DDPM Score Matching and Distribution Learning

Sinho Chewi
Yale University

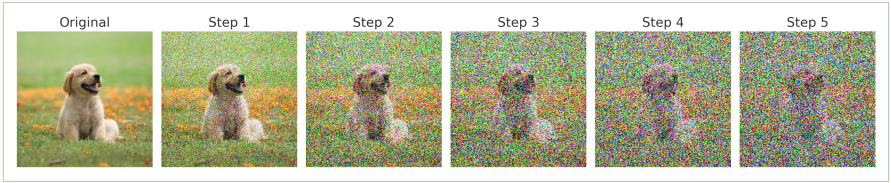
Alkis Kalavasis
Yale University

Omar Montasser
Yale University

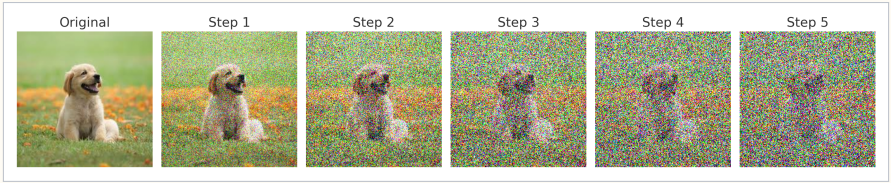
Anay Mehrotra
Stanford University

COLT, July 2, 2026

DDPM Diffusion Models



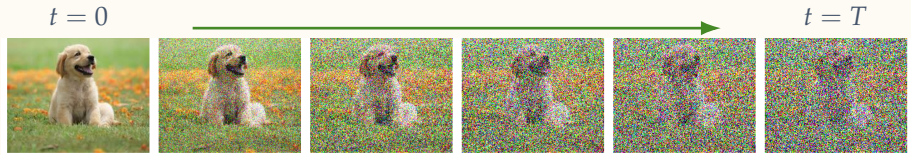
DDPM Diffusion Models



forward noising

reverse generation via scores

DDPM Diffusion Models

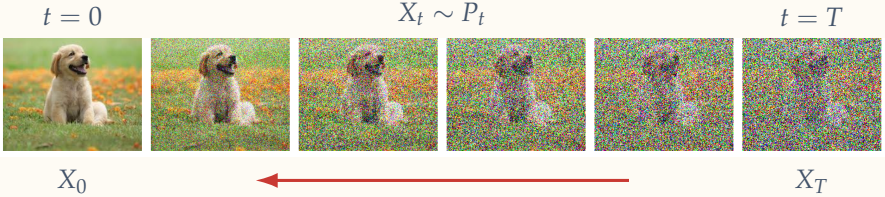


Noising process.
Start with data $X_0 \sim P$ and slowly add Gaussian noise:

$$X_t = e^{-t} X_0 + \sqrt{1 - e^{-2t}}\, Z_t\,, \qquad Z_t \sim \mathcal{N}(0, I_d)\,,$$
$$\mathrm{d}X_t = -X_t\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,, \qquad X_0 \sim P\,.$$

This SDE smoothly transforms complex data-distribution P to $\mathcal{N}(0, I)$ by injecting noise.

DDPM Diffusion Models



Forward SDE can be **time-reversed** given accurate estimates of scores.

Reverse SDE with terminal time T.

$$\mathrm{d}X_{T-t} = \{X_{T-t} + 2\nabla \log P_{T-t}(X_{T-t})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\bar{B}_t\,, \quad X_T \sim P_T\,.$$

What does it mean to learn a distribution?

Question.
Let $\mathcal{P}$ be a class of distributions over $\mathbb{R}^d$. Given i.i.d. samples from P, what does it mean to "learn" P?

Density Estimation

Density Evaluator [Kearns–Mansour–Ron–Rubinfeld–Schapire–Sellie, STOC'94]**.**

An *evaluator* $\widehat{P}$ for P is a function $\mathbb{R}^d \to \mathbb{R}$ such that on any input $x \in \mathbb{R}^d$, it outputs an estimate $\widehat{P}(x)$ of the log-density $\log P(x)$.

Density Estimation

Density Evaluator [Kearns–Mansour–Ron–Rubinfeld–Schapire–Sellie, STOC'94]**.**

An *evaluator* $\widehat{P}$ for P is a function $\mathbb{R}^d \to \mathbb{R}$ such that on any input $x \in \mathbb{R}^d$, it outputs an estimate $\widehat{P}(x)$ of the log-density $\log P(x)$.

Let $\mathcal{P}$ be a class of distributions over $\mathbb{R}^d$.

Density Estimation.

Given i.i.d. samples from $P \in \mathcal{P}$, compute an ε-*evaluator* $\widehat{P}$ of P

$$\mathbb{E}_{x \sim P} \left| (-\log P(x)) - \widehat{P}(x) \right| \leq \varepsilon \,.$$

Learning to Sample

Generator [Kearns–Mansour–Ron–Rubinfeld–Schapire–Sellie, STOC'94]**.**

A *generator* $\widehat{P}$ for P is a function $\mathbb{R}^d \to \mathbb{R}^d$ such that on random seed $z \sim \mathcal{N}(0, I)$, it outputs $x \sim \widehat{P}(z)$ such that x has law close to P.

Learning to Sample

Generator [Kearns–Mansour–Ron–Rubinfeld–Schapire–Sellie, STOC'94]**.**

A *generator* $\widehat{P}$ for P is a function $\mathbb{R}^d \to \mathbb{R}^d$ such that on random seed $z \sim \mathcal{N}(0, I)$, it outputs $x \sim \widehat{P}(z)$ such that x has law close to P.

Let $\mathcal{P}$ be a class of distributions over $\mathbb{R}^d$.

Learning to Sample.

Given i.i.d. samples S from $P \in \mathcal{P}$, compute an ε-*generator* $\widehat{P}$ of P

$$\mathrm{Law}(\widehat{P}(Z) \mid S) \approx_\varepsilon P \qquad \text{on random seed } Z\,.$$

What does it mean to learn a distribution?

Question.
Let $\mathcal{P}$ be a class of distributions over $\mathbb{R}^d$. Given i.i.d. samples from P, what does it mean to "learn" P?

Learning to Sample.
Given a random seed z, output a sample close to P.

Density Estimation.
Given x, estimate the log-density $\log P(x)$.

Different Lenses on Distribution Learning

Statistical lens.

With sufficient time and samples, generators $\rightleftarrows$ evaluators.

Different Lenses on Distribution Learning

Statistical lens.

With sufficient time and samples, generators $\rightleftarrows$ evaluators.

Computational lens.

With computational restrictions, generators $\not\rightleftarrows$ evaluators.

Score Estimation $\to$ Sampling

Informal Theorem [Chen et al., ICLR'23; Lee-Lu-Tan, ALT'23]**.**
Suppose score is L-Lipschitz, $m_2(P) \le \mathrm{poly}(d)$, and score estimates satisfy

$$\sup_{t \ge 0} \ \mathbb{E}_{X_t \sim P_t} \left\| \widehat{s}_t(X_t) - \nabla \log P_t(X_t) \right\|^2 \le \widetilde{O}(\varepsilon^2) \,.$$

Then, DDPM using $\widehat{s}_t$ outputs a distribution $\widehat{P}$ such that

$$\mathrm{d_{TV}}(\widehat{P}, P) \le \varepsilon \,.$$

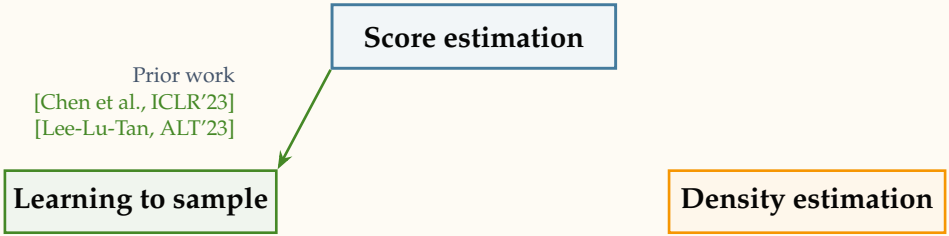
It uses $\mathrm{poly}(d, L, 1/\varepsilon)$ score-oracle calls and additional computation.

Takeaway

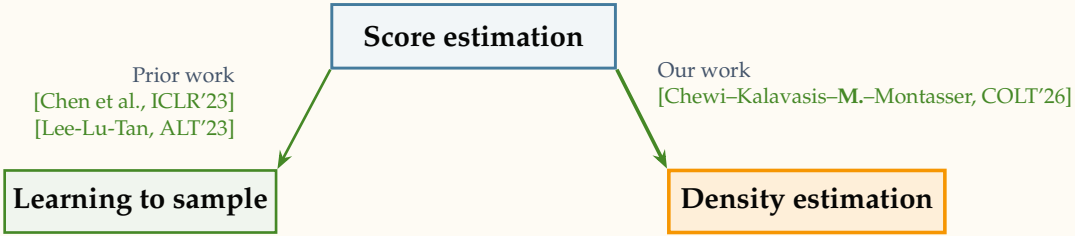
Learning to sample

Density estimation

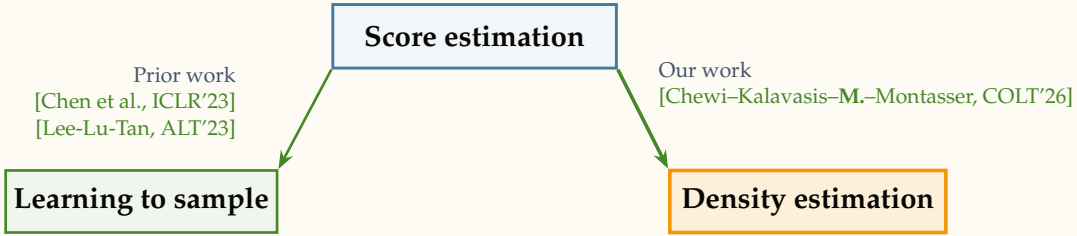
Takeaway


Score estimation
Prior work
[Chen et al., ICLR'23]
[Lee-Lu-Tan, ALT'23]
Learning to sample
Density estimation

Takeaway


Score estimation
Prior work
[Chen et al., ICLR'23]
[Lee-Lu-Tan, ALT'23]
Our work
[Chewi–Kalavasis–M.–Montasser, COLT'26]
Learning to sample
Density estimation

Takeaway



Takeaway. Accurate DDPM score estimation → log-density evaluator.

Our Result

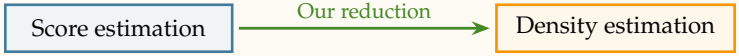
Informal Theorem [Chewi–Kalavasis–**M.**–Montasser, COLT'26].
Let scores are L-Lipschitz, $m_2(P) \le \mathrm{poly}(d)$, and a score-estimation, for suitable $0 < \tau < T$,

$$\int_\tau^T \mathrm{E}_{X_t \sim P_t} \left\| \widehat{s}_t(X_t) - \nabla \log P_t(X_t) \right\|^2 \mathrm{d}t \le \widetilde{O}(\varepsilon^2/d)\,.$$

Then there is an evaluator $\widehat{\ell}$ for P such that

$$\mathbb{E}_{x_0 \sim P} \left| \widehat{\ell}(x_0) - \log P(x_0) \right| \le \varepsilon\,.$$

It uses $\mathrm{poly}(d, L, 1/\varepsilon)$ score-oracle calls and additional computation.



Algorithmic Implications

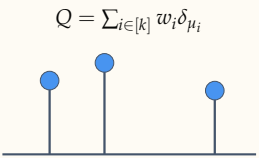
Corollary 1.
Efficient score estimation $\longrightarrow$ Efficient density estimation

Gaussian location mixtures.
Let $\mathcal{P}$ be the non-parametric class of Gaussian location mixtures on $\mathbb{R}^d$: $P = Q \star \mathcal{N}(0, \sigma^2 I_d)$, where Q is supported on a union of k constant-radius balls, each carrying mass at least $w_{\min}$.

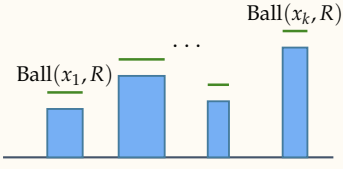
Algorithmic Implications

Corollary 1.
Efficient score estimation $\longrightarrow$ Efficient density estimation

Gaussian location mixtures.
Let $\mathcal{P}$ be the non-parametric class of Gaussian location mixtures on $\mathbb{R}^d$: $P = Q \star \mathcal{N}(0, \sigma^2 I_d)$, where Q is supported on a union of k constant-radius balls, each carrying mass at least $w_{\min}$.



P is a spherical GMM



Algorithmic Implications

Let P be a Gaussian location mixture on $\mathbb{R}^d$ with k components and $w_{\min} > \varepsilon$.

[Gatmiry–Kelner–Lee, COLT'25] gave a $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ algorithm to learn to sample from P.

Algorithmic Implications

Let P be a Gaussian location mixture on $\mathbb{R}^d$ with k components and $w_{\min} > \varepsilon$.

[Gatmiry–Kelner–Lee, COLT'25] gave a $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ algorithm to learn to sample from P.

Theorem (Score Estimation) [Gatmiry–Kelner–Lee, COLT'25].
There is a $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ time algorithm for estimating the *scores* along the DDPM path, using samples from P.

Algorithmic Implications

Let P be a Gaussian location mixture on $\mathbb{R}^d$ with k components and $w_{\min} > \varepsilon$.

[Gatmiry–Kelner–Lee, COLT'25] gave a $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ algorithm to learn to sample from P.

Theorem (Score Estimation) [Gatmiry–Kelner–Lee, COLT'25].
There is a $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ time algorithm for estimating the *scores* along the DDPM path, using samples from P.

Theorem (Density Estimation) [Chewi–Kalavasis–**M.**–Montasser, COLT'26].
There is an algorithm that outputs an ε-*evaluator* for P in $d^{\mathrm{polylog}(d,k,1/\varepsilon)}$ time, using samples from P.

Lower Bound Implications

Corollary 2.
If the family of distributions $\mathcal{P}$ satisfies regularity conditions, then:
Lower bound for density estimation for $\mathcal{P}$
$\longrightarrow$ Lower bound for score estimation for $\mathcal{P}$

Instantiating this for GMMs conceptually recovers [Song, NeurIPS'24]'s result

Conclusion and open problems

Score estimation is not only sufficient for sampling but also for PAC log-density estimation

Conclusion and open problems

Score estimation is not only sufficient for sampling but also for PAC log-density estimation

Using score estimation, is it possible to output a *proper* density estimator?

Analogous results in discrete domains, infinite-dimensional spaces, manifolds?

Algorithms for density estimation of well-conditioned GMMs?

Beyond GMMs, when is score estimation cryptographically hard?