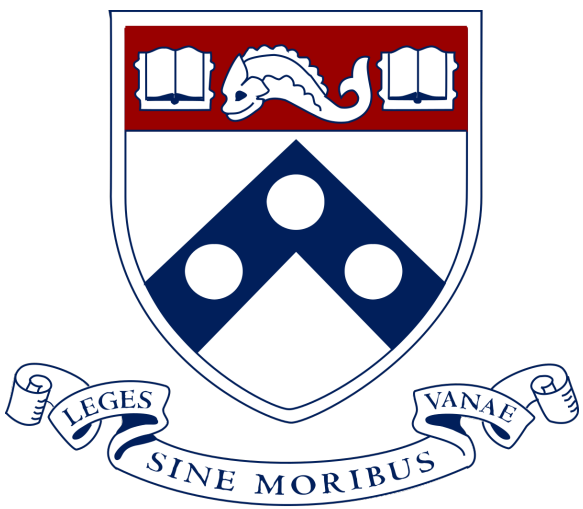
To Intrinsic Dimension and Beyond: Efficient Sampling in Diffusion Models

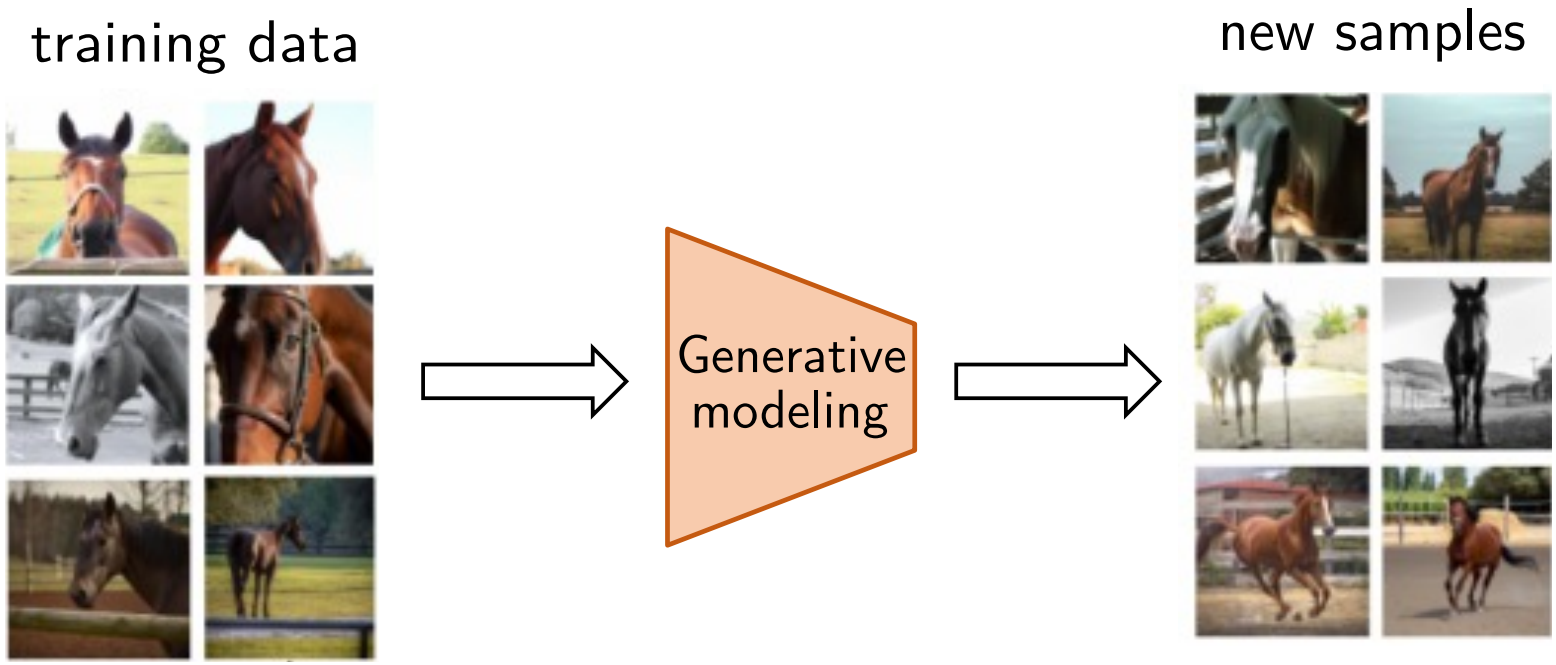


Yuting Wei

Statistics & Data Science, Wharton
University of Pennsylvania

SNAB 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

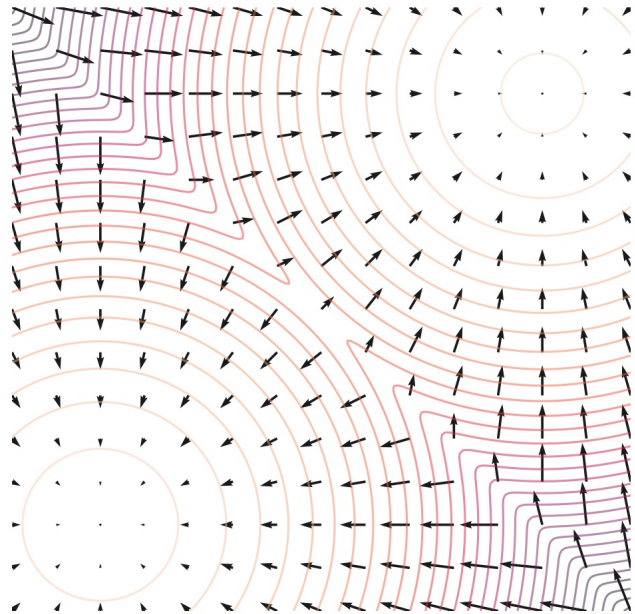
The **(Stein) score function** of a distribution $p(x)$ is defined as

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

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

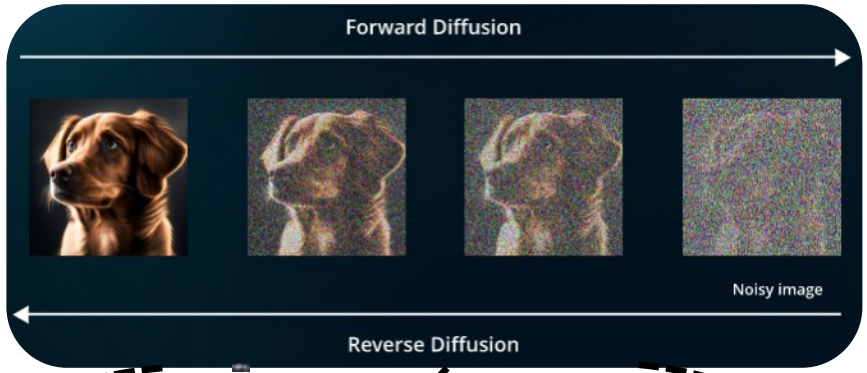
The score function points towards regions of higher probability.

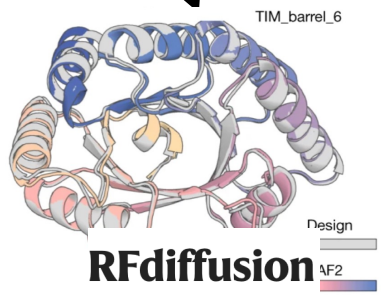
Score-based diffusion models

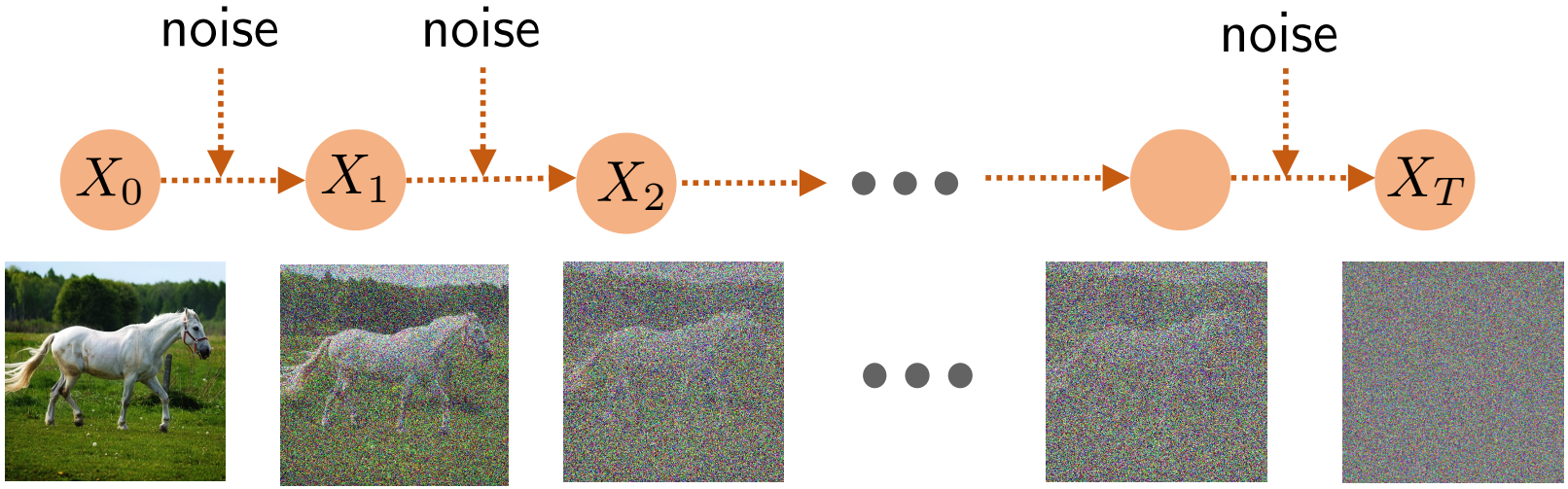
Inspired by nonequilibrium thermodynamics

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

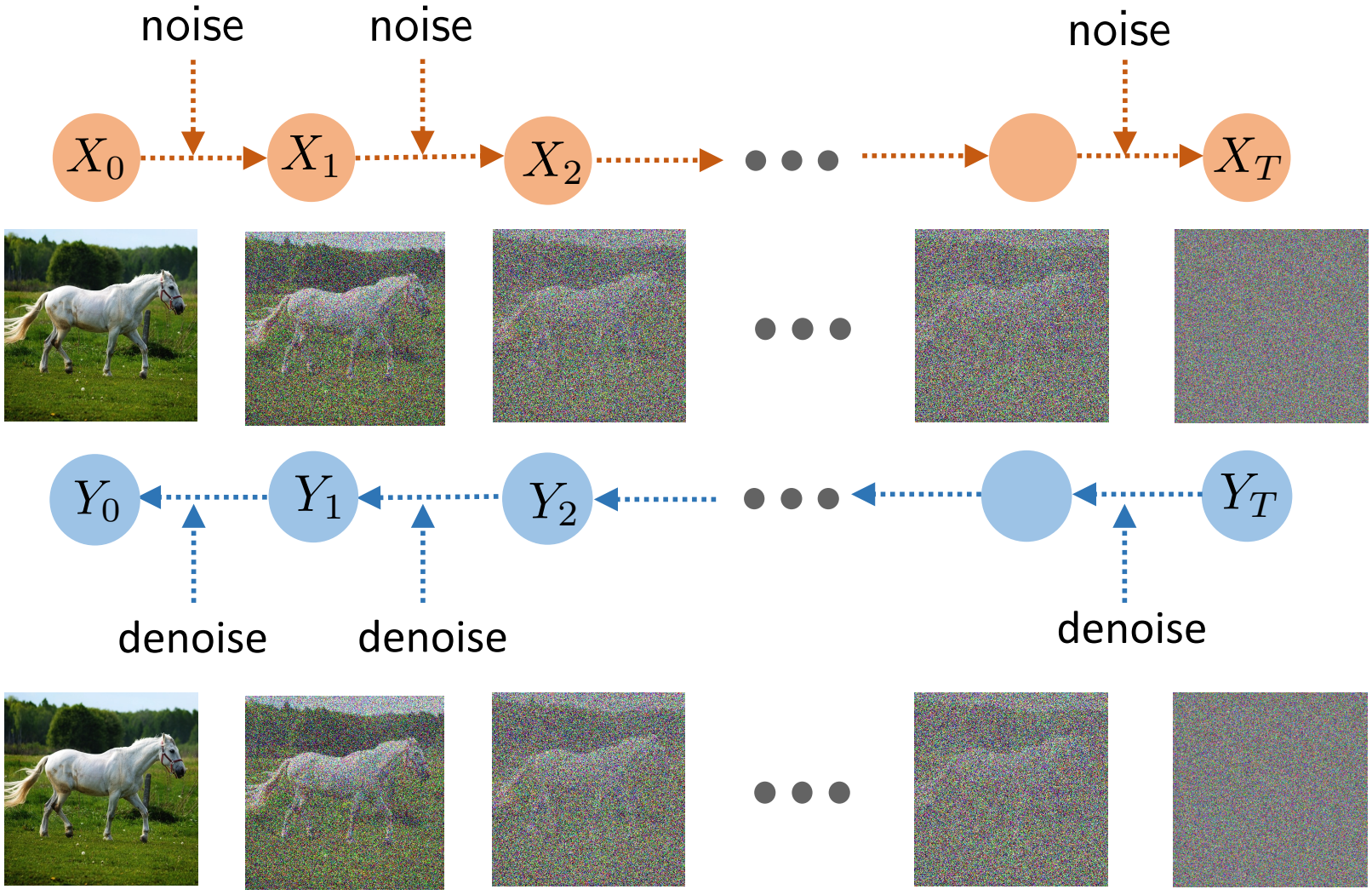
Diffusion models







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



- **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 $t = T, \cdots, 1$?

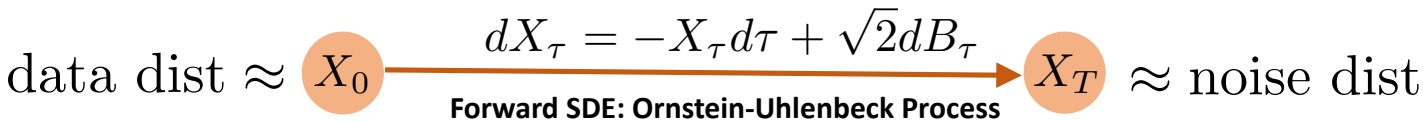
It is feasible as long as one knows the score function $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$

—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 $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$



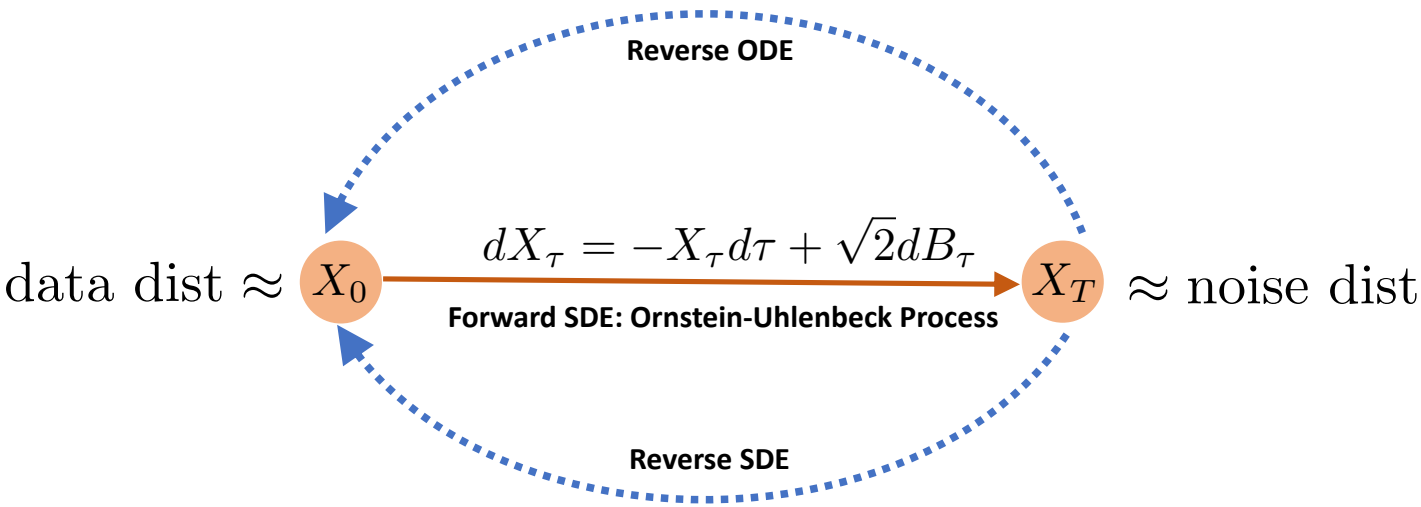
—Anderson'82; Haussmann and Pardoux'86; Song et el.'20...

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 $\underbrace{\nabla \log p_{X_t}(x)}_{\text{w.r.t. } X}$

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

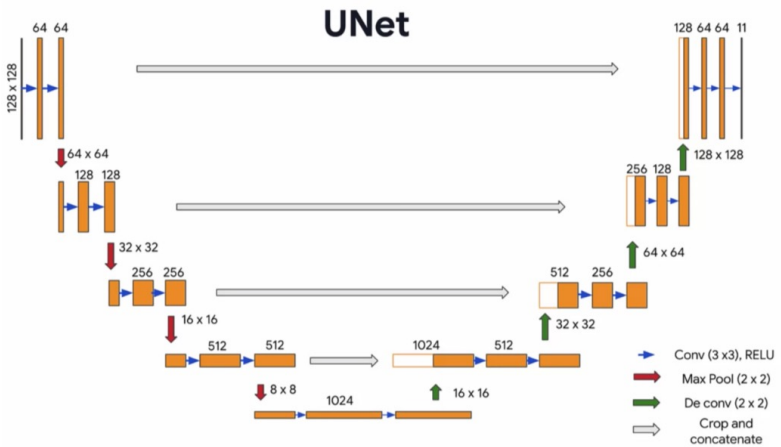


$$dY_\tau = \left(Y_\tau + 2\nabla \log p_{X_{T-\tau}}(Y_\tau)\right) d\tau + \sqrt{2} dB_\tau$$

—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

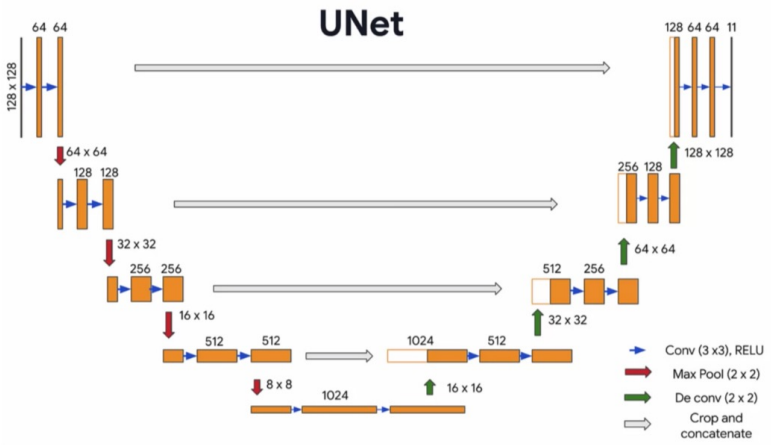


$\Longrightarrow$ learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

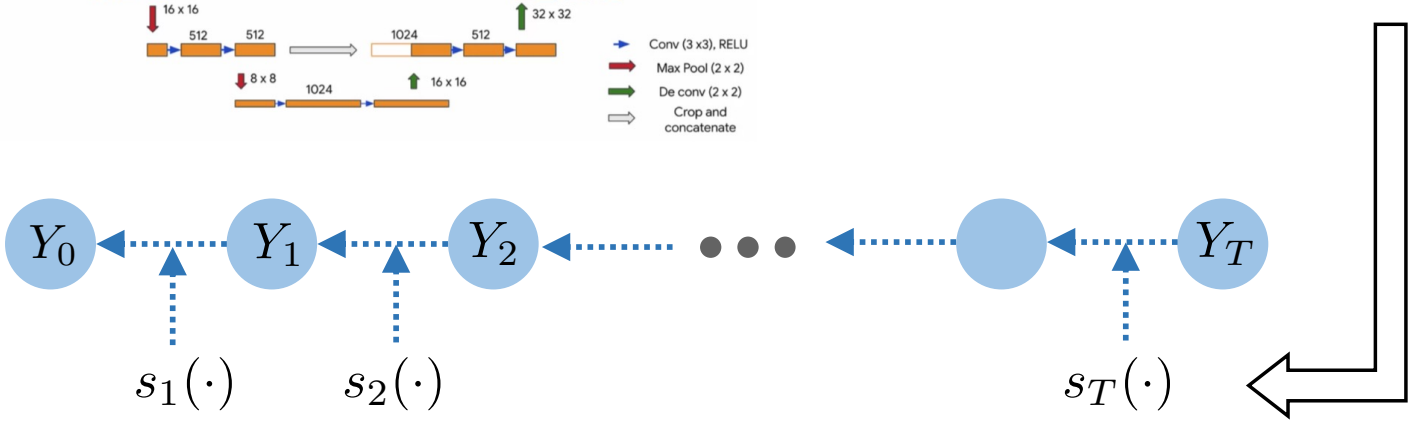
1. **score learning/matching:** learn estimates $s_t(\cdot)$ for $\nabla \log p_{X_t}(\cdot)$: often achieved by neural networks

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
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

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

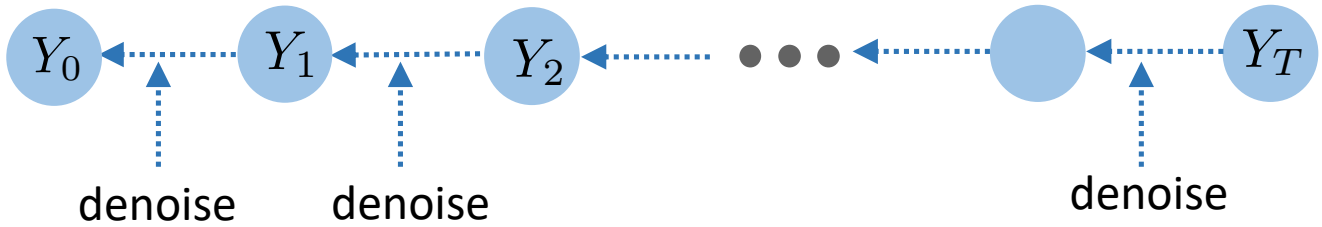
Tweedie's formula (Hyvarinen'05; Vincent'11):

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

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

Denoising diffusion probabilitic models (DDPMs)

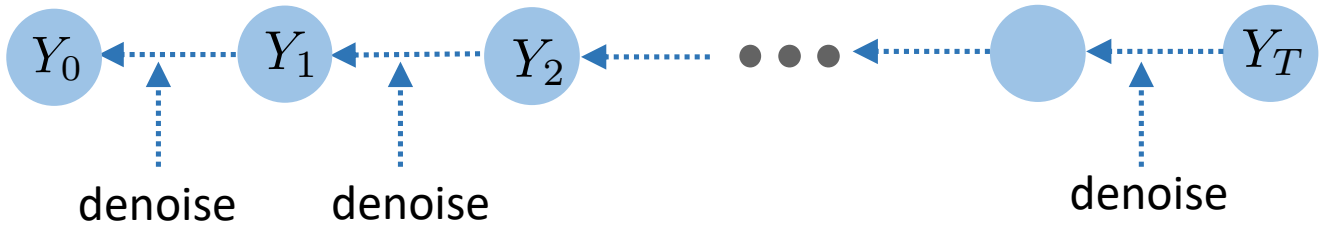


— *Ho, Jain, Abbeel '20*

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

DDPM

Denoising diffusion probabilitic models (DDPMs)



— *Ho, Jain, Abbeel '20*

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

DDPM

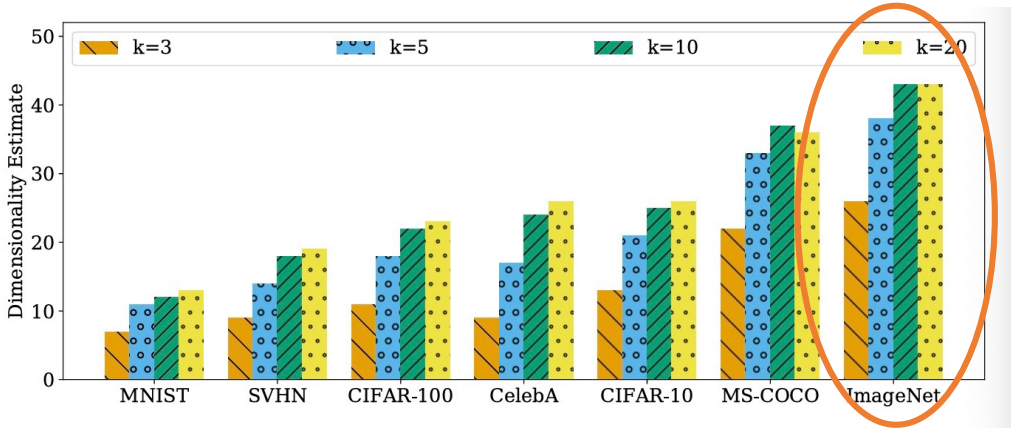
$$Y_T \sim \mathcal{N}(0, I_d)$$

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

Prior theory: linear scaling with d

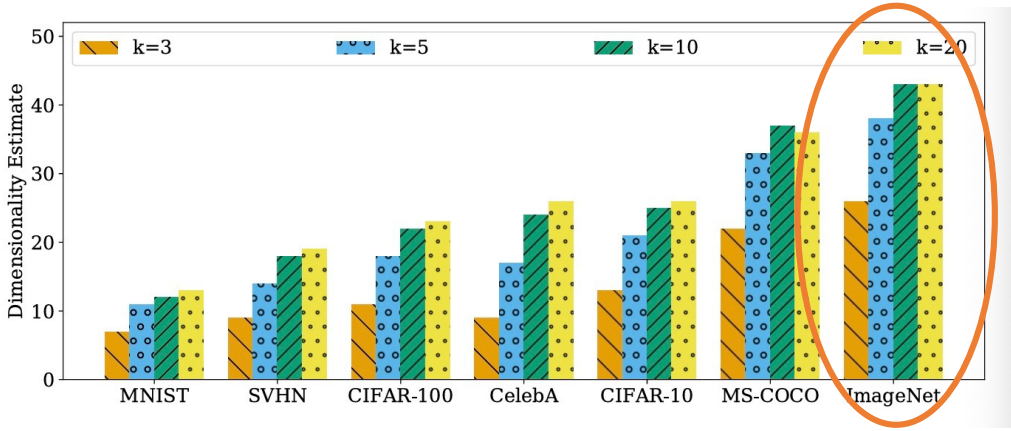
ImageNet: $d = 150,528$ pixels per image, $n = 14$ million+ images

Prior theory: linear scaling with d



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

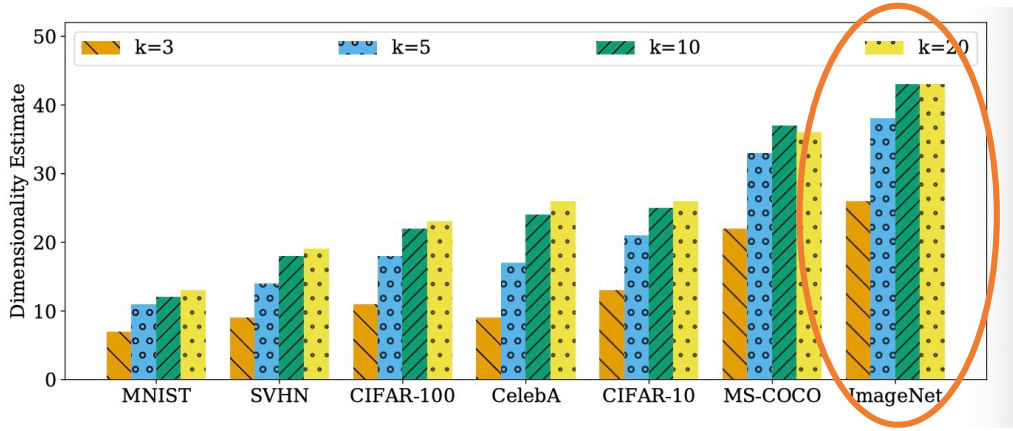
Prior theory: linear scaling with d



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

— **Prior theory:** # iterations $\widetilde{O}(d/\varepsilon)$ to generate accurate samples

Prior theory: linear scaling with d



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

— **Prior theory:** # iterations $\widetilde{O}(d/\varepsilon)$ to generate accurate samples

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

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

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

Main result: DDPM adapts to low dimensionality

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

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

- assume access to accurate score estimate, i.e, $\varepsilon_{\mathsf{score}} \le \varepsilon/2$

Theorem (Huang, Wei, Chen'24)

DDPM sampler (its original form) yields $\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \le \varepsilon$ *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
- Jin & Wang'15
- Cai, Ma, Zhang'19
- Tian, Weng, Xia, Feng'24
- 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 $\Sigma = \sigma^2 I_d$, the output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{1}{T} + \varepsilon_{\mathsf{score}}.$$

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 $\Sigma = \sigma^2 I_d$, the output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{1}{T} + \varepsilon_{\mathsf{score}}.$$

- 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 $\Sigma = \sigma^2 I_d$, the output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{1}{T} + \varepsilon_{\mathsf{score}}.$$

- 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 $\Sigma = \sigma^2 I_d$, the output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{1}{T} + \varepsilon_{\mathsf{score}}.$$

- 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 $\Sigma = \sigma^2 I_d$, the output of DDPM sampler obeys

$$\mathsf{TV}(X_0, Y_0) \lesssim \frac{1}{T} + \varepsilon_{\mathsf{score}}.$$

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

Take-home message

Diffusion models can automatically adapt to unknown distributional structures: *e.g. low-dim manifolds, GMMs*

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

— Thank you for your attention!

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 $c_0, c_1 > 0$,

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

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

Proof strategy

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I), \quad Y_{t-1} = \underbrace{\frac{1}{\sqrt{1-\beta_t}} Y_t + \frac{\beta_t}{2\sqrt{1-\beta_t}} s_t(Y_t)}_{=: \Phi_t(Y_t)}$$

$\mathsf{TV}\left(p_{X_t}, p_{Y_t}\right) \approx 0$

Proof strategy

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I), \quad Y_{t-1} = \underbrace{\frac{1}{\sqrt{1-\beta_t}} Y_t + \frac{\beta_t}{2\sqrt{1-\beta_t}} s_t(Y_t)}_{=: \Phi_t(Y_t)}$$

$$\mathrm{TV}\left(p_{X_t}, p_{Y_t}\right) \approx 0 \quad \Longleftarrow \quad \frac{p_{Y_t}(y_t)}{p_{X_t}(y_t)} \approx 1 \quad \forall y_t \in \mathcal{E}_t \text{ (some "typical" set)}$$

$t-1$ $\quad$ t $\quad$ T

$$\frac{p_{Y_{t-1}}(\Phi_t(y_t))}{p_{X_{t-1}}(\Phi_t(y_t))} = \underbrace{\frac{p_{Y_{t-1}}(\Phi_t(y_t))}{p_{Y_t}(y_t)}}_{\text{relation btw } Y_t \text{ \& } Y_{t-1}} \left(\underbrace{\frac{p_{X_{t-1}}(\Phi_t(y_t))}{p_{X_t}(y_t)}}_{\text{relation btw } X_t \text{ \& } X_{t-1}} \right)^{-1} \frac{p_{Y_t}(y_t)}{p_{X_t}(y_t)}$$

$$\frac{p_{\Phi_t(Y_t)}\left(\Phi_t(y_t)\right)}{p_{Y_t}(y_t)} = \det\left(\frac{\partial \Phi_t}{\partial y_t}\right)^{-1}$$

some concentration bounds

Comparison w/ prior theory of GMMs

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

Comparison w/ prior theory of GMMs

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

- Liang, Shi, Song, Zhou'24: $\underbrace{\text{provides } \widetilde{O}(d/\varepsilon^2) \text{ complexity bound}}_{\text{show no adaptation phenomenon}}$

Comparison w/ prior theory of GMMs

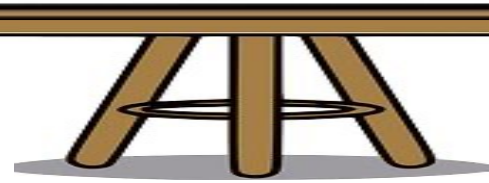
$$\text{(our theory)} \quad \mathsf{TV}(X_0, Y_0) \lesssim \frac{\log^2(KT)\log^2 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}.$$

- Liang, Shi, Song, Zhou'24: $\underbrace{\text{provides } \widetilde{O}(d/\varepsilon^2) \text{ complexity bound}}_{\text{show no adaptation phenomenon}}$
- Chen et al.'24; Gatmiry et al.'24: focus on score estimation using piecewise polynomial regression + existing DDPM convergence theory

Proof Road-Map

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

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



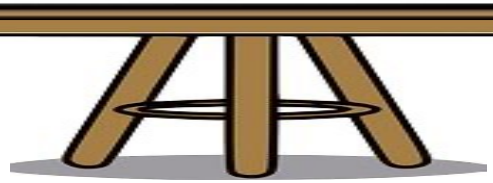
Proof Road-Map

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:

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



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:

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