SA-Solver: Stochastic Adams Solver for Fast Sampling of Diffusion Models

Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, Zhi-Ming Ma

Diffusion Models

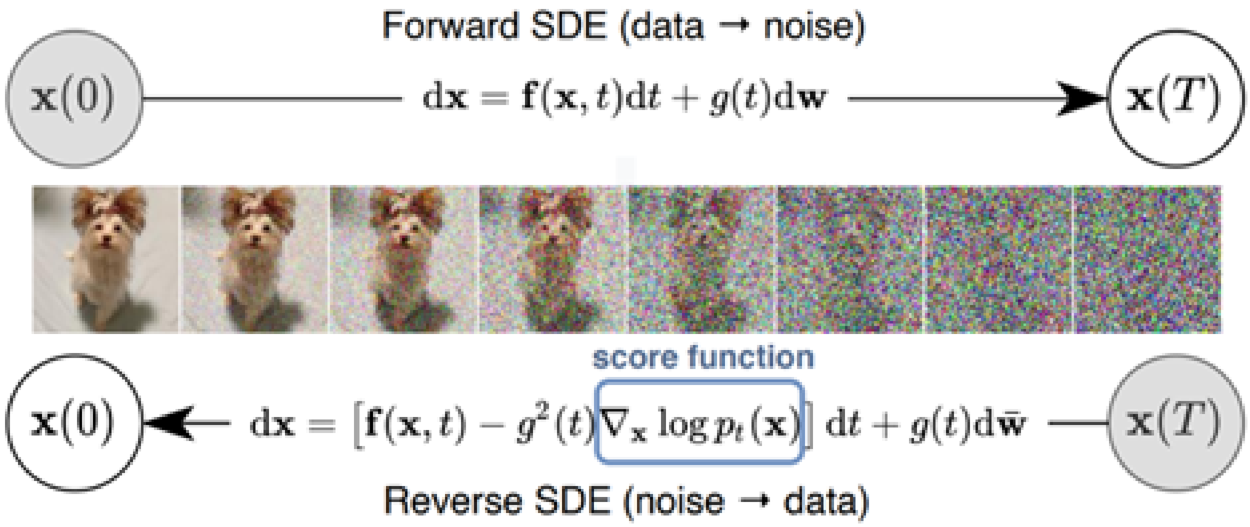


Image from Song et al., 2020

- Getting noise from data is easy (Forward SDE).
- Generating data by reversing the forward process.

Estimating the score function by Denoising Score matching (Vincent 2010).

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \Big\{ \lambda(t) \mathbb{E}_{\boldsymbol{x}_0} \mathbb{E}_{\boldsymbol{x}_t | \boldsymbol{x}_0} \big[\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t | \boldsymbol{x}_0) \|_2^2 \big] \Big\}$$

Motivation: Beyond Diffusion ODE and Diffusion SDE

- Diffusion ODE

$$\mathrm{d}\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right]\mathrm{d}t, \qquad \boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$$

$\updownarrow$

- Deterministic sampler: DDIM, PNDM, DEIS, DPM-Solver, UniPC.

Pros: Faster convergence.
Cons: Sub-optimal results when NFE is large.

- Diffusion SDE

$$\mathrm{d}\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t, \qquad \boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$$

$\updownarrow$

- Stochastic sampler: DDPM, Gotta Go Fast, Analytic-DPM.

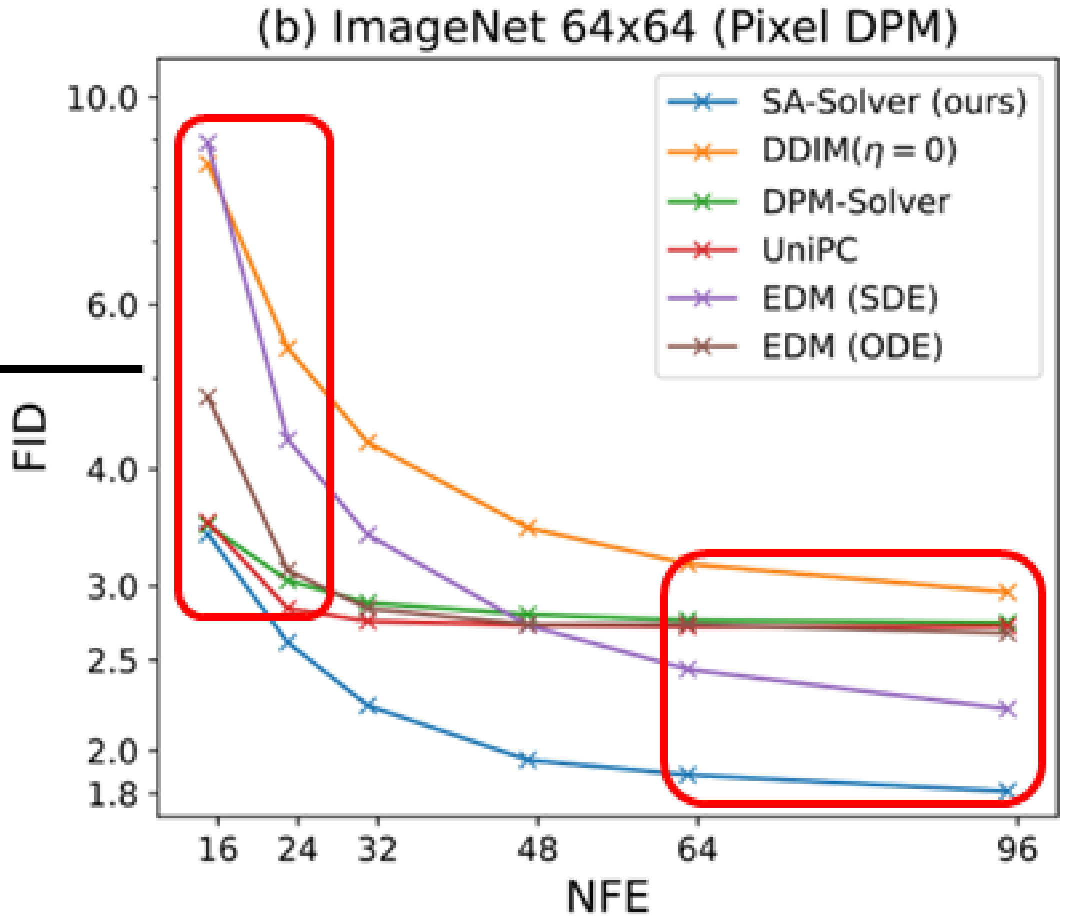
Pros: Slower convergence.
Cons: Optimal results when NFE is large.

Can we design sampling algorithm which shares fast convergence and optimal sampling results?

SA-Solver: Stochastic Adams Solver for Fast Sampling of Diffusion Models

Training-free!

Fast Convergence
(small NFEs)

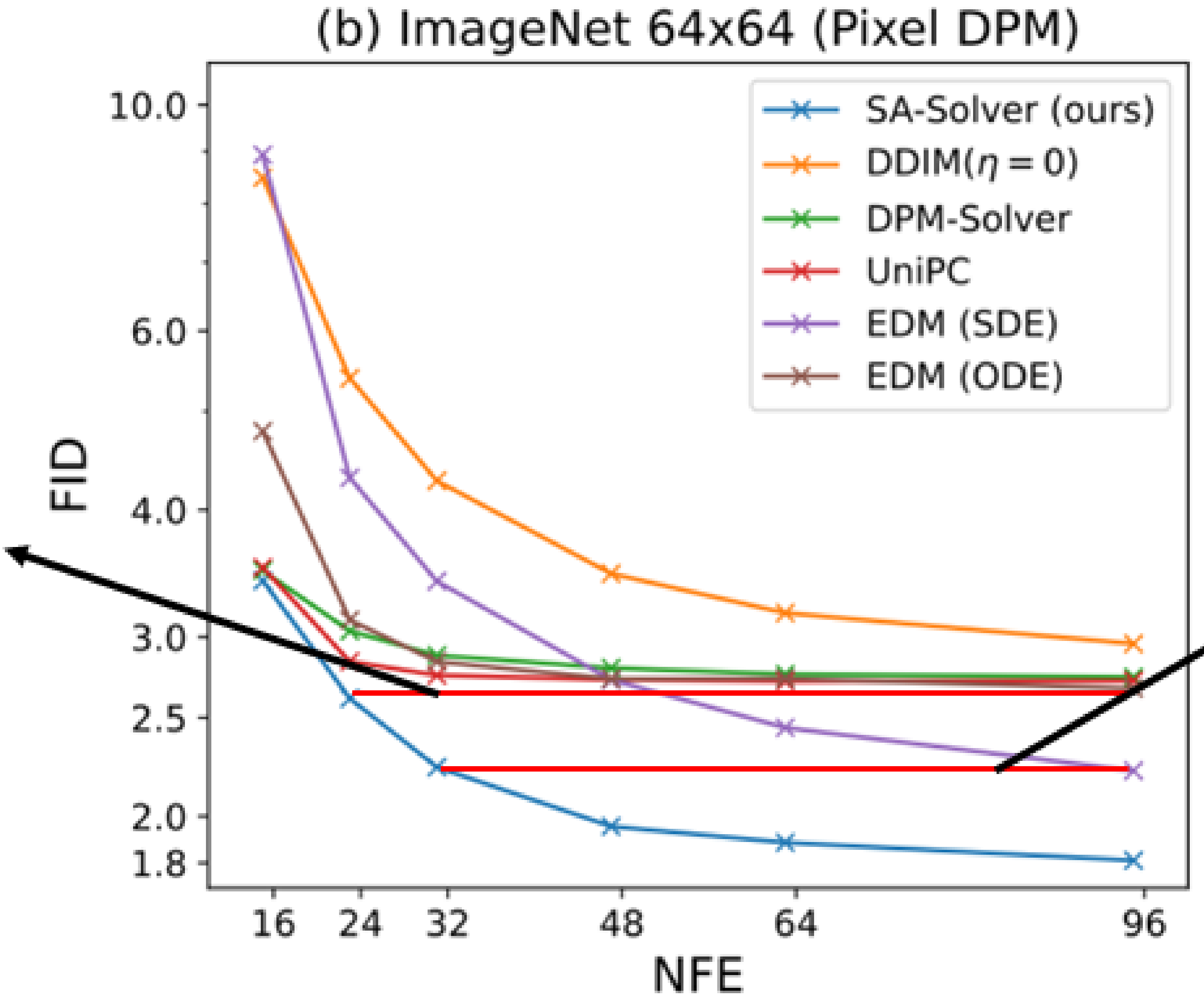


Optimal Result
(large NFEs)

Can we design sampling algorithm which shares fast convergence and optimal sampling results?
Yes!

SA-Solver: Stochastic Adams Solver for Fast Sampling of Diffusion Models

Approximately 4× acceleration than previous SOTA ODE-based sampler



Approximately 3× acceleration than previous SOTA SDE-based sampler

Can we design sampling algorithm which shares fast convergence and optimal sampling results?
Yes!

Contribution 1: Variance-Controlled Diffusion SDE

- Variance-Controlled Diffusion SDE, which extends the reverse process of diffusion models.

$$\mathrm{d}\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t - \left(\frac{1+\boxed{\tau^2(t)}}{2} \right) g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) \right] \mathrm{d}t + \boxed{\tau(t)} g(t) \mathrm{d}\bar{\boldsymbol{w}}_t, \quad \boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$$

- Degenerate to Diffusion ODE or Diffusion SDE when $\tau(t) = 0$ or $\tau(t) = 1$.

- Shares the same marginal probability distribution with Diffusion ODE and Diffusion SDE!

- With Variance-Controlled Diffusion SDE, we can add proper scale of noise in the sampling process with limited NFEs.

Contribution 2: Deriving analytical solution for Variance-Controlled Diffusion SDE.

- Variance-Controlled Diffusion SDE

$$\mathrm{d}\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t - \left(\frac{1+\tau^2(t)}{2} \right) g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) \right] \mathrm{d}t + \tau(t) g(t) \mathrm{d}\bar{\boldsymbol{w}}_t, \quad \boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$$

- Compute its analytical solution

$$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s} e^{-\int_{\lambda_s}^{\lambda_t} \tau^2(\tilde{\lambda}) \mathrm{d}\tilde{\lambda}} \boldsymbol{x}_s + \sigma_t \boldsymbol{F}_{\boldsymbol{\theta}}(s,t) + \sigma_t \boldsymbol{G}(s,t)$$

→ Follows a normal distribution with analytical variance which can be directly simulated!

↓ Exponential Integrator (Semi-linear structure in SDE)

↓

$$\sigma_t \boldsymbol{G}(s,t) \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_t^2 \left(1 - e^{-2\int_{\lambda_s}^{\lambda_t} \tau^2(\tilde{\lambda}) \mathrm{d}\tilde{\lambda}}\right)\right)$$

Contribution 3: Introducing Stochastic Adams Method as discretization scheme for SDEs.

- Variance-Controlled Diffusion SDE

$$\mathrm{d}\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t - \left(\frac{1 + \boxed{\tau^2(t)}}{2} \right) g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) \right] \mathrm{d}t + \boxed{\tau(t)} g(t) \mathrm{d}\bar{\boldsymbol{w}}_t, \quad \boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$$

- How to solve this SDE?

- Stochastic Adams Method utilizes 1 Model Evaluations per step; Better for Limited NFEs
- Stochastic Runge-Kutta Method utilizes 2/3/more (depends on order) Model Evaluations per step;

- Additional improvement:

- Exponential Integrator and Analytical computed variance in Contribution 2.
- Using Stochastic Adams-Bashforth Method (Explicit Method) as Predictor and Stochastic Adams-Moulton Method as Corrector to incorporate with Predictor-Corrector Method.

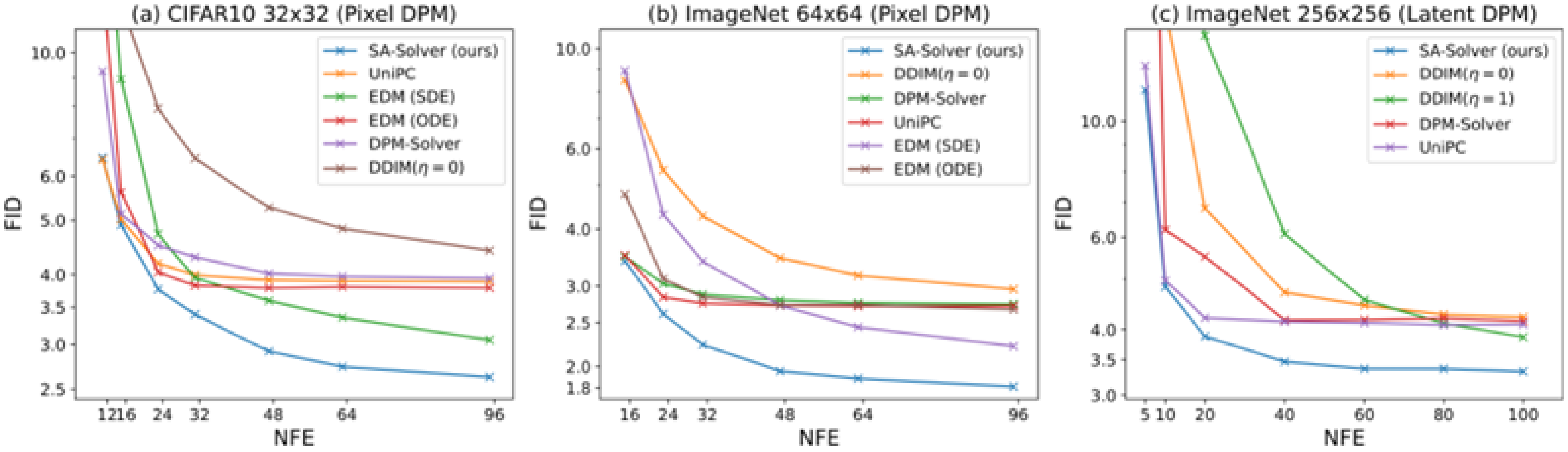
Overall pipeline

Algorithm 1 *SA-Solver*

Require: data prediction model $\boldsymbol{x_\theta}$, timesteps $\{t_i\}_{i=0}^{M}$, initial value $\boldsymbol{x}_{t_0}$, predictor step s_p, corrector step s_c, buffer B to store former evaluation of $\boldsymbol{x_\theta}$, $\tau(t)$ to control variance.

```
B ←buffer— x_θ(x_{t_0}, t_0)
for i = 1 to max(s_p, s_c) do                                              ▷ Warm-up
    sample ξ ~ N(0, I)
    calculate steps for warm-up s_p^m = min(i, s_p), s_c^m = min(i, s_c)
    x_{t_i}^p ← s_p^m-step SA-Predictor(x_{t_{i-1}}, B, ξ) (Eq. (14))       ▷ Prediction Step
    B ←buffer— x_θ(x_{t_i}^p, t_i)                                          ▷ Evaluation Step
    x_{t_i} ← s_c^m-step SA-Corrector(x_{t_i}^p, x_{t_{i-1}}, B, ξ) (Eq. (17)) ▷ Correction Step
for i = max(s_p, s_c) + 1 to M do
    sample ξ ~ N(0, I)
    x_{t_i}^p ← s_p-step SA-Predictor(x_{t_{i-1}}, B, ξ) (Eq. (14))         ▷ Prediction Step
    B ←buffer— x_θ(x_{t_i}^p, t_i)                                          ▷ Evaluation Step
    x_{t_i} ← s_c-step SA-Corrector(x_{t_i}^p, x_{t_{i-1}}, B, ξ) (Eq. (17))   ▷ Correction Step
return x_{t_M}
```

Experiments: SOTA FID results for samplings of Diffusion Model

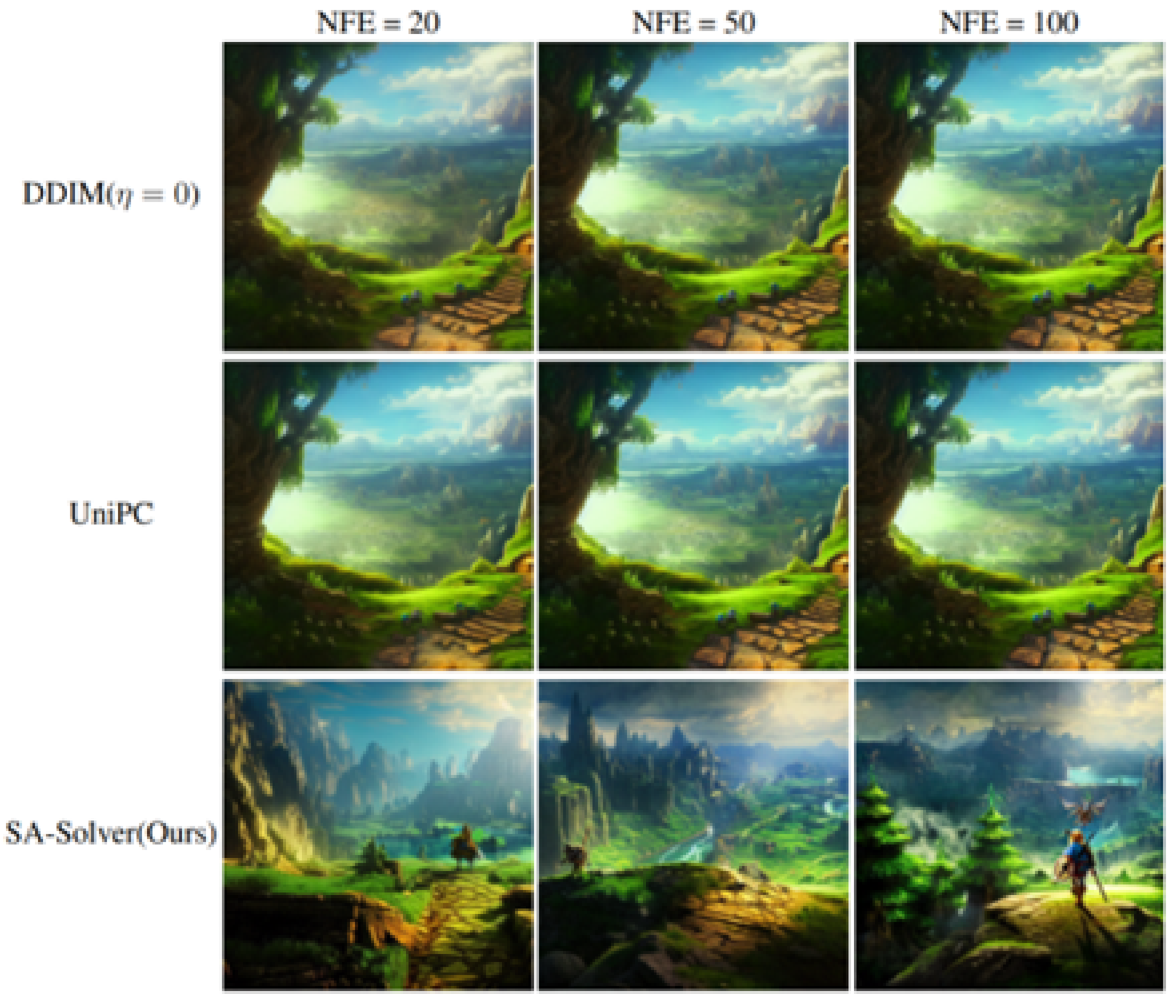


Model	FID (↓)	
DiT ImageNet 256x256	DDPM (NFE=250) 2.27	**SA-Solver (Ours)** (NFE=60) **2.02**
Min-SNR ImageNet 256x256	Heun (NFE=50) 2.06	**SA-Solver (Ours)** (NFE=20) **1.93**
DiT ImageNet 512x512	DDPM (NFE=250) 3.04	**SA-Solver (Ours)** (NFE=60) **2.80**

SA-Solver is a fast SDE-based solver which shares both the fast convergence in small NFEs and optimal results in large NFEs.

Experiments on Text-to-Image Tasks

Model: SD v1.5



The Legend of Zelda landscape

A portrait of curly orange haired mad scientist man

Summary

- We propose Variance-Controlled Diffusion SDE, which extends the reverse process of Diffusion Model.

- We propose a Fast SDE Solver which shares fast convergence and optimal results.

- Code will be released soon on GitHub!