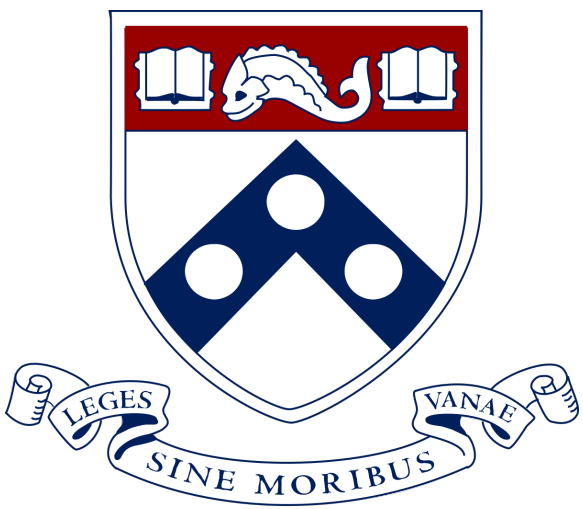
To Intrinsic Dimension and Beyond: Efficient Sampling in Diffusion Models

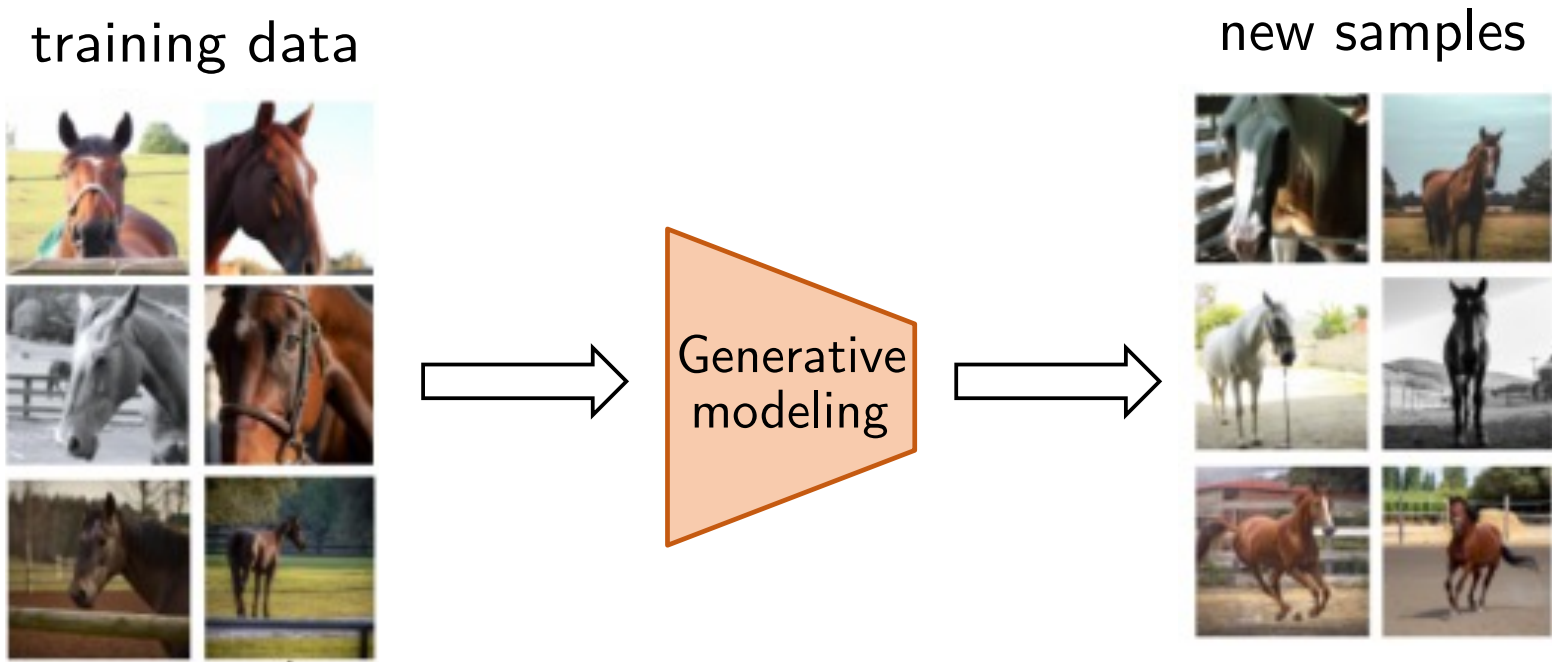


Yuting Wei

Statistics & Data Science, Wharton
University of Pennsylvania

September 9, 2025

Generative models



- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a general distribution}}$ $(1 \leq i \leq N)$ in $\mathbb{R}^d$
- Generate new samples $Y \sim p_{\text{data}}$

A natural approach: density estimation

- learn the distribution directly (parameterized by θ):

$$p(x \mid \theta) = \frac{e^{-f_\theta(x)}}{Z_\theta},$$

where Z_θ is a normalizing constant depending on θ

A natural approach: density estimation

- learn the distribution directly (parameterized by θ):

$$p(x \mid \theta) = \frac{e^{-f_\theta(x)}}{Z_\theta},$$

  where Z_θ is a normalizing constant depending on θ

- Use maximum likelihood (or posterior) to estimate θ:

$$\max_\theta \sum_{i=1}^{N} \log p(X_i \mid \theta)$$

A natural approach: density estimation

- learn the distribution directly (parameterized by θ):

$$p(x \mid \theta) = \frac{e^{-f_\theta(x)}}{Z_\theta},$$

  where Z_θ is a normalizing constant depending on θ

- Use maximum likelihood (or posterior) to estimate θ:

$$\max_\theta \sum_{i=1}^{N} \log p(X_i \mid \theta) \longrightarrow \text{ Intractable!}$$

Another approach: score function

The **(Stein) score function** of a distribution $p(x)$ is defined as

$$s(x) := \nabla_x \log p_X(x).$$

Note that

$$\begin{aligned} \nabla \log p(x \mid \theta) &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= -\nabla_x f_\theta(x) \end{aligned}$$

getting rid of the annoying Z_θ!

Another approach: score function

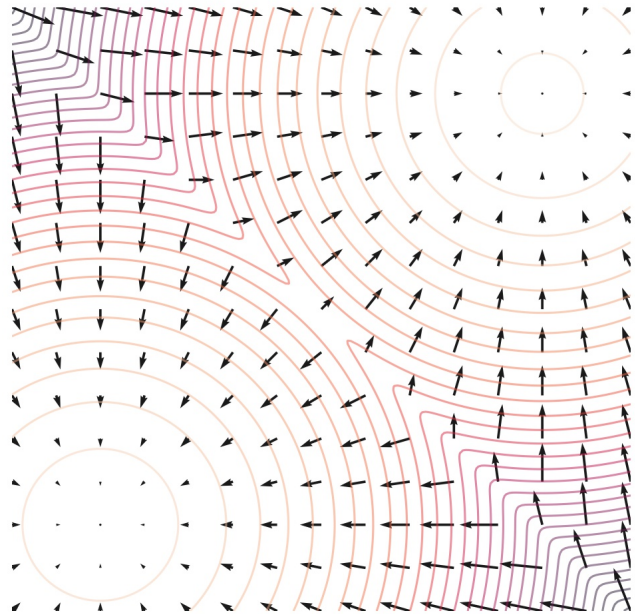
The **(Stein) score function** of a distribution $p(x)$ is defined as

$$s(x) := \nabla_x \log p_X(x).$$

Note that

$$\begin{aligned}\nabla \log p(x \mid \theta) &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= -\nabla_x f_\theta(x)\end{aligned}$$

getting rid of the annoying Z_θ!



Score function of Gaussian mixtures

The score function points towards regions of higher probability.

Langevin dynamics using scores

Unadjusted Langevin algorithm (ULA): from some $x_0 \sim \pi(x)$, perform iterative sampling

$$x_{t+1} = x_t + \eta s(x_t) + \sqrt{2\eta} z_t,$$

where $z_t \sim \mathcal{N}(0, I_d)$ and η is some learning rate.

Langevin dynamics using scores

Unadjusted Langevin algorithm (ULA): from some $x_0 \sim \pi(x)$, perform iterative sampling

$$x_{t+1} = x_t + \eta s(x_t) + \sqrt{2\eta} z_t,$$

where $z_t \sim \mathcal{N}(0, I_d)$ and η is some learning rate.

- In continuous-time, ULA recovers the Langevin dynamic:

$$\mathrm{d}X_\tau = -\nabla f(X_\tau)\mathrm{d}\tau + 2\mathrm{d}B_\tau$$

Langevin dynamics using scores

Unadjusted Langevin algorithm (ULA): from some $x_0 \sim \pi(x)$, perform iterative sampling

$$x_{t+1} = x_t + \eta s(x_t) + \sqrt{2\eta} z_t,$$

where $z_t \sim \mathcal{N}(0, I_d)$ and η is some learning rate.

- In continuous-time, ULA recovers the Langevin dynamic:

$$\mathrm{d}X_\tau = -\nabla f(X_\tau)\mathrm{d}\tau + 2\mathrm{d}B_\tau$$

- When $\eta \to 0$, x_t converges to a sample from $p(x)$

Langevin dynamics using scores

Unadjusted Langevin algorithm (ULA): from some $x_0 \sim \pi(x)$, perform iterative sampling

$$x_{t+1} = x_t + \eta s(x_t) + \sqrt{2\eta} z_t,$$

where $z_t \sim \mathcal{N}(0, I_d)$ and η is some learning rate.

- In continuous-time, ULA recovers the Langevin dynamic:

$$\mathrm{d}X_\tau = -\nabla f(X_\tau)\mathrm{d}\tau + 2\mathrm{d}B_\tau$$

- When $\eta \to 0$, x_t converges to a sample from $p(x)$

— Dismay performance in practice. Why?

Manifold hypothesis

- Real-world data live on low-dimensional manifold

Manifold hypothesis

- Real-world data live on low-dimensional manifold
- Reliable score estimation is available only in high-density regions
- However, our initial sample is highly likely in low density regions (where score estimates are poor)

Manifold hypothesis

- Real-world data live on low-dimensional manifold
- Reliable score estimation is available only in high-density regions
- However, our initial sample is highly likely in low density regions (where score estimates are poor)

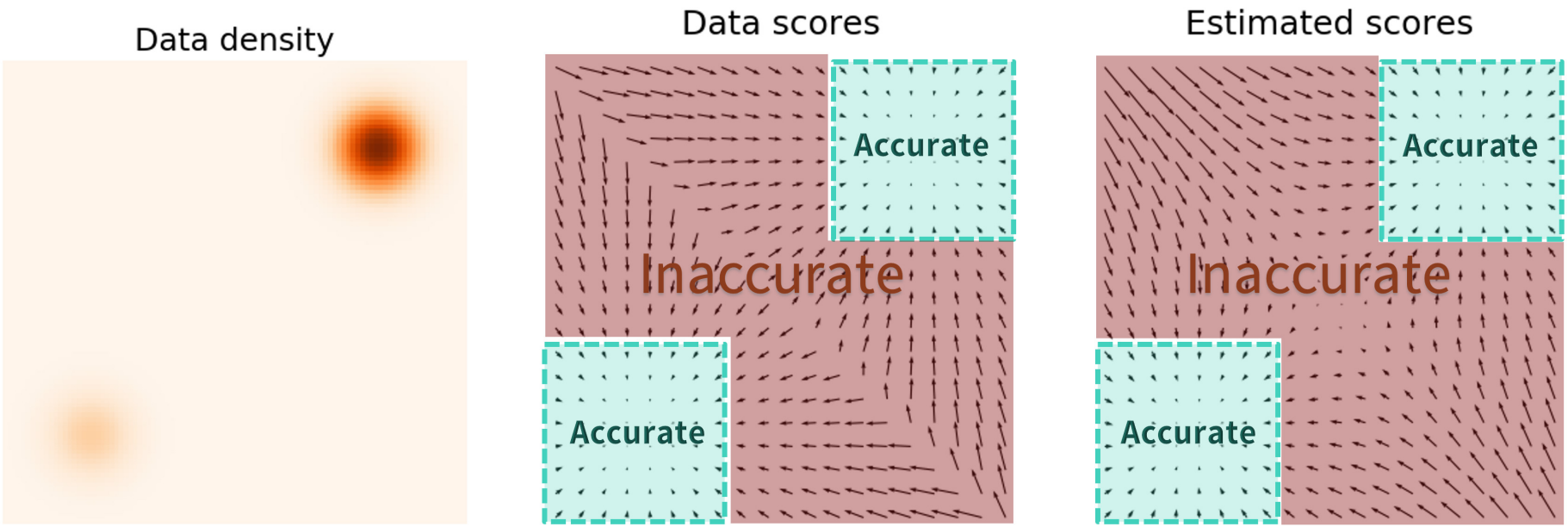


fig. credit: Y. Song

Adding noise to data

- To improve data coverage/score estimation, we can add noise to it

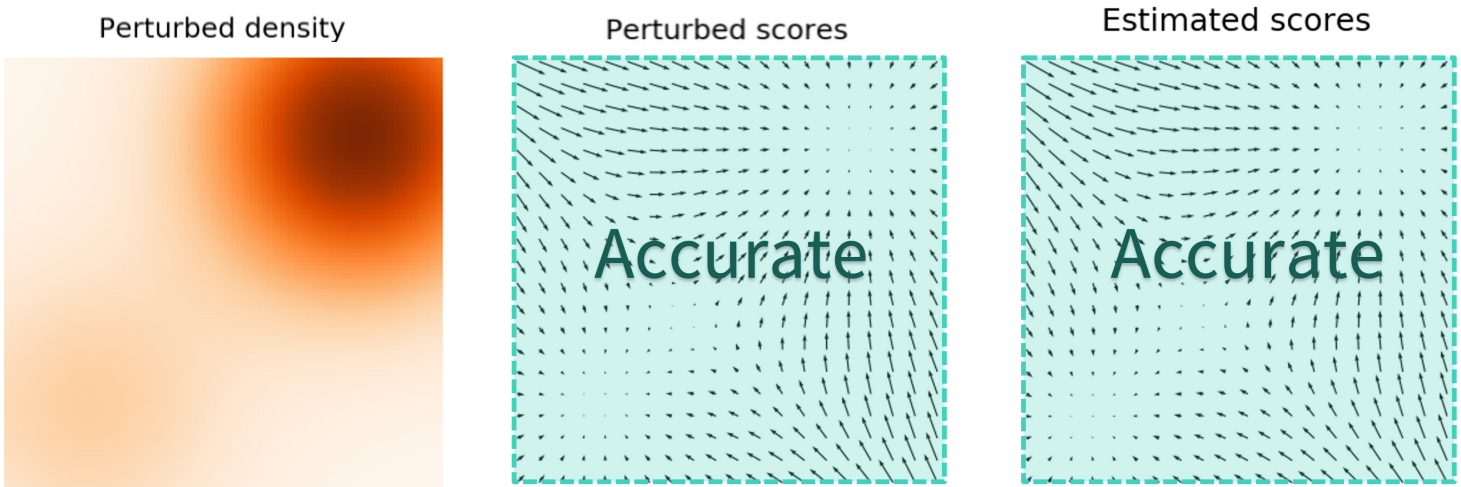


fig. credit: Y. Song

Adding noise to data

- To improve data coverage/score estimation, we can add noise to it
- However, this makes the data distribution different from what we want

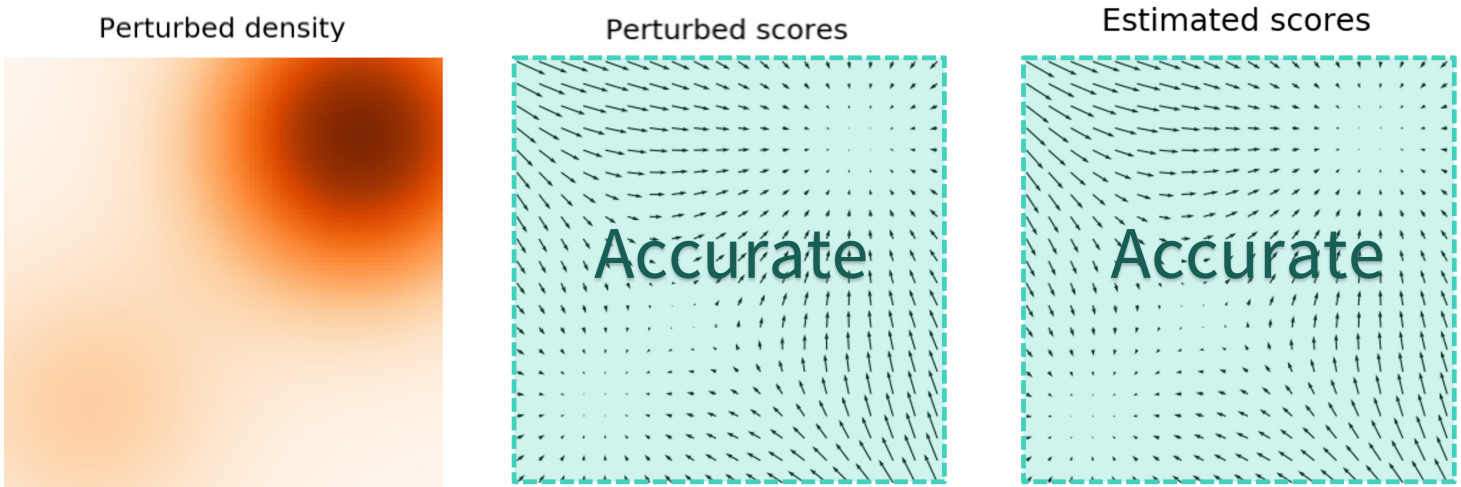


fig. credit: Y. Song

Adding noise to data

- To improve data coverage/score estimation, we can add noise to it
- However, this makes the data distribution different from what we want

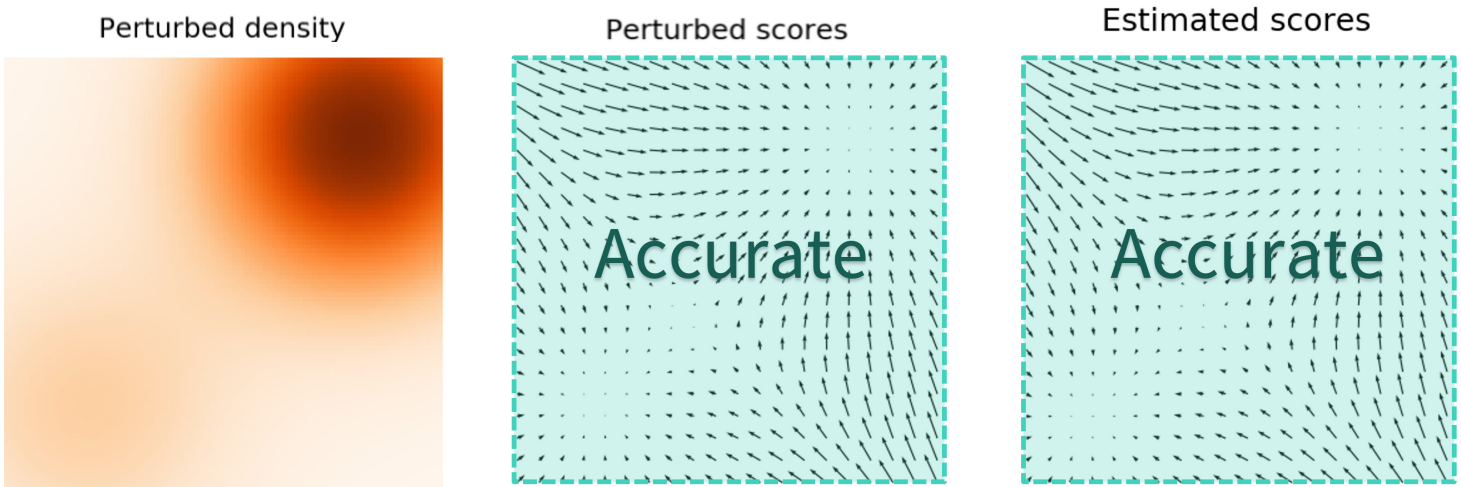


fig. credit: Y. Song

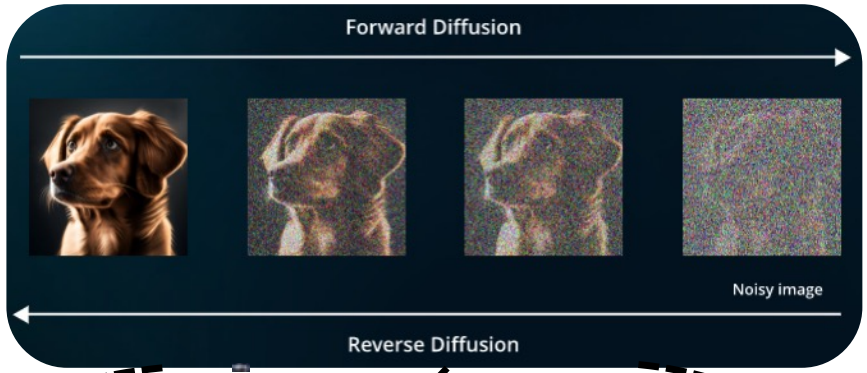
Noise annealing: introducing data perturbation at multiple noise levels!

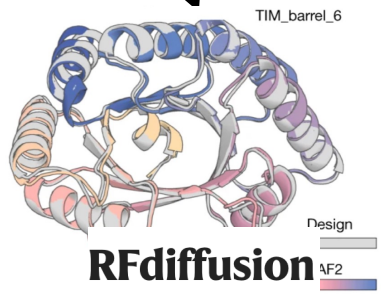
Diffusion models

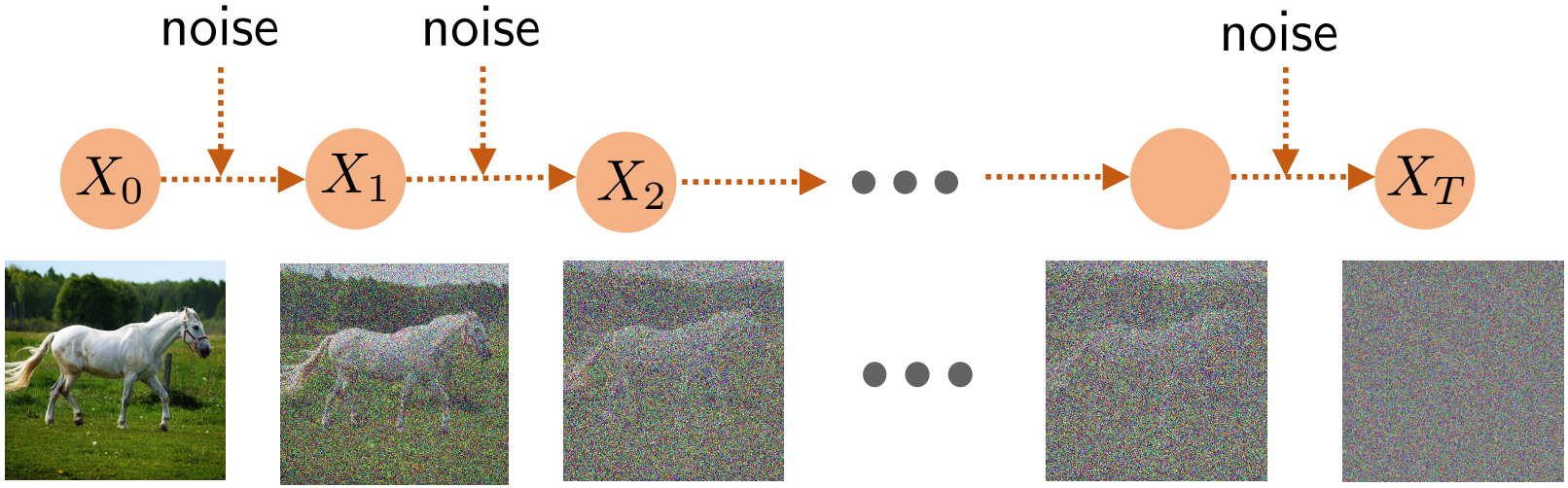
Inspired by nonequilibrium thermodynamics

— Sohl-Dickstein, Weiss, Maheswaranathan, Ganguli '15

Diffusion models

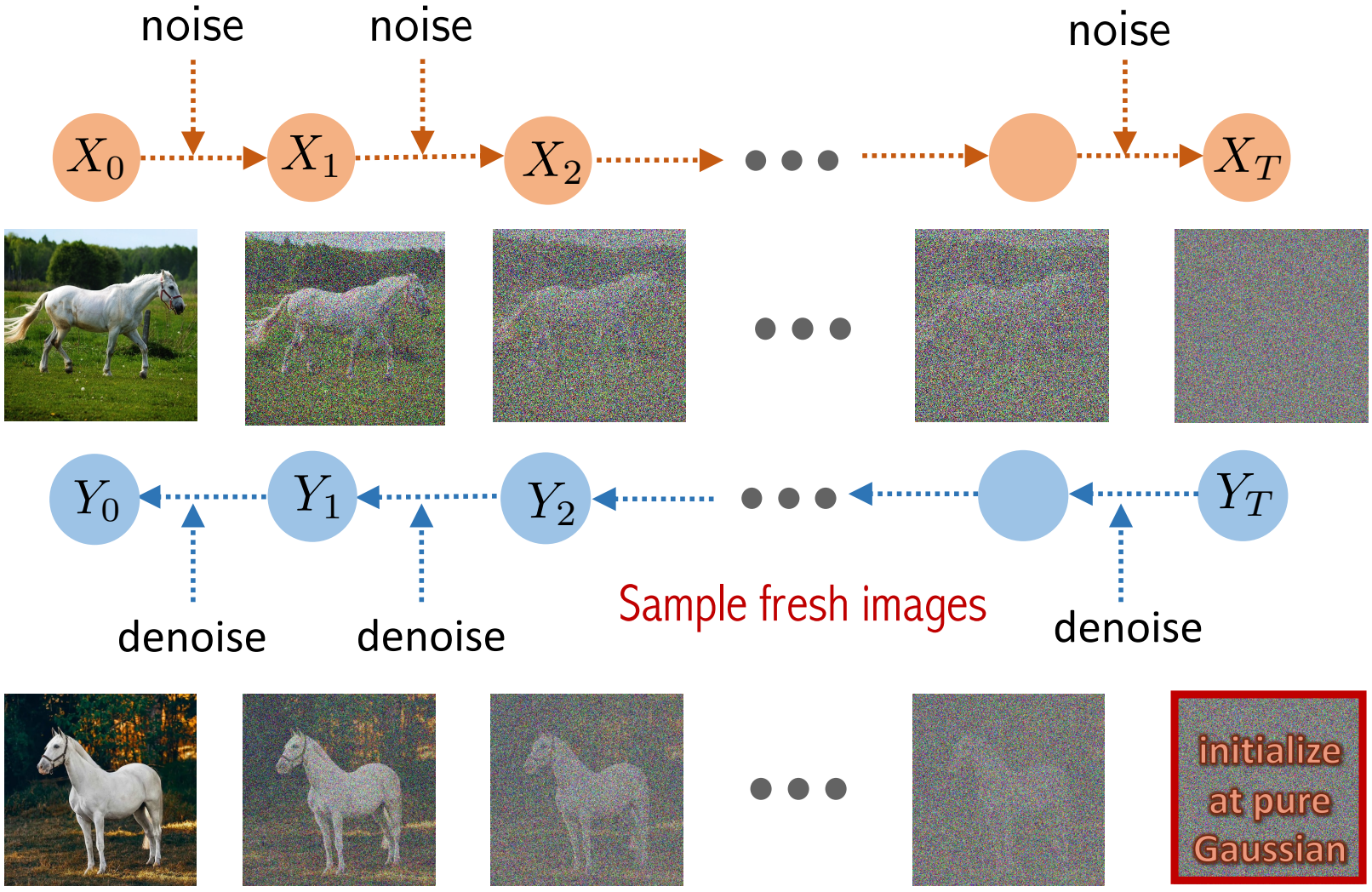






- **forward process:** (progressively) diffuse data into noise

$$X_0 \sim p_{\text{data}}, \quad X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \quad 1 \leq t \leq T$$



- **forward process:** (progressively) diffuse data into noise
- **reverse process:** convert pure noise into data-like distributions

Score is all you need

How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

Score is all you need

How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

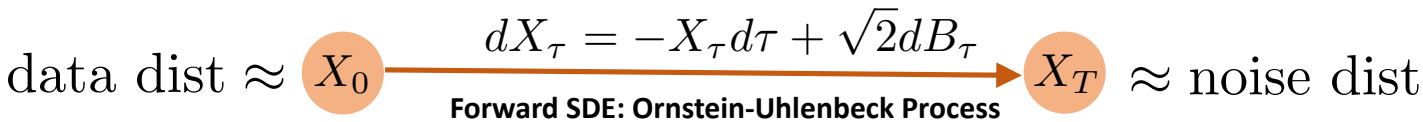
It is feasible as long as one knows the score function $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$

—Anderson'82; Haussmann and Pardoux'86; Song et el.'20...

Score is all you need

How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

It is feasible as long as one knows the score function $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$



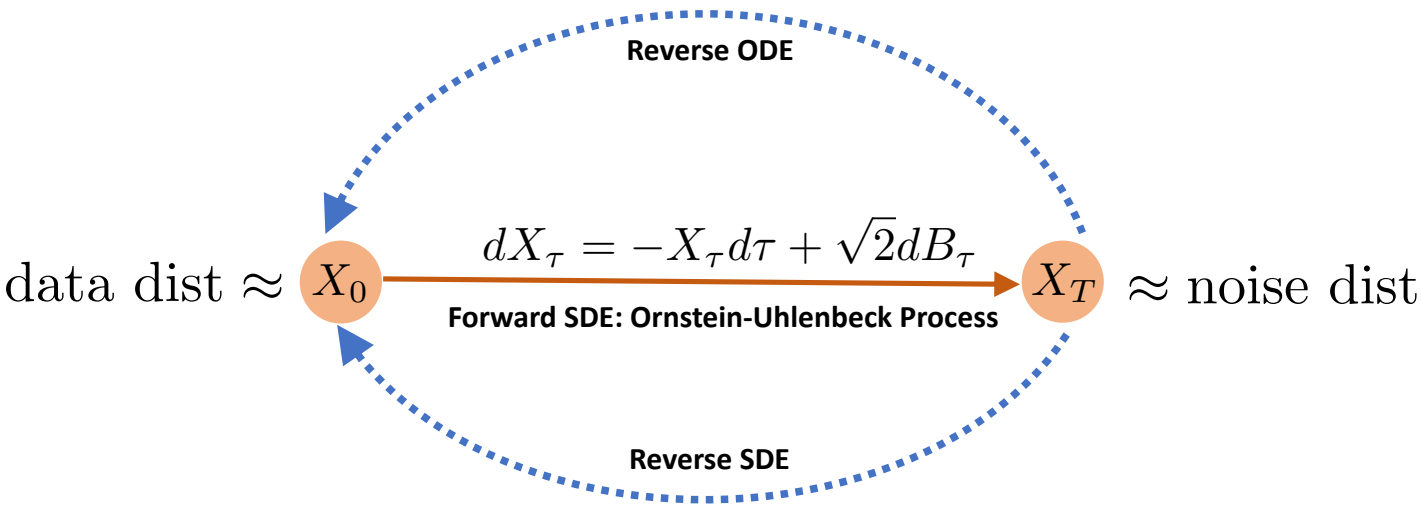
—Anderson'82; Haussmann and Pardoux'86; Song et el.'20...

Score is all you need

How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

It is feasible as long as one knows the score function $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$

$$dY_\tau = \left(Y_\tau + \nabla \log p_{X_{T-\tau}}(Y_\tau)\right) d\tau$$

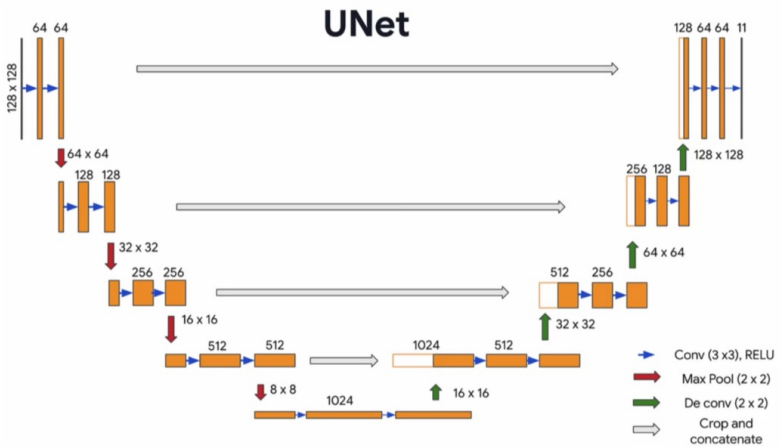


$$dY_\tau = \left(Y_\tau + 2\nabla \log p_{X_{T-\tau}}(Y_\tau)\right) d\tau + \sqrt{2} dB_\tau$$

—Anderson'82; Haussmann and Pardoux'86; Song et el.'20...

A divide-and-conquer approach

— Li, Lu, Tan '22
— Chen, Chewi, Li, Li, Salim, Zhang '22
— Benton, De Bortoli, Doucet, Deligiannidis '23

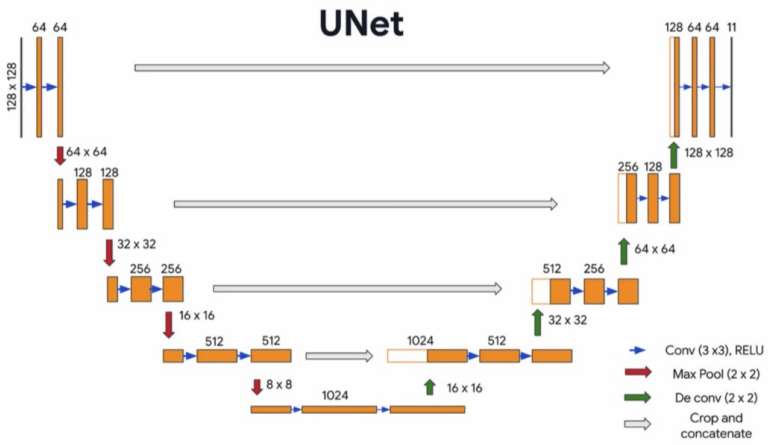


$\Longrightarrow$ learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

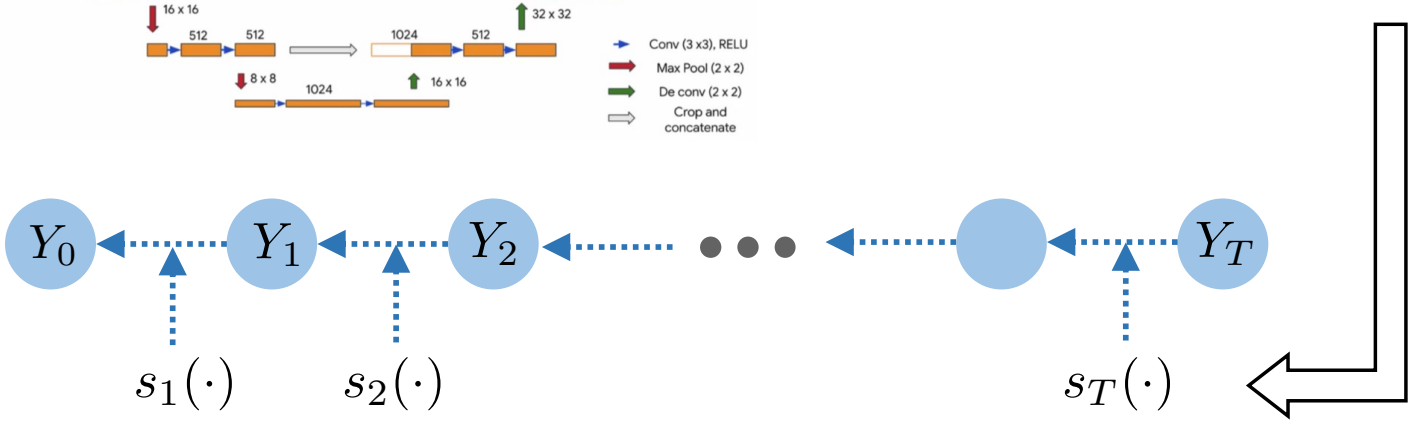
1. **score learning/matching:** learn estimates $s_t(\cdot)$ for $\nabla \log p_{X_t}(\cdot)$: often achieved by neural networks with different architectures

A divide-and-conquer approach

— Li, Lu, Tan '22
— Chen, Chewi, Li, Li, Salim, Zhang '22
— Benton, De Bortoli, Doucet, Deligiannidis '23



$\Longrightarrow$ learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$



1. **score learning/matching:** learn estimates $s_t(\cdot)$ for $\nabla \log p_{X_t}(\cdot)$: often achieved by neural networks with different architectures
2. **data generation:** sampling w/ the aid of score estimates $\{s_t(\cdot)\}$

Score matching via denoising

$$X_0 \sim p_{\mathsf{data}}, \quad X_t = \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1 - \overline{\alpha}_t} \mathcal{N}(0, I_d), \quad 1 \leq t \leq T$$

Score matching via denoising

$$X_0 \sim p_{\text{data}}, \quad X_t = \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\mathcal{N}(0, I_d), \quad 1 \le t \le T$$

Tweedie's formula (Hyvarinen'05; Vincent'11):

$$s^\star(x) = \frac{1}{\sqrt{1-\overline{\alpha}_t}} \mathbb{E}\left[W \;\middle|\; \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t} W = x\right],$$

where the expectation is taken over $W \sim \mathcal{N}(0, I_d), X_0 \sim p_{\text{data}}$.

Score matching via denoising

$$X_0 \sim p_{\mathsf{data}}, \quad X_t = \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\mathcal{N}(0, I_d), \quad 1 \le t \le T$$

Tweedie's formula (Hyvarinen'05; Vincent'11):

$$s^\star(x) = \frac{1}{\sqrt{1-\overline{\alpha}_t}} \mathbb{E}\left[W \;\middle|\; \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t} W = x\right],$$

where the expectation is taken over $W \sim \mathcal{N}(0, I_d), X_0 \sim p_{\mathsf{data}}$.

- nonparametric methods Wibisono et al.'24; Zhang et al.'24; Dou et al.'24
- AMP Wu & Montanari'23
- neural networks Cole and Lu'24, Mei and Wu'23, Oko et al.'23

Sampling: Two mainstream approaches

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

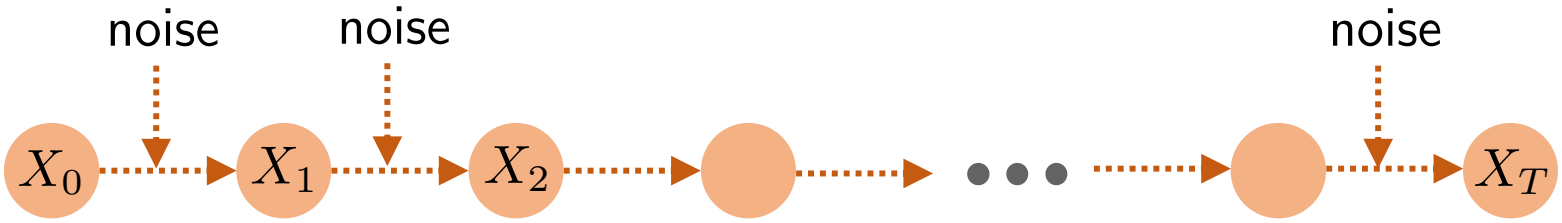
Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

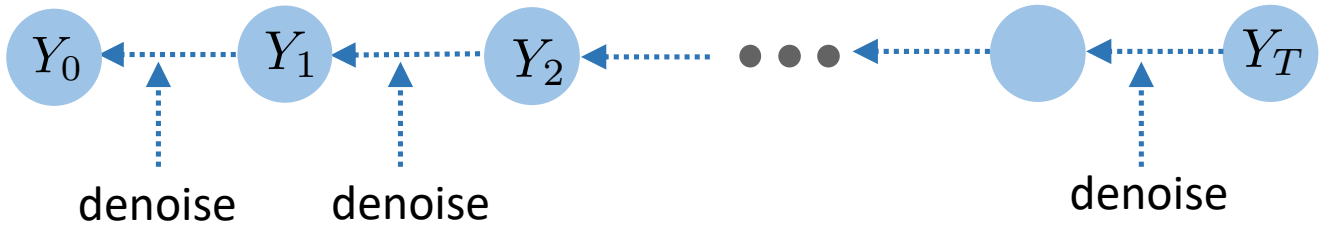
DDPM vs. DDIM



forward process: $X_0 \sim p_{\text{data}}$,

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\,\mathcal{N}(0, I_d), \quad t \geq 1$$

DDPM vs. DDIM

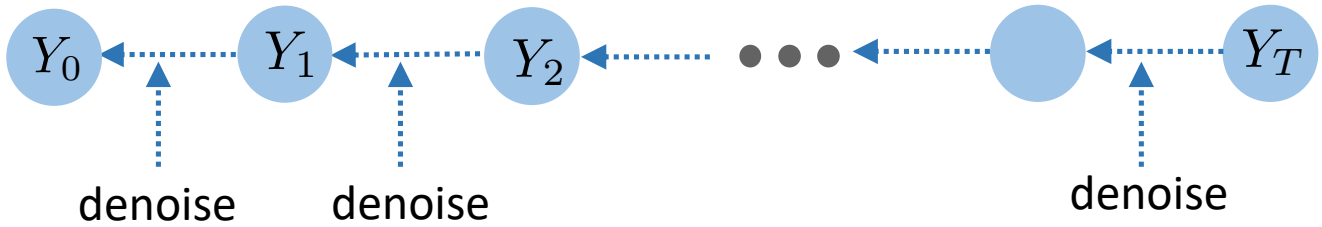


— *Ho, Jain, Abbeel '20*

1. A stochastic sampler: **denoising diffusion probabilistic models**

DDPM

DDPM vs. DDIM



— *Ho, Jain, Abbeel '20*

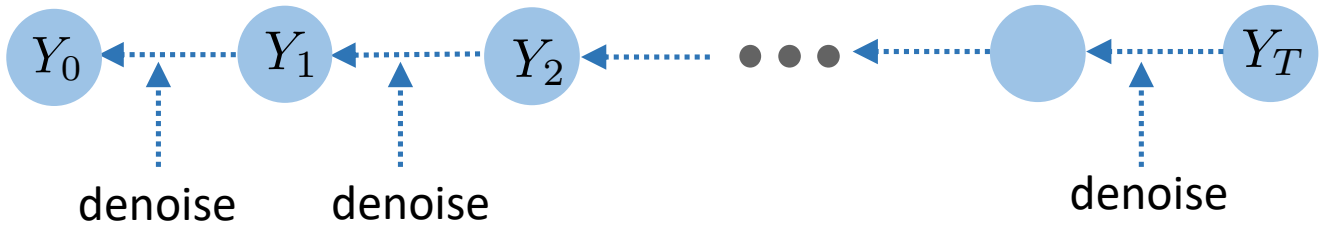
1. A stochastic sampler: **denoising diffusion probabilistic models**

DDPM

$$Y_T \sim \mathcal{N}(0, I_d)$$

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(\underbrace{Y_t + (1-\alpha_t)s_t(Y_t)}_{\text{deterministic}}\Big) + \underbrace{\sqrt{(1-\alpha_t)}\mathcal{N}(0, I_d)}_{\text{stochastic}}, \quad t = T, \cdots, 1$$

DDPM vs. DDIM

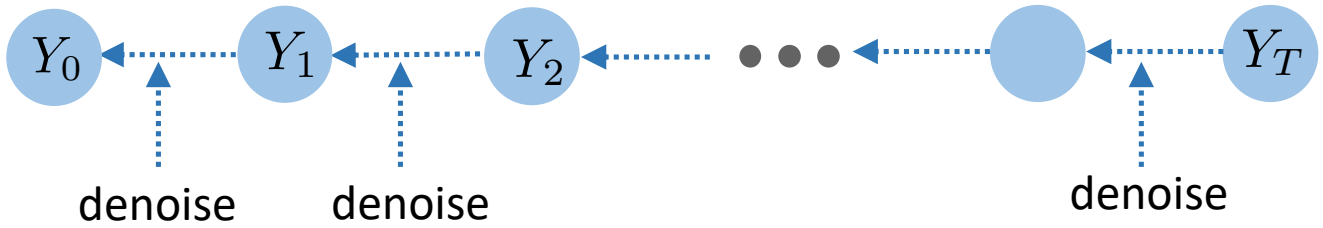


— *Song, Meng, Ermon '20*

2. A <u>deterministic</u> sampler: **denoising diffusion implicit models**

DDIM

DDPM vs. DDIM



— *Song, Meng, Ermon '20*

2. A deterministic sampler: **denoising diffusion implicit models**
   DDIM

$$Y_T \sim \mathcal{N}(0, I_d)$$

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(\underbrace{Y_t + (1-\alpha_t)/2}_{\text{deterministic}}\, s_t(Y_t)\Big), \quad t = T, \cdots, 1$$

Prior theory: Convergence theory for DDIM & DDPM

Theorem (Li, Wei, Chi, Chen '24)

Under mild assumptions on the target distribution, the DDIM sampler obeys (up to log factor)

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}}$$

Prior theory: Convergence theory for DDIM & DDPM

Theorem (Li, Wei, Chi, Chen '24)

Under mild assumptions on the target distribution, the DDIM sampler obeys (up to log factor)

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield TV dist} \le \varepsilon}$ d/ε for small enough ε

Prior theory: Convergence theory for DDIM & DDPM

Theorem (Li, Wei, Chi, Chen '24)

Under mild assumptions on the target distribution, the DDIM sampler obeys (up to log factor)

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\text{score}} + d\varepsilon_{\text{Jacobi}}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield TV dist} \le \varepsilon}$ d/ε for small enough ε
- **robustness:** $\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \propto$ score error measures $\varepsilon_{\text{score}}$ and $\varepsilon_{\text{Jacobi}}$

  – *ℓ_2 score estimation error:* $s_t^\star(X) := \nabla \log p_{X_t}(X)$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim p_{X_t}} \left[\|s_t(X) - s_t^\star(X)\|_2^2\right] \le \varepsilon_{\text{score}}^2$$

Prior theory: Convergence theory for DDIM & DDPM

Theorem (Li, Wei, Chi, Chen '24)

Under mild assumptions on the target distribution, the DDIM sampler obeys (up to log factor)

$$\mathrm{TV}\left(p_{X_1}, p_{Y_1}\right) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\text{score}} + d\varepsilon_{\text{Jacobi}}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield TV dist} \le \varepsilon}$ d/ε for small enough ε
- **robustness:** $\mathrm{TV}\left(p_{X_1}, p_{Y_1}\right) \propto$ score error measures $\varepsilon_{\text{score}}$ and $\varepsilon_{\text{Jacobi}}$
- d-**dependency:** sharp dependence on d using *stochastic localization techniques*

Prior theory: Convergence theory for DDIM & DDPM

Theorem (Li, Wei, Chi, Chen '24)

Under mild assumptions on the target distribution, the DDIM sampler obeys (up to log factor)

$$\mathrm{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\text{score}} + d\varepsilon_{\text{Jacobi}}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield TV dist} \le \varepsilon}$ d/ε for small enough ε
- **robustness:** $\mathrm{TV}\big(p_{X_1}, p_{Y_1}\big) \propto$ score error measures $\varepsilon_{\text{score}}$ and $\varepsilon_{\text{Jacobi}}$
- d-**dependency:** sharp dependence on d using *stochastic localization techniques*

— $\widetilde{O}(d/T)$ *scaling proved for DDPM in Li & Yan'24*

This talk: adaptation to (unknown) low dimensionality

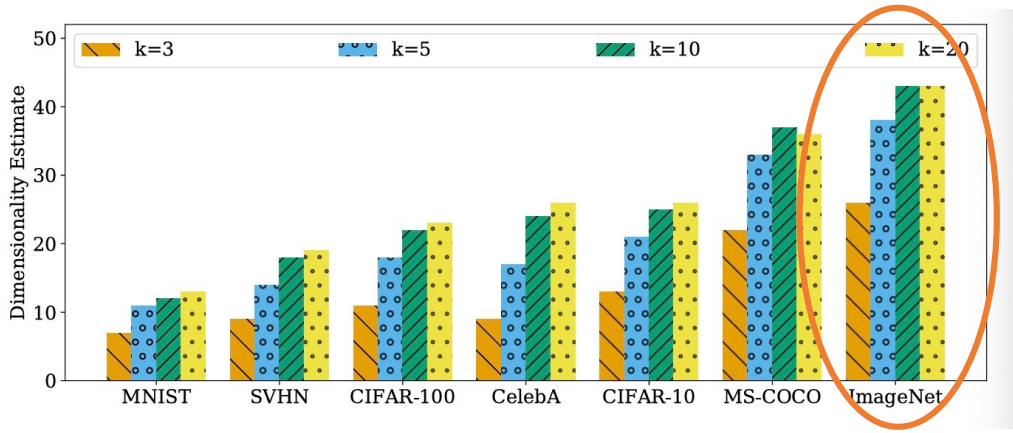
"Denoising diffusion probabilistic models are optimally adaptive to unknown low dimensionality," Z. Huang, Y. Wei, Y. Chen, arXiv:2410.18784, 2024

"Dimension-free convergence of diffusion models for approximate Gaussian mixtures," G. Li*, C. Cai*, Y. Wei, arXiv:2504.05300, 2025

d/ε iterations are too slow ...

ImageNet: $d = 150,528$ pixels per image, $n = 14$ million+ images

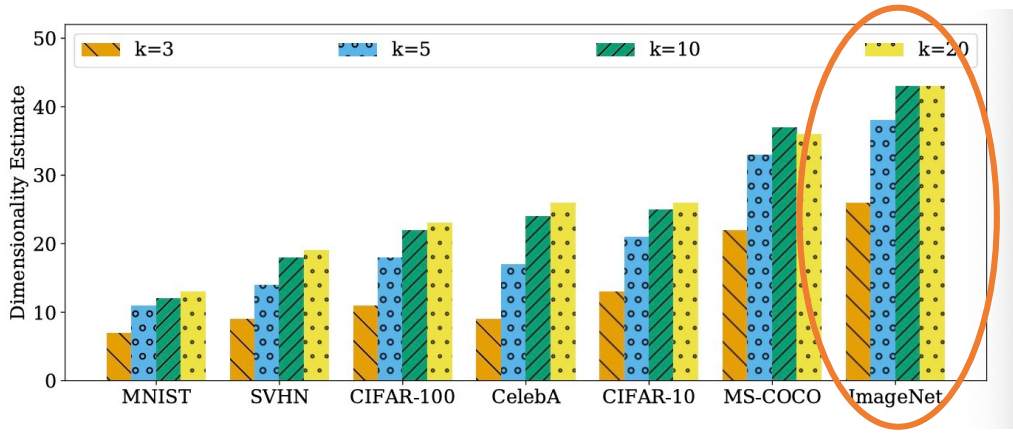
d/ε iterations are too slow . . .



ImageNet: $d = 150,528$ pixels per image, $n = 14$ million+ images

$k = 43$ intrinsic dimension (Pope et al. '21)

d/ε iterations are too slow ...



ImageNet: $d = 150,528$ pixels per image, $n = 14$ million+ images
$k = 43$ intrinsic dimension (Pope et al. '21)

In practice, DDIM/DDPM yield good samples in hundreds (or tens) of iterations ...

Can diffusion models adapt to intrinsic low dimensionality?

Intrinsic dimension

The target distribution has intrinsic dimension k if

$$\log N^{\text{cover}}(\mathcal{X}_{\text{data}}, \|\cdot\|_2, \varepsilon_0) \lesssim k \log\left(\frac{1}{\varepsilon_0}\right)$$

- k-dimensional linear subspaces
- low-dimensional manifolds
- ...

Main result: DDPM adapts to low dimensionality

Theorem (Huang, Wei, Chen'24)

DDPM sampler (its original form) yields $\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}(k/\varepsilon) \quad \textit{iterations}$$

— *concurrent work* Potaptchik et al.'24

Main result: DDPM adapts to low dimensionality

Theorem (Huang, Wei, Chen'24)

DDPM sampler (its original form) yields $\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}(k/\varepsilon) \quad \textit{iterations}$$

— *concurrent work* Potaptchik et al.'24

- Optimal dependence on k
- Li & Yan'24: k^4 dependence on the intrinisic dimension
- Azangulov, Deligiannidis, Rousseau'24: k^3 dependence on the intrinisic dimension
- $\widetilde{O}(k/\varepsilon)$ complexity in terms of TV distance Liang, Huang, Chen'25

Intuition: DDPM as an adaptively discretized SDE

Backward SDE:

$$\mathrm{d}Y_t = \big(Y_t + 2s_{T-t}(Y_t)\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

Intuition: DDPM as an adaptively discretized SDE

Backward SDE:

$$\mathrm{d}Y_t = \big(Y_t + 2s_{T-t}(Y_t)\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

equivalently, with $\mu_t(x) := \mathbb{E}[X_0 \mid X_t = x]$ and $\sigma_t^2 = 1 - e^{-2t}$

$$\mathrm{d}Y_t = \left((1 - \frac{2}{\sigma_{T-t}^2})Y_t + \frac{2\sqrt{1-\sigma_{T-t}^2}}{\sigma_{T-t}^2}\mu_{T-t}(Y_t)\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

Tweedie's formula:

$$\mu_{T-t}(Y_{T-t}) = \frac{1}{\sqrt{1-\sigma_{T-t}^2}}\left(Y_{T-t} + \sigma_t^2 s_{T-t}(Y_{T-t})\right)$$

Intuition: DDPM as an adaptively discretized SDE

Backward SDE:

$$\mathrm{d}Y_t = \big(Y_t + 2s_{T-t}(Y_t)\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

equivalently, with $\mu_t(x) := \mathbb{E}[X_0 \mid X_t = x]$ and $\sigma_t^2 = 1 - e^{-2t}$

$$\mathrm{d}Y_t = \left((1 - \frac{2}{\sigma_{T-t}^2})Y_t + \frac{2\sqrt{1-\sigma_{T-t}^2}}{\sigma_{T-t}^2}\mu_{T-t}(Y_t)\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

use itô's formula determine function f, st:

$$\mathrm{d}\big(f(t)Y_t\big) = 2\big(f(t)\big)^2\widehat{\mu}_{T-t_n}(Y_{t_n})\mathrm{d}t + \sqrt{2}f(t)\mathrm{d}B_t,$$

for $f(t) := e^{-(T-t)}/(1 - e^{-2(T-t)})$

Intuition: DDPM as an adaptively discretized SDE

Backward SDE:

$$\mathrm{d}Y_t = \big(Y_t + 2s_{T-t}(Y_t)\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

equivalently, with $\mu_t(x) := \mathbb{E}[X_0 \mid X_t = x]$ and $\sigma_t^2 = 1 - e^{-2t}$

$$\mathrm{d}Y_t = \left((1 - \frac{2}{\sigma_{T-t}^2})Y_t + \frac{2\sqrt{1-\sigma_{T-t}^2}}{\sigma_{T-t}^2}\mu_{T-t}(Y_t)\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

use itô's formula determine function f, st:

$$\mathrm{d}\big(f(t)Y_t\big) = 2\big(f(t)\big)^2\widehat{\mu}_{T-t_n}(Y_{t_n})\mathrm{d}t + \sqrt{2}f(t)\mathrm{d}B_t,$$

for $f(t) := e^{-(T-t)}/(1 - e^{-2(T-t)})$

— Solve analytically? DDPM sampler!

Can diffusion models adapt to other structures, e.g. Gaussian mixture models?

figure credit: Dall-E 3 from OpenAI

An incomplete list of prior art

Gaussian mixture models (Pearson'94)

$$X_0 \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma), \qquad \sum_{i=1}^{K} \pi_i = 1.$$

An incomplete list of prior art

Gaussian mixture models (Pearson'94)

$$X_0 \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma), \qquad \sum_{i=1}^{K} \pi_i = 1.$$

- Dasgupta'99
- Vempala & Wang'04
- Arora & Kannan'05
- Kalai et al.'10
- Moitra & Valiant'10
- Hsu & Kakade'13
- Diakonikolas et al.'18
- Hopkins & Li'18
- Ghosal & Van Der Vaart'01
- Dwivedi, Wainwright, et al.'20
- Chen and Qin'06
- Heinrich & Kahn'18
- Wu & Yang'20
- Doss et al.'23
- Saha & Guntuboyina'20
- Ashtiani et al.'18
- Shah et al.'23
- Liang et al.'24
- Wu et al.'24
- Chidambaram et al.'24
- Chen et al.'24
- Gatmiry et al.'24
- Wang et al.'24

Main result: DDPM for GMMs

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \qquad t = 1, \cdots, T$$

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)\Big), \qquad t = T, \cdots, 1$$

Theorem (Li*, Cai*, Wei'25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, assume each component of GMM satisfies $\|\mu_k/\sigma\|_2 \le T^{c_R}$. The output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

Main result: DDPM for GMMs

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \qquad t = 1, \cdots, T$$
$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)\Big), \qquad t = T, \cdots, 1$$

Theorem (Li*, Cai*, Wei'25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, assume each component of GMM satisfies $\|\mu_k/\sigma\|_2 \le T^{c_R}$. The output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

- To yield an ε-accurate distribution, it requires $\widetilde{O}(1/\varepsilon)$ iterations

Main result: DDPM for GMMs

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \qquad t = 1, \cdots, T$$
$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)\Big), \qquad t = T, \cdots, 1$$

Theorem (Li*, Cai*, Wei'25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, assume each component of GMM satisfies $\|\mu_k/\sigma\|_2 \le T^{c_R}$. The output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

- To yield an ε-accurate distribution, it requires $\widetilde{O}(1/\varepsilon)$ iterations
- Robust to score estimation error

Main result: DDPM for GMMs

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \qquad t = 1, \cdots, T$$
$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)\Big), \qquad t = T, \cdots, 1$$

Theorem (Li*, Cai*, Wei'25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, assume each component of GMM satisfies $\|\mu_k/\sigma\|_2 \le T^{c_R}$. The output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

- To yield an ε-accurate distribution, it requires $\widetilde{O}(1/\varepsilon)$ iterations
- Robust to score estimation error
- Discrete time analysis $\Rightarrow$ a TV distance control

Main result: DDPM for GMMs

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d), \qquad t = 1, \cdots, T$$

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)\Big), \qquad t = T, \cdots, 1$$

Theorem (Li*, Cai*, Wei'25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, assume each component of GMM satisfies $\|\mu_k/\sigma\|_2 \le T^{c_R}$. The output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

Even in ultra-high-dimensions, diffusion models are highly effective in sampling GMMs!

Concluding remarks

- Sharp convergence theory for DDIM
- Diffusion models are adaptive to unknown distribution structure: *low-dim manifolds, GMMs*; *more to come*

— Thanks for your attention!

Papers:

"Towards non-asymptotic convergence for diffusion-based generative models," G. Li, Y. Wei, Y. Chen, Y. Chi, ICLR 2024.

"A sharp convergence theory for the probability flow ODEs of diffusion models," G. Li, Y. Wei, Y. Chi, Y. Chen, 2024.

"Denoising diffusion probabilistic models are optimally adaptive to unknown low dimensionality," Z Huang, Y Wei, Y Chen, 2024.

"Dimension-free convergence of diffusion models for approximate Gaussian mixtures," G. Li*, C. Cai*, Y. Wei, 2025

Assumptions: learning rates

$$X_0 \sim p_{\text{data}}, \quad X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t}\mathcal{N}(0, I_d)$$

- **learning rates:** for some consts $c_0, c_1 > 0$,

$$1 - \alpha_1 = \frac{1}{T^{c_0}}$$

$$1 - \alpha_t = \underbrace{\frac{c_1 \log T}{T} \min\left\{(1-\alpha_1)\left(1 + \frac{c_1 \log T}{T}\right)^t, 1\right\}}_{\text{2 phases: exp growth} \to \text{flat}}$$

Assumptions: score estimates $\{s_t(\cdot)\}$

- *ℓ_2 score estimation error:* $s_t^\star(X) := \nabla \log p_{X_t}(X)$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim p_{X_t}} \left[\|s_t(X) - s_t^\star(X)\|_2^2\right] \le \varepsilon_{\text{score}}^2$$

  Needed for both stochastic and deterministic samplers

Assumptions: score estimates $\{s_t(\cdot)\}$

- *ℓ_2 score estimation error:* $s_t^\star(X) := \nabla \log p_{X_t}(X)$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim p_{X_t}} \left[\|s_t(X) - s_t^\star(X)\|_2^2\right] \le \varepsilon_{\text{score}}^2$$

  Needed for both stochastic and deterministic samplers

- *Jacobian estimation error:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim p_{X_t}} \left[\left\|\frac{\partial s_t}{\partial X}(X) - \frac{\partial s_t^\star}{\partial X}(X)\right\|\right] \le \varepsilon_{\text{Jacobi}}$$

  Needed for deterministic samplers (counterexamples exist)

Proof strategies I

Proof Road-Map

Key quantities: $\mu_t(x) := \mathbb{E}[X_0 \mid X_t = x] \quad \Sigma_t(x) := \mathrm{Cov}[X_0 \mid X_t = x]$

discretization error: • $\mathbb{E}\left[\left\|\mu_{T-t}(Y_t) - \mu_{T-s}(Y_s)\right\|_2^2\right] = \mathbb{E}\left[\mathrm{Tr}(\Sigma_{T-s}(Y_s)\right] - \mathbb{E}\left[\mathrm{Tr}(\Sigma_{T-t}(Y_t)\right]$

Itô's formula
connection to Jacobian of score function
Jacobian is connected to $\Sigma_t(x)$

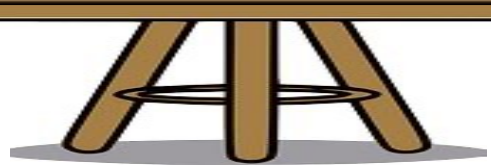
$$\Rightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[\left\|\mu_{T-t}(Y_t) - \mu_{T-s}(Y_s)\right\|_2^2\right] = c_t\, \mathbb{E}\left[\|\Sigma_{T-t}(Y_t)\|_{\mathrm{F}}^2\right]$$

Stochastic localization

$$\Rightarrow \quad \mathbb{E}\left[\|\Sigma_{T-t}(Y_t)\|_{\mathrm{F}}^2\right] = -c_t^{-1}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[\mathrm{Tr}\left(\Sigma_{T-t}(Y_t)\right)\right]$$

• $\mathbb{E}\left[\mathrm{Tr}\left(\Sigma_{T-t}(Y_t)\right)\right] \lesssim k \log k$

Integrate to obtain an error control that only depends on k !



Proof strategies II

Proof Road-Map

along the forward process: $X_t \sim \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\sqrt{\overline{\alpha}_t}\mu_k, I_d\big)$

$$s_t^\star(x) := \nabla \log p_{X_t}(x) = -\sum_{k=1}^{K} \pi_k^{(t)}(x)\big(x - \sqrt{\overline{\alpha}_t}\mu_k\big) = -x + \sum_{k=1}^{K} \pi_k^{(t)}(x)\sqrt{\overline{\alpha}_t}\mu_k$$

Jacobian matrix of score function:

$$J_t(x) := \frac{\partial s_t^\star(x)}{\partial x} = -I_d + \overline{\alpha}_t \sum_{k=1}^{K} \pi_k^{(t)} \left(\mu_k - \sum_{i=1}^{K} \pi_i^{(t)}\mu_i\right)\left(\mu_k - \sum_{i=1}^{K} \pi_i^{(t)}\mu_i\right)^\top$$

Key property:

$$\mathrm{trace}\big(I_d + J_t(x)\big) \lesssim \log(KT)$$