DDPM score matching and distribution learning

Sinho Chewi
Yale University

April 28, 2025
ICLR Workshop on Deep Generative Models in Machine Learning: Theory, Principles, and Efficacy (DeLTa)

Alkis Kalavasis
Yale

Anay Mehrotra
Yale

Omar Montasser
Yale

Diffusion models

- Data distribution: $p_{\text{data}} = p_0$.
- Forward process: $\mathrm{d}X_t = -X_t \,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \quad X_t \sim p_t$.
- Reverse process: $\mathrm{d}X_t^{\leftarrow} = \{X_t^{\leftarrow} + 2\,\nabla \log p_{T-t}(X_t^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$.

Diffusion models

- Data distribution: $p_{\text{data}} = p_0$.
- Forward process: $\mathrm{d}X_t = -X_t \,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \quad X_t \sim p_t$.
- Reverse process: $\mathrm{d}X_t^{\leftarrow} = \{X_t^{\leftarrow} + 2\,\nabla \log p_{T-t}(X_t^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$.

Training

- Learn score functions via the score matching objective:
  $\nabla \log p_t = \arg\min_{s_t : \mathbb{R}^d \to \mathbb{R}^d} \mathbb{E}_{x_0 \sim p_0} \mathrm{SM}_t(s_t, x_0)$, where

$$\begin{aligned}
&\mathrm{SM}_t(s_t, x_0) \\
&\quad := \mathbb{E}_{x_t \sim q^{\mathrm{OU}}_{t|0}(\cdot|x_0)} \left[\|s_t(x_t)\|^2 + 2\,\langle s_t(x_t), \nabla \log q^{\mathrm{OU}}_{t|0}(x_t \mid x_0)\rangle \right].
\end{aligned}$$

Diffusion models

- Data distribution: $p_{\text{data}} = p_0$.
- Forward process: $\mathrm{d}X_t = -X_t\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \quad X_t \sim p_t$.
- Reverse process: $\mathrm{d}X_t^{\leftarrow} = \{X_t^{\leftarrow} + 2\,\nabla \log p_{T-t}(X_t^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$.

Training

- Learn score functions via the score matching objective:
  $\nabla \log p_t = \arg\min_{s_t:\mathbb{R}^d \to \mathbb{R}^d} \mathbb{E}_{x_0 \sim p_0} \mathrm{SM}_t(s_t, x_0)$, where

$$\mathrm{SM}_t(s_t, x_0) := \mathbb{E}_{x_t \sim q_{t|0}^{\mathrm{OU}}(\cdot|x_0)} \left[\|s_t(x_t)\|^2 + 2\left\langle s_t(x_t), \nabla \log q_{t|0}^{\mathrm{OU}}(x_t \mid x_0)\right\rangle\right].$$

- Empirical version: $\widehat{s}_t = \arg\min_{s_t \in \mathcal{S}_t} n^{-1} \sum_{i=1}^{n} \mathrm{SM}_t(s_t, x_0^{(i)})$.

Sampling from diffusion models

Generation

- Once we have estimated scores $\{\widehat{s}_t\}_{t\in[0,T]}$, we discretize the reverse process: $\mathrm{d}\widehat{X}_t^{\leftarrow} = \{\widehat{X}_t^{\leftarrow} + 2\,\widehat{s}_{t_-}(\widehat{X}_{t_-}^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$, where $\widehat{X}_0^{\leftarrow} \sim \widehat{p}_T = \mathrm{N}(0, I_d)$ and $\widehat{X}_T^{\leftarrow} \sim \widehat{p}_0$.

Sampling from diffusion models

Generation

- Once we have estimated scores $\{\widehat{s}_t\}_{t\in[0,T]}$, we discretize the reverse process: $\mathrm{d}\widehat{X}_t^{\leftarrow} = \{\widehat{X}_t^{\leftarrow} + 2\,\widehat{s}_{t_-}(\widehat{X}_{t_-}^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$, where $\widehat{X}_0^{\leftarrow} \sim \widehat{p}_T = \mathrm{N}(0, I_d)$ and $\widehat{X}_T^{\leftarrow} \sim \widehat{p}_0$.

Theorem: Denoising diffusion models (DDPM) achieve $\mathrm{TV}(\widehat{p}_0, p_0) \leq \varepsilon_*$ in $\mathrm{poly}(d, L, 1/\varepsilon_*)$ steps, where:

- $\|\nabla \log p_t\|_{\mathrm{Lip}} \leq L$ and $\mathbb{E}_{x_0\sim p_0}[\|x_0\|^2] \leq \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_{t_-} - \nabla \log p_{t_-}\|^2_{L^2(p_t)}\,\mathrm{d}t \leq \varepsilon_*^2$.

Sampling from diffusion models

Generation

- Once we have estimated scores $\{\widehat{s}_t\}_{t\in[0,T]}$, we discretize the reverse process: $\mathrm{d}\widehat{X}_t^{\leftarrow} = \{\widehat{X}_t^{\leftarrow} + 2\,\widehat{s}_{t_-}(\widehat{X}_{t_-}^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$, where $\widehat{X}_0^{\leftarrow} \sim \widehat{p}_T = \mathsf{N}(0, I_d)$ and $\widehat{X}_T^{\leftarrow} \sim \widehat{p}_0$.

Theorem: Denoising diffusion models (DDPM) achieve $\mathsf{TV}(\widehat{p}_0, p_0) \le \varepsilon_*$ in $\mathsf{poly}(d, L, 1/\varepsilon_*)$ steps, where:

- $\|\nabla \log p_t\|_{\mathsf{Lip}} \le L$ and $\mathbb{E}_{x_0\sim p_0}[\|x_0\|^2] \le \mathsf{poly}(d)$.
- $\int_0^T \|\widehat{s}_{t_-} - \nabla \log p_{t_-}\|_{L^2(p_t)}^2\,\mathrm{d}t \le \varepsilon_*^2$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathsf{poly}(d, \log(1/\tau), 1/\varepsilon_*)$.

Sampling from diffusion models

Theorem: Denoising diffusion models (DDPM) achieve $\mathrm{TV}(\widehat{p}_0, p_0) \le \varepsilon_*$ in $\mathrm{poly}(d, L, 1/\varepsilon_*)$ steps, where:

- $\|\nabla \log p_t\|_{\mathrm{Lip}} \le L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \le \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_{t_-} - \nabla \log p_{t_-}\|^2_{L^2(p_t)}\, \mathrm{d}t \le \varepsilon_*^2$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathrm{poly}(d, \log(1/\tau), 1/\varepsilon_*)$.

- Sitan Chen, **S.C.**, Jerry Li, Yuanzhi Li, Adil Salim, Anru Zhang, *Sampling is as easy as learning the score.* ICLR 2023.
- [Concurrent] Holden Lee, Jianfeng Lu, Yixin Tan, *Convergence of score-based generative modeling for general data distributions.* ALT 2023.
- Many, many follow-up works…

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0. ✓ (see previous theorem)

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0. ✓ (see previous theorem)
- Given samples from p_0, generate new samples from p_0.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0. ✓ (see previous theorem)
- Given samples from p_0, generate new samples from p_0. **NO**, this is information-theoretically impossible.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0. ✓ (see previous theorem)
- Given samples from p_0, generate new samples from p_0. **NO**, this is information-theoretically impossible.
- Given samples from p_0, "learn a sampler" for p_0.

Which of these tasks make sense?

- Given access to evaluations of the likelihood of p_0, output a sample from p_0. ✓ (standard MCMC setup)
- Given samples from p_0, learn the score functions along the diffusion. ✓ (statistical theory for score matching)
- Given score functions along the diffusion, output a new sample from p_0. ✓ (see previous theorem)
- Given samples from p_0, generate new samples from p_0. **NO**, this is information-theoretically impossible.
- Given samples from p_0, "learn a sampler" for p_0. ✓ (yes, we shall see that this makes sense)

Distribution learning

What does it mean to "learn" a family $\mathscr{P}$ of distributions, given samples $X^{(1)}, \ldots, X^{(n)} \overset{\text{i.i.d.}}{\sim} p \in \mathscr{P}$?

Distribution learning

What does it mean to "learn" a family $\mathscr{P}$ of distributions, given samples $X^{(1)}, \ldots, X^{(n)} \overset{\text{i.i.d.}}{\sim} p \in \mathscr{P}$? Depends on the *representation*.

Distribution learning

What does it mean to "learn" a family $\mathscr{P}$ of distributions, given samples $X^{(1)}, \ldots, X^{(n)} \overset{\text{i.i.d.}}{\sim} p \in \mathscr{P}$? Depends on the *representation*.

- (Parameter Recovery) If $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, the goal is to output an estimate $\widehat{\theta}$ of the parameter.

Distribution learning

What does it mean to "learn" a family $\mathscr{P}$ of distributions, given samples $X^{(1)}, \dots, X^{(n)} \overset{\text{i.i.d.}}{\sim} p \in \mathscr{P}$? Depends on the *representation*.

- (Parameter Recovery) If $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, the goal is to output an estimate $\widehat{\theta}$ of the parameter.
- (Density Estimation) The goal is to output an *evaluator*, i.e., a function $\widehat{p} : \mathbb{R}^d \to \mathbb{R}$ such that $\widehat{p}(x)$ is a good estimator of the density $p(x)$ at x.

Yale

Distribution learning

What does it mean to "learn" a family $\mathscr{P}$ of distributions, given samples $X^{(1)}, \ldots, X^{(n)} \overset{\text{i.i.d.}}{\sim} p \in \mathscr{P}$? Depends on the *representation*.

- (Parameter Recovery) If $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, the goal is to output an estimate $\widehat{\theta}$ of the parameter.
- (Density Estimation) The goal is to output an *evaluator*, i.e., a function $\widehat{p} : \mathbb{R}^d \to \mathbb{R}$ such that $\widehat{p}(x)$ is a good estimator of the density $p(x)$ at x.
- (Learning a Sampler) The goal is to output a *generator*, i.e., a function $\widehat{\mathscr{G}} : [0, 1] \to \mathbb{R}^d$ which takes a random seed $U \sim \text{uniform}([0, 1])$, such that $\text{law}(\widehat{\mathscr{G}}(U) \mid X) \approx p$.

The distinction is computational

Remark: Density estimation and learning a sampler are equivalent from the lens of information theory, but not from the lens of computational complexity.

The distinction is computational

Remark: Density estimation and learning a sampler are equivalent from the lens of information theory, but not from the lens of computational complexity.

- E.g., minimax lower bounds in statistics apply to both models.

The distinction is computational

Remark: Density estimation and learning a sampler are equivalent from the lens of information theory, but not from the lens of computational complexity.

- E.g., minimax lower bounds in statistics apply to both models.
- This is not true for computational lower bounds.

Reinterpreting the DDPM result as distribution learning

Theorem (informal): Let $\mathscr{P}$ be nearly any ("realistic") family of distributions. Then, the sample complexity of learning a sampler for $\mathscr{P}$ is at most the sample complexity of learning the score functions along DDPM for $\mathscr{P}$.

Moreover, the computational complexity is at most a polynomial factor worse.

Reinterpreting the DDPM result as distribution learning

Theorem (informal): Let $\mathscr{P}$ be nearly any ("realistic") family of distributions. Then, the sample complexity of learning a sampler for $\mathscr{P}$ is at most the sample complexity of learning the score functions along DDPM for $\mathscr{P}$.

Moreover, the computational complexity is at most a polynomial factor worse.

Learning a sampler is as easy as learning the scores.

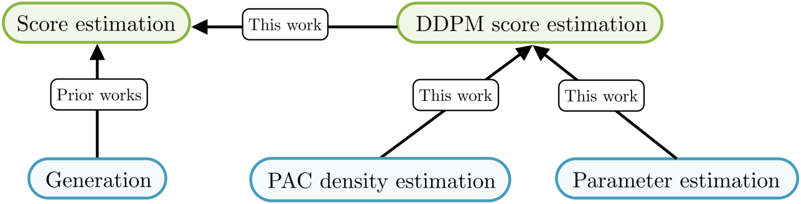
Examples of learning a sampler via DDPM

This is a growing literature!

- [Oko, Akiyama, Suzuki '23; Dou, Kotekal, Xu, Zhou '24] Score matching along DDPM yields minimax optimal samplers for Besov and Hölder classes of densities.
- [Chen, Kontonis, Shah '24; Gatmiry, Kelner, Lee '24] Score matching along DDPM yields new algorithmic results for learning mixtures of Gaussians (in the sense of learning a sampler).
- ...

Our new results

[**S.C.**, Alkis Kalavasis, Anay Mehrotra, Omar Montasser, *DDPM score matching and distribution learning.* '25]



Yale

Outline

- A key identity for the likelihood
- Implications for parameter estimation
- Implications for density estimation
- Implications for computational lower bounds

Likelihood identity

Recall:

- $\nabla \log p_t = \arg\min_{s_t : \mathbb{R}^d \to \mathbb{R}^d} \mathbb{E}_{x_0 \sim p_0} \mathrm{SM}_t(s_t, x_0)$.
- $\widehat{s}_t = \arg\min_{s_t \in \mathcal{S}_t} n^{-1} \sum_{i=1}^{n} \mathrm{SM}_t(s_t, x_0^{(i)})$.

Yale

Likelihood identity

Recall:

- $\nabla \log p_t = \arg\min_{s_t : \mathbb{R}^d \to \mathbb{R}^d} \mathbb{E}_{x_0 \sim p_0} \mathrm{SM}_t(s_t, x_0)$.
- $\widehat{s}_t = \arg\min_{s_t \in \mathcal{S}_t} n^{-1} \sum_{i=1}^n \mathrm{SM}_t(s_t, x_0^{(i)})$.

Lemma:

$$-\log p_0(x_0) = \int_0^T \mathrm{SM}_t(\nabla \log p_t, x_0)\, \mathrm{d}t + C_{d,T} + O(e^{-2T}),$$

where $C_{d,T} = \frac{d}{2} \log(2\pi \mathrm{e}\,(1 - e^{-2T}))$.

We do not claim novelty: see, e.g., [Song, Durkan, Murray, Ermon ’21; Chen, Liu, Theodorou ’22; Li, Yan ’24; …].

Parameter estimation

Parameter estimation

Setting: $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, data $x_0^{(1)}, \ldots, x_0^{(n)} \overset{\text{i.i.d.}}{\sim} p_{\theta^\star}$.

Parameter estimation

Setting: $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, data $x_0^{(1)}, \ldots, x_0^{(n)} \overset{\text{i.i.d.}}{\sim} p_{\theta^\star}$.

Prior works

- [Koehler, Heckett, Risteski '23] studied the implicit score matching (ISM) estimator:
  $\widehat{\theta}_n^{\mathrm{ISM}} := \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \{\|\nabla \log p_\theta(x_0^{(i)})\|^2 + 2\,\Delta \log p_\theta(x_0^{(i)})\}$.

  They showed that $\sqrt{n}\,(\widehat{\theta}_n^{\mathrm{ISM}} - \theta^\star) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Sigma^{\mathrm{ISM}})$.

Yale

Parameter estimation

Setting: $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, data $x_0^{(1)}, \ldots, x_0^{(n)} \overset{\text{i.i.d.}}{\sim} p_{\theta^\star}$.

Prior works

- [Koehler, Heckett, Risteski '23] studied the implicit score matching (ISM) estimator:
  $\widehat{\theta}_n^{\mathrm{ISM}} := \arg\min_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^n \{\|\nabla \log p_\theta(x_0^{(i)})\|^2 + 2\,\Delta \log p_\theta(x_0^{(i)})\}$.

  They showed that $\sqrt{n}\,(\widehat{\theta}_n^{\mathrm{ISM}} - \theta^\star) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Sigma^{\mathrm{ISM}})$.

  ⊵ When $\mathscr{P}$ satisfies a "restricted Poincaré inequality", Σ^{ISM} can be bounded in terms of $\Sigma^{\mathrm{MLE}} = \mathcal{I}(\theta^\star)^{-1}$.

Yale

Parameter estimation

Setting: $\mathscr{P} = \{p_\theta : \theta \in \Theta\}$, data $x_0^{(1)}, \ldots, x_0^{(n)} \overset{\text{i.i.d.}}{\sim} p_{\theta^\star}$.

Prior works

- [Koehler, Heckett, Risteski '23] studied the implicit score matching (ISM) estimator:
  $\widehat{\theta}_n^{\mathrm{ISM}} := \arg\min_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^n \{\|\nabla \log p_\theta(x_0^{(i)})\|^2 + 2\,\Delta \log p_\theta(x_0^{(i)})\}$.

  They showed that $\sqrt{n}\,(\widehat{\theta}_n^{\mathrm{ISM}} - \theta^\star) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Sigma^{\mathrm{ISM}})$.

  - ⊵ When $\mathscr{P}$ satisfies a "restricted Poincaré inequality", Σ^{ISM} can be bounded in terms of $\Sigma^{\mathrm{MLE}} = \mathcal{I}(\theta^\star)^{-1}$.
  - ⊵ In general, $\Sigma^{\mathrm{ISM}} \gg \Sigma^{\mathrm{MLE}}$ (provably inefficient!).

Yale

Parameter estimation

Prior works

- Since a Poincaré inequality is classically related to *mixing times* for Markov chains, their main message was:

$$\text{rapid mixing} \qquad \leftrightsquigarrow \qquad \text{statistical efficiency}$$

Parameter estimation

Prior works

- Since a Poincaré inequality is classically related to *mixing times* for Markov chains, their main message was:

$$\text{rapid mixing} \qquad \rightsquigarrow \qquad \text{statistical efficiency}$$

- For Gaussian mixtures, [Shah, Chen, Klivans ’23; Qin, Risteski ’24] established polynomial sample complexity via score matching along other diffusions.

DDPM score matching and parameter estimation

To estimate $\theta^\star$, let us minimize the DDPM score matching loss:

$$\widehat{\theta}_n^{\mathrm{DDPM}} := \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \int_0^T \mathrm{SM}_t(\nabla \log p_{\theta,t}, x_0^{(i)}) \, \mathrm{d}t$$

DDPM score matching and parameter estimation

To estimate $\theta^\star$, let us minimize the DDPM score matching loss:

$$\widehat{\theta}_n^{\mathrm{DDPM}} := \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\int_0^T \mathrm{SM}_t(\nabla \log p_{\theta,t}, x_0^{(i)})\,\mathrm{d}t$$

By the likelihood identity:

$$\arg\min_{\theta\in\Theta}\left\{-\frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_0^{(i)})\right\}$$

$$= \arg\min_{\theta\in\Theta}\left\{\frac{1}{n}\sum_{i=1}^{n}\int_0^T \mathrm{SM}_t(\nabla \log p_{\theta,t}, x_0^{(i)})\,\mathrm{d}t + C_{d,T} + O(e^{-2T})\right\}$$

DDPM score matching achieves full efficiency

Theorem [CKMM '25]: Under standard conditions, provided that $T = T_n$ satisfies $T_n - \frac{1}{2}\log n \to \infty$,

$$\sqrt{n}\,(\widehat{\theta}_n^{\mathrm{DDPM}} - \theta^\star) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Sigma^{\mathrm{MLE}})\,.$$

DDPM score matching achieves full efficiency

Theorem [CKMM '25]: Under standard conditions, provided that $T = T_n$ satisfies $T_n - \frac{1}{2}\log n \to \infty$,

$$\sqrt{n}\,(\widehat{\theta}_n^{\mathrm{DDPM}} - \theta^\star) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Sigma^{\mathrm{MLE}})\,.$$

Parameter estimation is as easy as (properly) learning scores.

Density estimation

DDPM score matching and density estimation

Likelihood identity:

$$-\log p_0(x_0) = \int_0^T \mathrm{SM}_t(\nabla \log p_t, x_0)\,\mathrm{d}t + C_{d,T} + O(e^{-2T})\,.$$

Yale

DDPM score matching and density estimation

Likelihood identity:

$$-\log p_0(x_0) = \int_0^T \mathrm{SM}_t(\nabla \log p_t, x_0)\, \mathrm{d}t + C_{d,T} + O(e^{-2T})\,.$$

An obvious idea is to estimate $-\log p_0(x_0)$ by outputting

$$-\log \widehat{p}_0(x_0) := \int_0^T \mathrm{SM}_t(\widehat{s}_t, x_0)\, \mathrm{d}t + C_{d,T}\,.$$

Yale

Reduction framework

Theorem [CKMM '25]: The DDPM density estimator achieves $\mathbb{E}_{x_0 \sim p_0} |\log(\widehat{p}(x_0)/p(x_0))| \leq \varepsilon$ in $\mathrm{poly}(d, L, 1/\varepsilon)$ time, where:

- $\|\nabla \log p_0\|_{\mathrm{Lip}} \leq L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \leq \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_t - \nabla \log p_t\|_{L^2(p_t)}^2 \, \mathrm{d}t \leq \widetilde{O}(\varepsilon^2/d)$.

Reduction framework

Theorem [CKMM '25]: The DDPM density estimator achieves $\mathbb{E}_{x_0 \sim p_0} |\log(\widehat{p}(x_0)/p(x_0))| \le \varepsilon$ in $\mathrm{poly}(d, L, 1/\varepsilon)$ time, where:

- $\|\nabla \log p_0\|_{\mathsf{Lip}} \le L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \le \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_t - \nabla \log p_t\|^2_{L^2(p_t)} \, \mathrm{d}t \le \widetilde{O}(\varepsilon^2/d)$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathrm{poly}(d, 1/\tau, 1/\varepsilon)$.

Reduction framework

Theorem [CKMM '25]: The DDPM density estimator achieves $\mathbb{E}_{x_0 \sim p_0} |\log(\widehat{p}(x_0)/p(x_0))| \leq \varepsilon$ in $\mathrm{poly}(d, L, 1/\varepsilon)$ time, where:

- $\|\nabla \log p_0\|_{\mathsf{Lip}} \leq L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \leq \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_t - \nabla \log p_t\|^2_{L^2(p_t)} \, \mathrm{d}t \leq \widetilde{O}(\varepsilon^2/d)$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathrm{poly}(d, 1/\tau, 1/\varepsilon)$.

Remarks: $\widehat{p} \geq 0$, but $\int \widehat{p} \neq 1$.

Reduction framework

Theorem [CKMM '25]: The DDPM density estimator achieves $\mathbb{E}_{x_0 \sim p_0} |\log(\widehat{p}(x_0)/p(x_0))| \leq \varepsilon$ in $\mathrm{poly}(d, L, 1/\varepsilon)$ time, where:

- $\|\nabla \log p_0\|_{\mathrm{Lip}} \leq L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \leq \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_t - \nabla \log p_t\|^2_{L^2(p_t)} \, \mathrm{d}t \leq \widetilde{O}(\varepsilon^2/d)$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathrm{poly}(d, 1/\tau, 1/\varepsilon)$.

Remarks: $\widehat{p} \geq 0$, but $\int \widehat{p} \neq 1$.

It implies that on 99% of the space, $e^{-\varepsilon} p \leq \widehat{p} \leq e^{\varepsilon} p$.

Yale

Reduction framework

Theorem [CKMM '25]: The DDPM density estimator achieves $\mathbb{E}_{x_0 \sim p_0} |\log(\widehat{p}(x_0)/p(x_0))| \leq \varepsilon$ in $\mathrm{poly}(d, L, 1/\varepsilon)$ time, where:

- $\|\nabla \log p_0\|_{\mathrm{Lip}} \leq L$ and $\mathbb{E}_{x_0 \sim p_0}[\|x_0\|^2] \leq \mathrm{poly}(d)$.
- $\int_0^T \|\widehat{s}_t - \nabla \log p_t\|^2_{L^2(p_t)} \, \mathrm{d}t \leq \widetilde{O}(\varepsilon^2/d)$.

The Lipschitz score assumption is removed by stopping early at time τ, with complexity $\mathrm{poly}(d, 1/\tau, 1/\varepsilon)$.

PAC density estimation is as easy as learning the scores.

Example: application to the Hölder class

Let $\mathscr{H}_s(C, L)$ consist of Hölder densities on $[-1, 1]$. (Here, s = smoothness, C = lower bd. on density, L = size of Hölder ball.)

[Dou, Kotekal, Xu, Zhou '24] obtained optimal rates of score estimation for $\mathscr{H}_s(C, L)$, leading to a minimax optimal sampler.

Example: application to the Hölder class

Let $\mathscr{H}_s(C, L)$ consist of Hölder densities on $[-1, 1]$. (Here, s = smoothness, C = lower bd. on density, L = size of Hölder ball.)

[Dou, Kotekal, Xu, Zhou '24] obtained optimal rates of score estimation for $\mathscr{H}_s(C, L)$, leading to a minimax optimal sampler.

Theorem [CKMM '25]: There is a density estimator based on DDPM score matching such that for the L^1 risk $\mathscr{R}(\widehat{p}, p) := \int_{[-1,1]} \mathbb{E}|\widehat{p}(x_0) - p(x_0)|$, the estimator achieves the minimax risk $n^{-2s/(2s+1)}$ over $\mathscr{H}_s(C, L)$ up to a $\sqrt{\log n}$ factor.

Yale

Example: application to the Hölder class

Let $\mathscr{H}_s(C, L)$ consist of Hölder densities on $[-1, 1]$. (Here, s = smoothness, C = lower bd. on density, L = size of Hölder ball.)

[Dou, Kotekal, Xu, Zhou '24] obtained optimal rates of score estimation for $\mathscr{H}_s(C, L)$, leading to a minimax optimal sampler.

> **Theorem [CKMM '25]:** There is a density estimator based on DDPM score matching such that for the L^1 risk $\mathscr{R}(\widehat{p}, p) := \int_{[-1,1]} \mathbb{E}|\widehat{p}(x_0) - p(x_0)|$, the estimator achieves the minimax risk $n^{-2s/(2s+1)}$ over $\mathscr{H}_s(C, L)$ up to a $\sqrt{\log n}$ factor.

See paper for a Gaussian mixture example.

Computational lower bounds

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?
- Some frameworks, such as SQ, are information-theoretic, so they apply to density estimators and generators.

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?
- Some frameworks, such as SQ, are information-theoretic, so they apply to density estimators and generators.
- What about lower bounds from computational complexity?

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?
- Some frameworks, such as SQ, are information-theoretic, so they apply to density estimators and generators.
- What about lower bounds from computational complexity?
  - ⊵ Cryptographic hardness: Solving a task is as hard as breaking a cryptosystem.

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?
- Some frameworks, such as SQ, are information-theoretic, so they apply to density estimators and generators.
- What about lower bounds from computational complexity?
  - Cryptographic hardness: Solving a task is as hard as breaking a cryptosystem.
  - Recently, cryptographic hardness has been proven for some density estimation tasks.

Computational lower bounds

- There are various frameworks for proving hardness of density estimation. Do they lead to hardness of score estimation?
- Some frameworks, such as SQ, are information-theoretic, so they apply to density estimators and generators.
- What about lower bounds from computational complexity?
  - Cryptographic hardness: Solving a task is as hard as breaking a cryptosystem.
  - Recently, cryptographic hardness has been proven for some density estimation tasks.
  - Cryptographic hardness results for learning a sampler remain elusive.

Cryptographic hardness for learning score functions

Recently, [Song '24] proved crypto hardness of learning score functions for Gaussian mixtures via a tailored argument.

Cryptographic hardness for learning score functions

Recently, [Song '24] proved crypto hardness of learning score functions for Gaussian mixtures via a tailored argument.

Our reduction framework yields a *general* blueprint:

To prove crypto hardness for learning the scores of a family $\mathscr{P}$:

1. Check that $\mathscr{P}$ satisfies the conditions of our reduction.
2. Prove that PAC density estimation over $\mathscr{P}$ is crypto hard.

Cryptographic hardness for learning score functions

Recently, [Song '24] proved crypto hardness of learning score functions for Gaussian mixtures via a tailored argument.

Our reduction framework yields a *general* blueprint:

To prove crypto hardness for learning the scores of a family $\mathscr{P}$:

1. Check that $\mathscr{P}$ satisfies the conditions of our reduction.
2. Prove that PAC density estimation over $\mathscr{P}$ is crypto hard.

Learning the scores is as hard as PAC density estimation.

Yale

Application to Gaussian mixtures

Corollary [CKMM '25]: For any $\varepsilon > 0$, it is cryptographically hard to learn the score functions of mixtures of Gaussians with up to d^{ε} components.

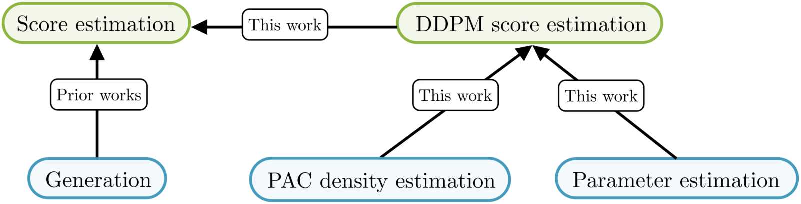
Application to Gaussian mixtures

Corollary [CKMM '25]: For any $\varepsilon > 0$, it is cryptographically hard to learn the score functions of mixtures of Gaussians with up to d^{ε} components.

Reduction chain for experts:

- score estimation $\leftarrow$ PAC density estimation (our framework)
- $\leftarrow$ CLWE (following [Bruna, Regev, Song, Tang '21])
- $\leftarrow$ LWE (following [Gupte, Vafa, Vaikuntanathan '22])
- $\leftarrow$ lattice problems [Regev '09]
- $\leftarrow$ post-quantum cryptography.

Summary



Thank you for your attention!