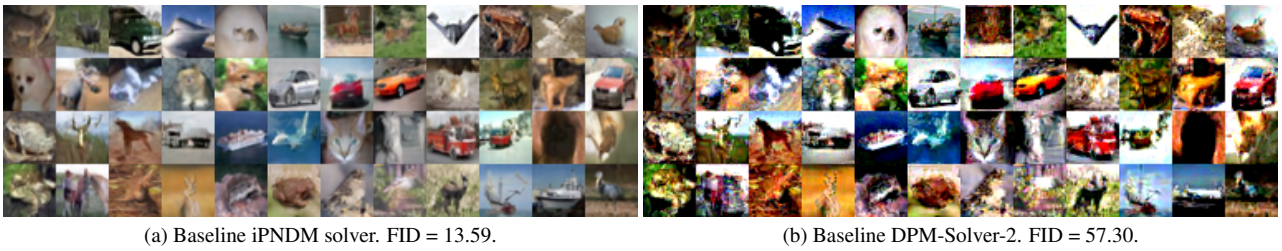
(a) Baseline iPNDM solver. FID = 45.98.

(b) Baseline DPM-Solver-2. FID = 266.0.

(c) AMED-Plugin applied on iPNDM solver. FID = 26.87.

(d) AMED-Solver. FID = 47.31.

Figure 12. Uncurated samples on FFHQ 64×64 with 3 NFE.



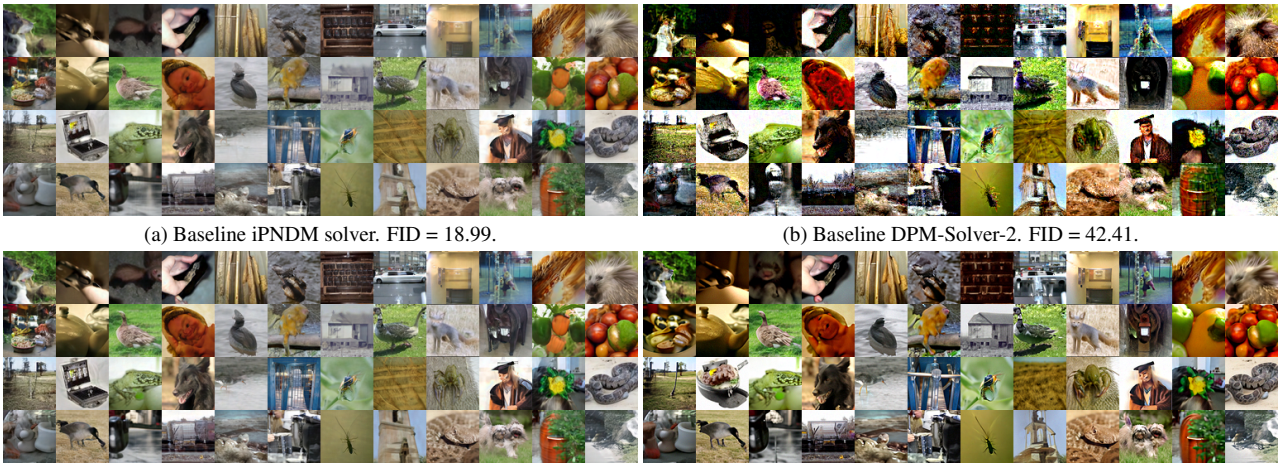
(a) Baseline iPNDM solver. FID = 13.59.

(b) Baseline DPM-Solver-2. FID = 57.30.

(c) AMED-Plugin applied on iPNDM solver. FID = 6.61.

(d) AMED-Solver. FID = 7.59.

Figure 13. Uncurated samples on CIFAR10 32×32 with 5 NFE.



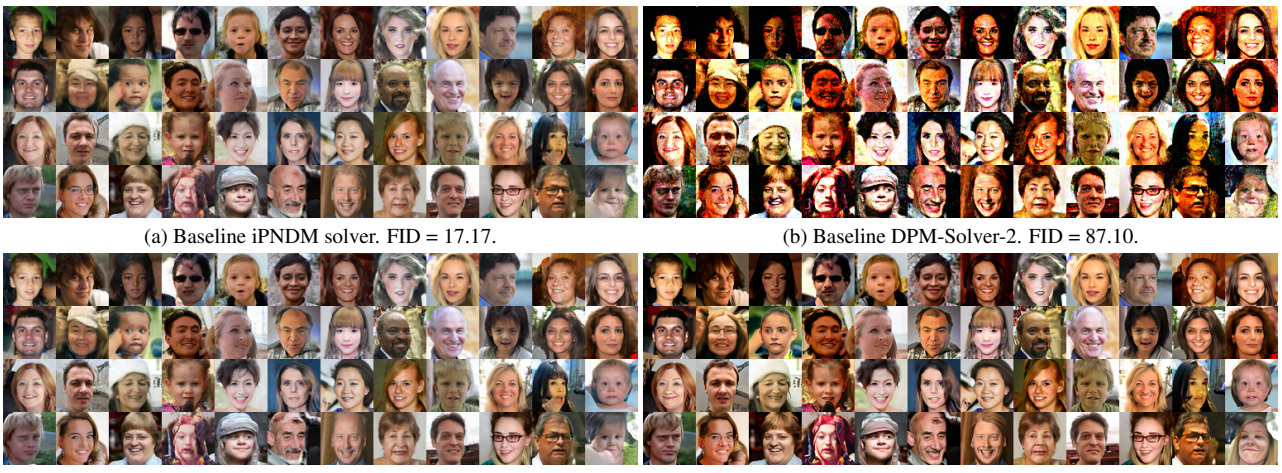
(a) Baseline iPNDM solver. FID = 18.99.

(b) Baseline DPM-Solver-2. FID = 42.41.

(c) AMED-Plugin applied on iPNDM solver. FID = 13.83.

(d) AMED-Solver. FID = 10.74.

Figure 14. Uncurated samples on Imagenet 64×64 with 5 NFE.



(a) Baseline iPNDM solver. FID = 17.17.

(b) Baseline DPM-Solver-2. FID = 87.10.

(c) AMED-Plugin applied on iPNDM solver. FID = 12.49.

(d) AMED-Solver. FID = 14.80.

Figure 15. Uncurated samples on FFHQ 64×64 with 5 NFE.