Haoxuan Chen*[1], Yinuo Ren*[1],
Lexing Ying[2,1], Grant M. Rotskoff[3,1]
{haoxuanc,yinuoren,lexing,rotskoff}@stanford.edu

* Equal Contribution
[1] ICME [2] Department of Mathematics [3] Department of Chemistry
Stanford University

November 12, 2024

Accelerating Diffusion Models with Parallel Sampling:

Inference at Sub-Linear Time Complexity

1 Introduction

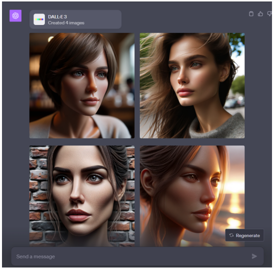
2 Algorithm

3 Main Results

Section 1:
Introduction

Continuous Diffusion Models

Introduction



(a) DALL·E 3

(b) Stable Diffusion

(c) AI4Science

Figure: Diffusion and flow-based generative models have exerted huge impacts on scientific research in many fields.

Introduction

Problem Setting

- **Task:** Sample from data distribution p_0 *accurately* and *efficiently*

Introduction

Problem Setting

- **Task:** Sample from data distribution p_0 *accurately* and *efficiently*
- **Forward SDE:**

$$\mathrm{d}\boldsymbol{x}_s = \boldsymbol{\beta}_s(\boldsymbol{x}_s)\mathrm{d}s + \boldsymbol{\sigma}_s \mathrm{d}\boldsymbol{w}_s, \quad \text{with} \quad \boldsymbol{x}_0 \sim p_0$$

Introduction

Problem Setting

- **Task:** Sample from data distribution p_0 *accurately* and *efficiently*
- **Forward SDE:**

$$\mathrm{d}\boldsymbol{x}_s = \boldsymbol{\beta}_s(\boldsymbol{x}_s)\mathrm{d}s + \boldsymbol{\sigma}_s \mathrm{d}\boldsymbol{w}_s, \quad \text{with} \quad \boldsymbol{x}_0 \sim p_0$$

- **Backward SDE:**

$$\mathrm{d}\overleftarrow{\boldsymbol{x}}_t = \left[-\overleftarrow{\boldsymbol{\beta}}_t(\overleftarrow{\boldsymbol{x}}_t) + \frac{\overleftarrow{\boldsymbol{\sigma}}_t\overleftarrow{\boldsymbol{\sigma}}_t^\top + \overleftarrow{\boldsymbol{v}}_t\overleftarrow{\boldsymbol{v}}_t^\top}{2}\nabla \log \overleftarrow{p}_t(\overleftarrow{\boldsymbol{x}}_t)\right]\mathrm{d}t + \overleftarrow{\boldsymbol{v}}_t \mathrm{d}\boldsymbol{w}_t$$

with $\overleftarrow{p}_0 = p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\overleftarrow{p}_T = p_0$

Introduction

Problem Setting

- **Task:** Sample from data distribution p_0 *accurately* and *efficiently*
- **Forward SDE:**

$$\mathrm{d}\boldsymbol{x}_s = \boldsymbol{\beta}_s(\boldsymbol{x}_s)\mathrm{d}s + \boldsymbol{\sigma}_s \mathrm{d}\boldsymbol{w}_s, \quad \text{with} \quad \boldsymbol{x}_0 \sim p_0$$

- **Backward SDE:**

$$\mathrm{d}\overleftarrow{\boldsymbol{x}}_t = \left[-\overleftarrow{\boldsymbol{\beta}}_t(\overleftarrow{\boldsymbol{x}}_t) + \frac{\overleftarrow{\boldsymbol{\sigma}}_t\overleftarrow{\boldsymbol{\sigma}}_t^\top + \overleftarrow{\boldsymbol{v}}_t\overleftarrow{\boldsymbol{v}}_t^\top}{2}\nabla \log \overleftarrow{p}_t(\overleftarrow{\boldsymbol{x}}_t)\right]\mathrm{d}t + \overleftarrow{\boldsymbol{v}}_t \mathrm{d}\boldsymbol{w}_t$$

  with $\overleftarrow{p}_0 = p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\overleftarrow{p}_T = p_0$
- **Score Function:** $\boldsymbol{s}_t^\theta(\boldsymbol{x}_t) \approx \nabla \log p_t(\boldsymbol{x}_t)$ by optimizing

$$\mathcal{L}(\theta) = \int_0^T \psi_t \mathbb{E}_{\boldsymbol{x}_t \sim p_t}\left[\left\|\nabla \log p_t(\boldsymbol{x}_t) - \boldsymbol{s}_t^\theta(\boldsymbol{x}_t)\right\|^2\right]\mathrm{d}t$$

Introduction

Problem Setting

- **Task:** Sample from data distribution p_0 *accurately* and *efficiently*
- **Forward SDE:**

$$\mathrm{d}\boldsymbol{x}_s = \boldsymbol{\beta}_s(\boldsymbol{x}_s)\mathrm{d}s + \boldsymbol{\sigma}_s \mathrm{d}\boldsymbol{w}_s, \quad \text{with} \quad \boldsymbol{x}_0 \sim p_0$$

- **Backward SDE:**

$$\mathrm{d}\overleftarrow{\boldsymbol{x}}_t = \left[-\overleftarrow{\boldsymbol{\beta}}_t(\overleftarrow{\boldsymbol{x}}_t) + \frac{\overleftarrow{\boldsymbol{\sigma}}_t\overleftarrow{\boldsymbol{\sigma}}_t^\top + \overleftarrow{\boldsymbol{v}}_t\overleftarrow{\boldsymbol{v}}_t^\top}{2}\nabla \log \overleftarrow{p}_t(\overleftarrow{\boldsymbol{x}}_t)\right]\mathrm{d}t + \overleftarrow{\boldsymbol{v}}_t \mathrm{d}\boldsymbol{w}_t$$

  with $\overleftarrow{p}_0 = p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\overleftarrow{p}_T = p_0$
- **Score Function**: $\boldsymbol{s}_t^\theta(\boldsymbol{x}_t) \approx \nabla \log p_t(\boldsymbol{x}_t)$ by optimizing

$$\mathcal{L}(\theta) = \int_0^T \psi_t \mathbb{E}_{\boldsymbol{x}_t \sim p_t}\left[\left\|\nabla \log p_t(\boldsymbol{x}_t) - \boldsymbol{s}_t^\theta(\boldsymbol{x}_t)\right\|^2\right]\mathrm{d}t$$

- **Implementations:** SDE ($\boldsymbol{v}_t = \boldsymbol{\sigma}_t$), Probability Flow ODE (PF-ODE, $\boldsymbol{v}_t \equiv \mathbf{0}$)

Introduction

Error Analysis

Take $\boldsymbol{\beta}_s(\boldsymbol{x}_s) = -\frac{1}{2}\boldsymbol{x}_s$ and $\boldsymbol{\sigma}_s = \boldsymbol{I}$:

- **Forward SDE:** $\mathrm{d}\boldsymbol{x}_s = -\frac{1}{2}\boldsymbol{x}_s \mathrm{d}s + \mathrm{d}\boldsymbol{w}_s$ with $\boldsymbol{x}_0 \sim p_0$

Introduction

Error Analysis

Take $\boldsymbol{\beta}_s(\boldsymbol{x}_s) = -\frac{1}{2}\boldsymbol{x}_s$ and $\boldsymbol{\sigma}_s = \boldsymbol{I}$:

- **Forward SDE:** $\mathrm{d}\boldsymbol{x}_s = -\frac{1}{2}\boldsymbol{x}_s \mathrm{d}s + \mathrm{d}\boldsymbol{w}_s$ with $\boldsymbol{x}_0 \sim p_0$
- **Backward SDE:** $\mathrm{d}\overleftarrow{\boldsymbol{x}}_t = \left[\frac{1}{2}\overleftarrow{\boldsymbol{x}}_t + \frac{1+\upsilon^2}{2}\nabla \log \overleftarrow{p}_t(\overleftarrow{\boldsymbol{x}}_t)\right]\mathrm{d}t + \upsilon \mathrm{d}\boldsymbol{w}_t$, with $\overleftarrow{p}_0 = p_T \approx \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\overleftarrow{p}_T = p_0$

Introduction

Error Analysis

Take $\boldsymbol{\beta}_s(\boldsymbol{x}_s) = -\frac{1}{2}\boldsymbol{x}_s$ and $\boldsymbol{\sigma}_s = \boldsymbol{I}$:

- Forward SDE: $\mathrm{d}\boldsymbol{x}_s = -\frac{1}{2}\boldsymbol{x}_s\mathrm{d}s + \mathrm{d}\boldsymbol{w}_s$ with $\boldsymbol{x}_0 \sim p_0$
- Backward SDE: $\mathrm{d}\overleftarrow{\boldsymbol{x}}_t = \left[\frac{1}{2}\overleftarrow{\boldsymbol{x}}_t + \frac{1+v^2}{2}\nabla \log \overleftarrow{p}_t(\overleftarrow{\boldsymbol{x}}_t)\right]\mathrm{d}t + v\mathrm{d}\boldsymbol{w}_t$, with $\overleftarrow{p}_0 = p_T \approx \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\overleftarrow{p}_T = p_0$

Theorem (Error Analysis of Continuous Diffusion Models [BDBDD23])

Suppose $t_0 = 0 \leq \cdots \leq t_N = T - \delta$ *satisfies* $t_{k+1} - t_k \leq \kappa(T - t_{k+1})$ *and*

$$\sum_{k=0}^{N-1}(s_{k+1} - s_k)\mathbb{E}_{\overleftarrow{\boldsymbol{x}}_{s_k} \sim \overleftarrow{p}_{s_k}}\left[\left\|\nabla \log \overleftarrow{p}_{s_k}(\overleftarrow{\boldsymbol{x}}_{s_k}) - \widehat{\overleftarrow{\boldsymbol{s}}}^{\theta}_{s_k}(\boldsymbol{x}_{s_k})\right\|^2\right] \leq \epsilon.$$

Then with

$$T = \mathcal{O}(\log(d\epsilon^{-1})),\ \kappa = \mathcal{O}(d^{-1}\epsilon \log^{-1}(d\epsilon^{-1})),\ N = \mathcal{O}(d\epsilon^{-1}\log^2(d\epsilon^{-1})),$$

we have

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{t_N}) \lesssim de^{-T} + \epsilon + d\kappa T \lesssim \epsilon.$$

Introduction

Error Analysis

Theorem (Error Analysis of Continuous Diffusion Models [BDBDD23])

With

$$T = \mathcal{O}(\log(d\epsilon^{-1})), \; \kappa = \mathcal{O}(d^{-1}\epsilon \log^{-1}(d\epsilon^{-1})), \; N = \mathcal{O}(d\epsilon^{-1}\log^2(d\epsilon^{-1})),$$

we have

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{t_N}) \lesssim de^{-T} + \epsilon + d\kappa T \lesssim \epsilon.$$

Introduction

Error Analysis

Theorem (Error Analysis of Continuous Diffusion Models [BDBDD23])

With

$$T = \mathcal{O}(\log(d\epsilon^{-1})),\ \kappa = \mathcal{O}(d^{-1}\epsilon\log^{-1}(d\epsilon^{-1})),\ N = \mathcal{O}(d\epsilon^{-1}\log^2(d\epsilon^{-1})),$$

we have

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{t_N}) \lesssim de^{-T} + \epsilon + d\kappa T \lesssim \epsilon.$$

- **Truncation Error:** Error caused by approximating p_T by p_∞, of the order $\mathcal{O}(d\exp(-T))$;

Introduction

Error Analysis

Theorem (Error Analysis of Continuous Diffusion Models [BDBDD23])

With

$$T = \mathcal{O}(\log(d\epsilon^{-1})),\ \kappa = \mathcal{O}(d^{-1}\epsilon \log^{-1}(d\epsilon^{-1})),\ N = \mathcal{O}(d\epsilon^{-1}\log^2(d\epsilon^{-1})),$$

we have

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{t_N}) \lesssim de^{-T} + \epsilon + d\kappa T \lesssim \epsilon.$$

- **Truncation Error:** Error caused by approximating p_T by p_∞, of the order $\mathcal{O}(d\exp(-T))$;
- **Approximation Error:** Error caused by approximating $\nabla \log p_t(\boldsymbol{x}_t)$ by NN $\widehat{\boldsymbol{s}}_t^\theta(\boldsymbol{x}_t)$, assumed to be of $\mathcal{O}(\epsilon)$;

Introduction

Error Analysis

Theorem (Error Analysis of Continuous Diffusion Models [BDBDD23])

With

$$T = \mathcal{O}(\log(d\epsilon^{-1})),\ \kappa = \mathcal{O}(d^{-1}\epsilon\log^{-1}(d\epsilon^{-1})),\ N = \mathcal{O}(d\epsilon^{-1}\log^2(d\epsilon^{-1})),$$

we have

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{t_N}) \lesssim de^{-T} + \epsilon + d\kappa T \lesssim \epsilon.$$

- **Truncation Error:** Error caused by approximating p_T by p_∞, of the order $\mathcal{O}(d\exp(-T))$;
- **Approximation Error:** Error caused by approximating $\nabla \log p_t(\boldsymbol{x}_t)$ by NN $\widehat{\boldsymbol{s}}_t^\theta(\boldsymbol{x}_t)$, assumed to be of $\mathcal{O}(\epsilon)$;
- **Discretization Error:** Error caused by numerically solving the backward SDE, *e.g.* exponential integrator [ZC22].

Introduction

Inference Cost

Inference Cost

- The evaluation of the score function s_t^θ is expensive
- The inference process of continuous diffusion models requires $\widetilde{\mathcal{O}}(d)$ times of score function evaluations

Possible Solutions

- DDIM [SME20]
- Higher-order schemes [DVK22, KAAL22, LHE$^+$24]
- Operator learning [ZNV$^+$23]
- Knowledge distillation [LL21, MRG$^+$23]
- Consistency model [SDCS23, SD23, LS24]
- Parallel sampling [SBE$^+$24, TTL$^+$24]

Section 2:
Algorithm

Algorithm

Parallel Sampling

Picard Iteration

For $k \in [0 : K-1]$,

Algorithm

Parallel Sampling

Picard Iteration

For $k \in [0 : K-1]$,

- Solve ODE $\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}_t(\boldsymbol{x}_t)\mathrm{d}t$ in parallel

$$\boldsymbol{x}_t^{(0)} \equiv \boldsymbol{x}_0, \text{ and } \boldsymbol{x}_t^{(k+1)} := \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}_s(\boldsymbol{x}_s^{(k)})\mathrm{d}s$$

Picard Iteration

For $k \in [0 : K-1]$,

- Solve ODE $\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}_t(\boldsymbol{x}_t)\mathrm{d}t$ in parallel

$$\boldsymbol{x}_t^{(0)} \equiv \boldsymbol{x}_0, \text{ and } \boldsymbol{x}_t^{(k+1)} := \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}_s(\boldsymbol{x}_s^{(k)})\mathrm{d}s$$

- Simulate Langevin dynamics $\mathrm{d}\boldsymbol{x}_t = -\nabla V(\boldsymbol{x}_t)\mathrm{d}t + \mathrm{d}\boldsymbol{w}_t$ in parallel [ACV24]

$$\boldsymbol{x}_t^{(0)} \equiv \boldsymbol{x}_0, \text{ and } \boldsymbol{x}_t^{(k+1)} := \boldsymbol{x}_0 - \int_0^t \nabla V(\boldsymbol{x}_t^{(k)})\mathrm{d}s + \boldsymbol{w}_t$$

Algorithm

Parallel Sampling

Picard Iteration

For $k \in [0 : K-1]$,

- Solve ODE $\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}_t(\boldsymbol{x}_t)\mathrm{d}t$ in parallel

$$\boldsymbol{x}_t^{(0)} \equiv \boldsymbol{x}_0, \text{ and } \boldsymbol{x}_t^{(k+1)} := \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}_s(\boldsymbol{x}_s^{(k)})\mathrm{d}s$$

- Simulate Langevin dynamics $\mathrm{d}\boldsymbol{x}_t = -\nabla V(\boldsymbol{x}_t)\mathrm{d}t + \mathrm{d}\boldsymbol{w}_t$ in parallel [ACV24]

$$\boldsymbol{x}_t^{(0)} \equiv \boldsymbol{x}_0, \text{ and } \boldsymbol{x}_t^{(k+1)} := \boldsymbol{x}_0 - \int_0^t \nabla V(\boldsymbol{x}_t^{(k)})\mathrm{d}s + \boldsymbol{w}_t$$

- Sample from diffusion models in parallel (This work)

$$\mathrm{d}\widehat{\boldsymbol{y}}_{t_n,\tau}^{(k+1)} = \left[\frac{1}{2}\widehat{\boldsymbol{y}}_{t_n,\tau}^{(k+1)} + \boldsymbol{s}_{t_n+g_n(\tau)}^{\theta}\left(\widehat{\boldsymbol{y}}_{t_n,g_n(\tau)}^{(k)}\right)\right]\mathrm{d}\tau + \mathrm{d}\boldsymbol{w}_{t_n+\tau}$$

Algorithm

Parallel Sampling

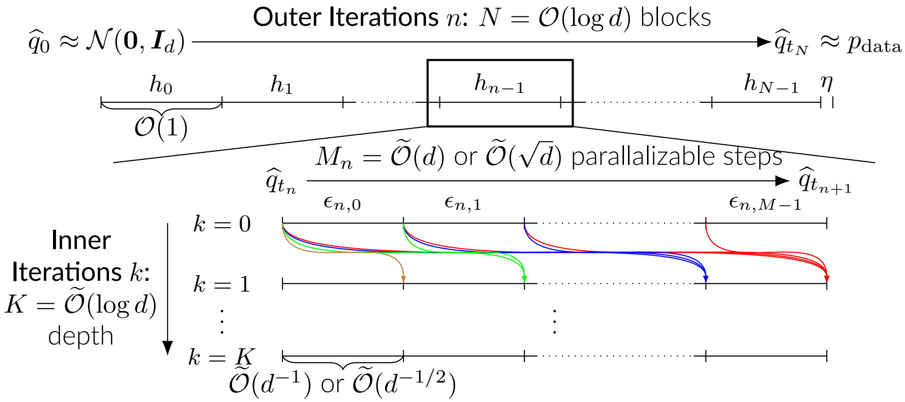


Figure: Illustration of PIADM-SDE/ODE.

Section 3:
Main Results

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$
- **Bounded learned score:** The learned score $\boldsymbol{s}_t^\theta$ has bounded C^1 norm with Lipschitz const $L_{\boldsymbol{s}}$.

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$
- **Bounded learned score:** The learned score $\boldsymbol{s}_t^\theta$ has bounded C^1 norm with Lipschitz const $L_{\boldsymbol{s}}$.
- **δ-accurate score estimation:**

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$
- **Bounded learned score:** The learned score $\boldsymbol{s}_t^\theta$ has bounded C^1 norm with Lipschitz const $L_{\boldsymbol{s}}$.
- δ**-accurate score estimation:**

SDE The learned score $\boldsymbol{s}_t^\theta$ is $L^2([0, t_N])$ δ-accurate:

$$\mathbb{E}_{\overleftarrow{p}}\left[\sum_{n=0}^{N-1}\sum_{m=0}^{M_n-1}\epsilon_{n,m}\left\|\boldsymbol{s}^\theta_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right) - \nabla\log\overleftarrow{p}_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right)\right\|^2\right] \leq \delta_2^2.$$

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$
- **Bounded learned score:** The learned score $\boldsymbol{s}_t^\theta$ has bounded C^1 norm with Lipschitz const $L_{\boldsymbol{s}}$.
- **δ-accurate score estimation:**

SDE The learned score $\boldsymbol{s}_t^\theta$ is $L^2([0, t_N])$ δ-accurate:

$$\mathbb{E}_{\overleftarrow{p}}\left[\sum_{n=0}^{N-1}\sum_{m=0}^{M_n-1}\epsilon_{n,m}\left\|\boldsymbol{s}^\theta_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right) - \nabla\log\overleftarrow{p}_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right)\right\|^2\right] \leq \delta_2^2.$$

PF-ODE The learned score $\boldsymbol{s}_t^\theta$ is $L^\infty([0, t_N])$ δ-accurate:

$$\mathbb{E}_{\overleftarrow{p}_{t_n+\tau_{n,m}}}\left[\left\|\boldsymbol{s}^\theta_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right) - \nabla\log\overleftarrow{p}_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right)\right\|^2\right] \leq \delta_\infty^2.$$

Main Results

Assumptions

- **Regularity of data distribution:** p_0 has finite second moment and is normalized, i.e., $\mathrm{cov}_{p_0}(\boldsymbol{x}_0) = \boldsymbol{I}_d$
- **Bounded learned score:** The learned score $\boldsymbol{s}_t^\theta$ has bounded C^1 norm with Lipschitz const $L_{\boldsymbol{s}}$.
- **δ-accurate score estimation:**

SDE The learned score $\boldsymbol{s}_t^\theta$ is $L^2([0, t_N])$ δ-accurate:

$$\mathbb{E}_{\overleftarrow{p}}\left[\sum_{n=0}^{N-1}\sum_{m=0}^{M_n-1} \epsilon_{n,m} \left\|\boldsymbol{s}^\theta_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right) - \nabla \log \overleftarrow{p}_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right)\right\|^2\right] \leq \delta_2^2.$$

PF-ODE The learned score $\boldsymbol{s}_t^\theta$ is $L^\infty([0, t_N])$ δ-accurate:

$$\mathbb{E}_{\overleftarrow{p}_{t_n+\tau_{n,m}}}\left[\left\|\boldsymbol{s}^\theta_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right) - \nabla \log \overleftarrow{p}_{t_n+\tau_{n,m}}\left(\overleftarrow{\boldsymbol{x}}_{t_n+\tau_{n,m}}\right)\right\|^2\right] \leq \delta_\infty^2.$$

- **Bounded true score (PF-ODE):** The true score $\nabla \log p_t$ has bounded C^1 norm with Lipschitz const L_p.

Main Results

SDE Implementation

Theorem (Parallel Acceleration for SDE Implementation)

Under assumptions aforementioned, given

$$T = \mathcal{O}(\log(d\delta^{-2})), \quad h = \Theta(1), \quad N = \mathcal{O}\left(\log(d\delta^{-2})\right),$$

$$\epsilon = \Theta\left(d^{-1}\delta^2 \log^{-1}(d\delta^{-2})\right), \quad M = \mathcal{O}\left(d\delta^{-2}\log(d\delta^{-2})\right), \quad K = \widetilde{\mathcal{O}}(\log(d\delta^{-2})),$$

we have the following error bound

$$D_{\mathrm{KL}}(p_\eta \| \widehat{q}_{t_N}) \lesssim de^{-T} + d\epsilon T + \delta_2^2 + dTe^{-K} \lesssim \delta^2,$$

with a total of

- $KN = \widetilde{\mathcal{O}}\left(\log^2(d\delta^{-2})\right)$ *approximate time complexity*
- $dM = \widetilde{\mathcal{O}}\left(d^2\delta^{-2}\right)$ *space complexity*

for parallalizable $L^2([0, t_N])$ δ-accurate score function evaluations.

Main Results

PF-ODE Implementation

PF-ODE with predictor-corrector [CCL+24] further improves space complexity:

Theorem (Parallel Acceleration for PF-ODE Implementation)

Under assumptions aforementioned, given proper parameter selections, we have

$$\mathrm{TV}(p_\eta, \widehat{q}_{t_N})^2 \lesssim de^{-T} + d\epsilon^2 T^2 + (T^2 + N^2)\delta_\infty^2 + dN^2 e^{-K} \lesssim \delta^2,$$

with a total of

- $(K + K^\dagger N^\dagger)N = \widetilde{\mathcal{O}}\left(\log^2(d\delta^{-2})\right)$ *approximate time complexity*
- $d(M \vee M^\dagger) = \widetilde{\Theta}\left(d^{3/2}\delta^{-1}\right)$ *space complexity*

for parallalizable $L^\infty([0, t_N])$ δ-accurate score function evaluations.

Remark

$$\mathbb{E}\left[f(x_t) - f(x_0)\right]^2 \lesssim \mathbb{E}\left[\int f'(x_t) b_t + f''(x_t)\sigma \mathrm{d}t\right]^2 + \mathbb{E}\left[\int f'(x_t)\sqrt{2\sigma}\mathrm{d}w_t\right]^2$$
$$\sim O(t^2) + \sigma O(t),$$

Main Results

Proof Sketch

Theorem (Generalized Girsanov's Theorem)

Let $\boldsymbol{\alpha}(t,\omega) \in \mathcal{V}^m$, $\boldsymbol{\Sigma}(t,\omega) \in \mathcal{V}^{m\times n}$, *and* $(\boldsymbol{w}_t(\omega))_{t\geq 0}$ *be a Wiener process on* $(\Omega, \mathcal{F}, q)$. *For* $t \in [0,T]$, *suppose* $\boldsymbol{z}_t(\omega)$ *satisfies*

$$\mathrm{d}\boldsymbol{z}_t(\omega) = \boldsymbol{\alpha}(t,\omega)\mathrm{d}t + \boldsymbol{\Sigma}(t,\omega)\mathrm{d}\boldsymbol{w}_t(\omega),$$

where $\boldsymbol{\Sigma}(t,\omega)\boldsymbol{\delta}(t,\omega) = \boldsymbol{\alpha}(t,\omega) - \boldsymbol{\beta}(t,\omega)$, *then there exists* p *on* $(\Omega, \mathcal{F})$ *s.t.*

1. $p \ll q$ *with the Radon-Nikodym derivative* $\dfrac{\mathrm{d}p}{\mathrm{d}q}(\omega) = M_T(\omega)$;
2. $\widetilde{\boldsymbol{w}}_t(\omega) = \boldsymbol{w}_t(\omega) + \int_0^t \boldsymbol{\delta}(s,\omega)\mathrm{d}s$ *is a Wiener process on* $(\Omega, \mathcal{F}, p)$;
3. *Any continuous path generated by the process* $\boldsymbol{z}_t$ *satisfies the following SDE under* p:

$$\mathrm{d}\widetilde{\boldsymbol{z}}_t(\omega) = \boldsymbol{\beta}(t,\omega)\mathrm{d}t + \boldsymbol{\Sigma}(t,\omega)\mathrm{d}\widetilde{\boldsymbol{w}}_t(\omega).$$

Main Results

Proof Sketch

In n-th block, let $q|_{\mathcal{F}_{t_n}}$ be the measure shared by $\boldsymbol{w}_t(\omega)$ in the Picard iteration

1. Define $\mathrm{d}\widetilde{\boldsymbol{w}}_{t_n+\tau}(\omega) = \mathrm{d}\boldsymbol{w}_{t_n+\tau}(\omega) + \boldsymbol{\delta}_{t_n}(\tau,\omega)d\tau$, where

$$\boldsymbol{\delta}_{t_n}(\tau,\omega) := \boldsymbol{s}^{\theta}_{t_n+g_n(\tau)}(\widehat{\boldsymbol{y}}^{(K-1)}_{t_n,g_n(\tau)}(\omega)) - \nabla \log \overleftarrow{p}_{t_n+\tau}(\widehat{\boldsymbol{y}}^{(K)}_{t_n+\tau}(\omega));$$

Main Results

Proof Sketch

In n-th block, let $q|_{\mathcal{F}_{t_n}}$ be the measure shared by $\boldsymbol{w}_t(\omega)$ in the Picard iteration

1. Define $\mathrm{d}\widetilde{\boldsymbol{w}}_{t_n+\tau}(\omega) = \mathrm{d}\boldsymbol{w}_{t_n+\tau}(\omega) + \boldsymbol{\delta}_{t_n}(\tau,\omega)d\tau$, where

$$\boldsymbol{\delta}_{t_n}(\tau,\omega) := \boldsymbol{s}^{\theta}_{t_n+g_n(\tau)}(\widehat{\boldsymbol{y}}^{(K-1)}_{t_n,g_n(\tau)}(\omega)) - \nabla \log \overleftarrow{p}_{t_n+\tau}(\widehat{\boldsymbol{y}}^{(K)}_{t_n+\tau}(\omega));$$

2. Invoke Girsanov's theorem

$$\log \frac{\mathrm{d}\overleftarrow{p}|_{\mathcal{F}_{t_n}}}{\mathrm{d}q|_{\mathcal{F}_{t_n}}}(\omega) = -\int_0^{h_n} \boldsymbol{\delta}_{t_n}(\tau,\omega)^\top \mathrm{d}\boldsymbol{w}_{t_n+\tau}(\omega) - \frac{1}{2}\int_0^{h_n} \|\boldsymbol{\delta}_{t_n}(\tau,\omega)\|^2 \mathrm{d}\tau;$$

Main Results

Proof Sketch

In n-th block, let $q|_{\mathcal{F}_{t_n}}$ be the measure shared by $\boldsymbol{w}_t(\omega)$ in the Picard iteration

1. Define $\mathrm{d}\widetilde{\boldsymbol{w}}_{t_n+\tau}(\omega) = \mathrm{d}\boldsymbol{w}_{t_n+\tau}(\omega) + \boldsymbol{\delta}_{t_n}(\tau,\omega)d\tau$, where

$$\boldsymbol{\delta}_{t_n}(\tau,\omega) := \boldsymbol{s}^{\theta}_{t_n+g_n(\tau)}(\widehat{\boldsymbol{y}}^{(K-1)}_{t_n,g_n(\tau)}(\omega)) - \nabla \log \overleftarrow{p}_{t_n+\tau}(\widehat{\boldsymbol{y}}^{(K)}_{t_n+\tau}(\omega));$$

2. Invoke Girsanov's theorem

$$\log \frac{\mathrm{d}\overleftarrow{p}|_{\mathcal{F}_{t_n}}}{\mathrm{d}q|_{\mathcal{F}_{t_n}}}(\omega) = -\int_0^{h_n} \boldsymbol{\delta}_{t_n}(\tau,\omega)^\top \mathrm{d}\boldsymbol{w}_{t_n+\tau}(\omega) - \frac{1}{2}\int_0^{h_n} \|\boldsymbol{\delta}_{t_n}(\tau,\omega)\|^2 \mathrm{d}\tau;$$

3. Conclude that $(\widetilde{\boldsymbol{w}}_{t_n+\tau})_{\tau\geq 0}$ is a Wiener process under $\overleftarrow{p}|_{\mathcal{F}_{t_n}}$ and thus:

$$\mathrm{d}\widehat{\boldsymbol{y}}^{(K)}_{t_n,\tau}(\omega) = \left[\frac{1}{2}\widehat{\boldsymbol{y}}^{(K)}_{t_n,\tau}(\omega) + \nabla \log \overleftarrow{p}_{t_n+\tau}\left(\widehat{\boldsymbol{y}}^{(K)}_{t_n,\tau}(\omega)\right)\right] \mathrm{d}\tau + \mathrm{d}\widetilde{\boldsymbol{w}}_{t_n+\tau}(\omega),$$

i.e. the true backward SDE with the true score function for $\tau \in [t_n, t_{n+1}]$.

Conclusion

Empirical Results

- Picard iteration with adaptive window size [SBE+24]
- Triangular Anderson acceleration [TTL+24]

Takeaways

- Parallelized inference algorithm for both SDE and PF-ODE implementations

Conclusion

Empirical Results

- Picard iteration with adaptive window size [SBE+24]
- Triangular Anderson acceleration [TTL+24]

Takeaways

- Parallelized inference algorithm for both SDE and PF-ODE implementations
- Convergence analysis that achieves the first poly-logarithmic error bound for diffusion models with generalized Girsanov's theorem

Conclusion

Empirical Results

- Picard iteration with adaptive window size [SBE+24]
- Triangular Anderson acceleration [TTL+24]

Takeaways

- Parallelized inference algorithm for both SDE and PF-ODE implementations
- Convergence analysis that achieves the first poly-logarithmic error bound for diffusion models with generalized Girsanov's theorem
- Improved space complexity for PF-ODE implementation with predictor-corrector from $\widetilde{\mathcal{O}}(d^2)$ to $\widetilde{\Theta}(d^{3/2})$

Thank you for your attention!

References I

- Nima Anari, Sinho Chewi, and Thuy-Duong Vuong, *Fast parallel sampling under isoperimetry*, arXiv preprint arXiv:2401.09016 (2024).
- Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis, *Linear convergence bounds for diffusion models via stochastic localization*, arXiv preprint arXiv:2308.03686 (2023).
- Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim, *The probability flow ode is provably fast*, Advances in Neural Information Processing Systems **36** (2024).
- Tim Dockhorn, Arash Vahdat, and Karsten Kreis, *Genie: Higher-order denoising diffusion solvers*, Advances in Neural Information Processing Systems **35** (2022), 30150–30166.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine, *Elucidating the design space of diffusion-based generative models*, Advances in Neural Information Processing Systems **35** (2022), 26565–26577.

References II

Gen Li, Yu Huang, Timofey Efimov, Yuting Wei, Yuejie Chi, and Yuxin Chen, *Accelerating convergence of score-based diffusion models, provably*, arXiv preprint arXiv:2403.03852 (2024).

Eric Luhman and Troy Luhman, *Knowledge distillation in iterative generative models for improved sampling speed*, arXiv preprint arXiv:2101.02388 (2021).

Cheng Lu and Yang Song, *Simplifying, stabilizing and scaling continuous-time consistency models*, arXiv preprint arXiv:2410.11081 (2024).

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans, *On distillation of guided diffusion models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14297–14306.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari, *Parallel sampling of diffusion models*, Advances in Neural Information Processing Systems **36** (2024).

References III

- Yang Song and Prafulla Dhariwal, *Improved techniques for training consistency models*, arXiv preprint arXiv:2310.14189 (2023).
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever, *Consistency models*, arXiv preprint arXiv:2303.01469 (2023).
- Jiaming Song, Chenlin Meng, and Stefano Ermon, *Denoising diffusion implicit models*, arXiv preprint arXiv:2010.02502 (2020).
- Zhiwei Tang, Jiasheng Tang, Hao Luo, Fan Wang, and Tsung-Hui Chang, *Accelerating parallel sampling of diffusion models*, arXiv preprint arXiv:2402.09970 (2024).
- Qinsheng Zhang and Yongxin Chen, *Fast sampling of diffusion models with exponential integrator*, arXiv preprint arXiv:2204.13902 (2022).
- Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar, *Fast sampling of diffusion models via operator learning*, International Conference on Machine Learning, PMLR, 2023, pp. 42390–42402.