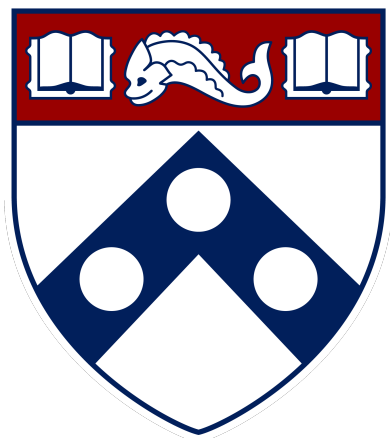
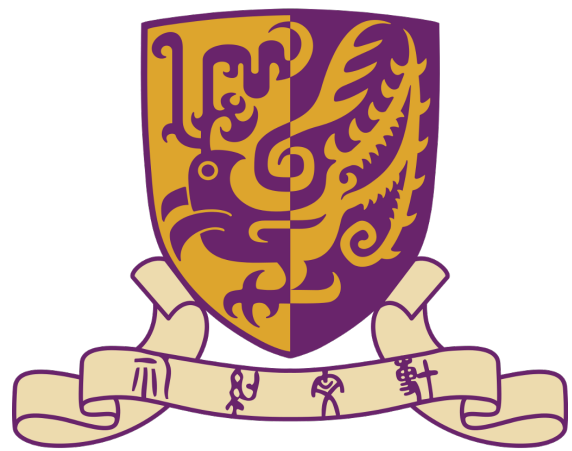
Theoretical Foundations of Diffusion Models



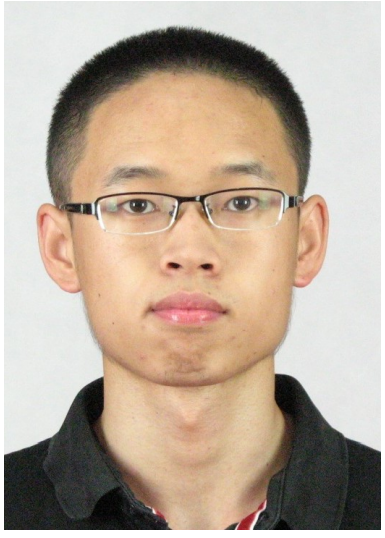


Yuxin Chen, Gen Li, Yuting Wei

UPenn & CUHK

Instructors

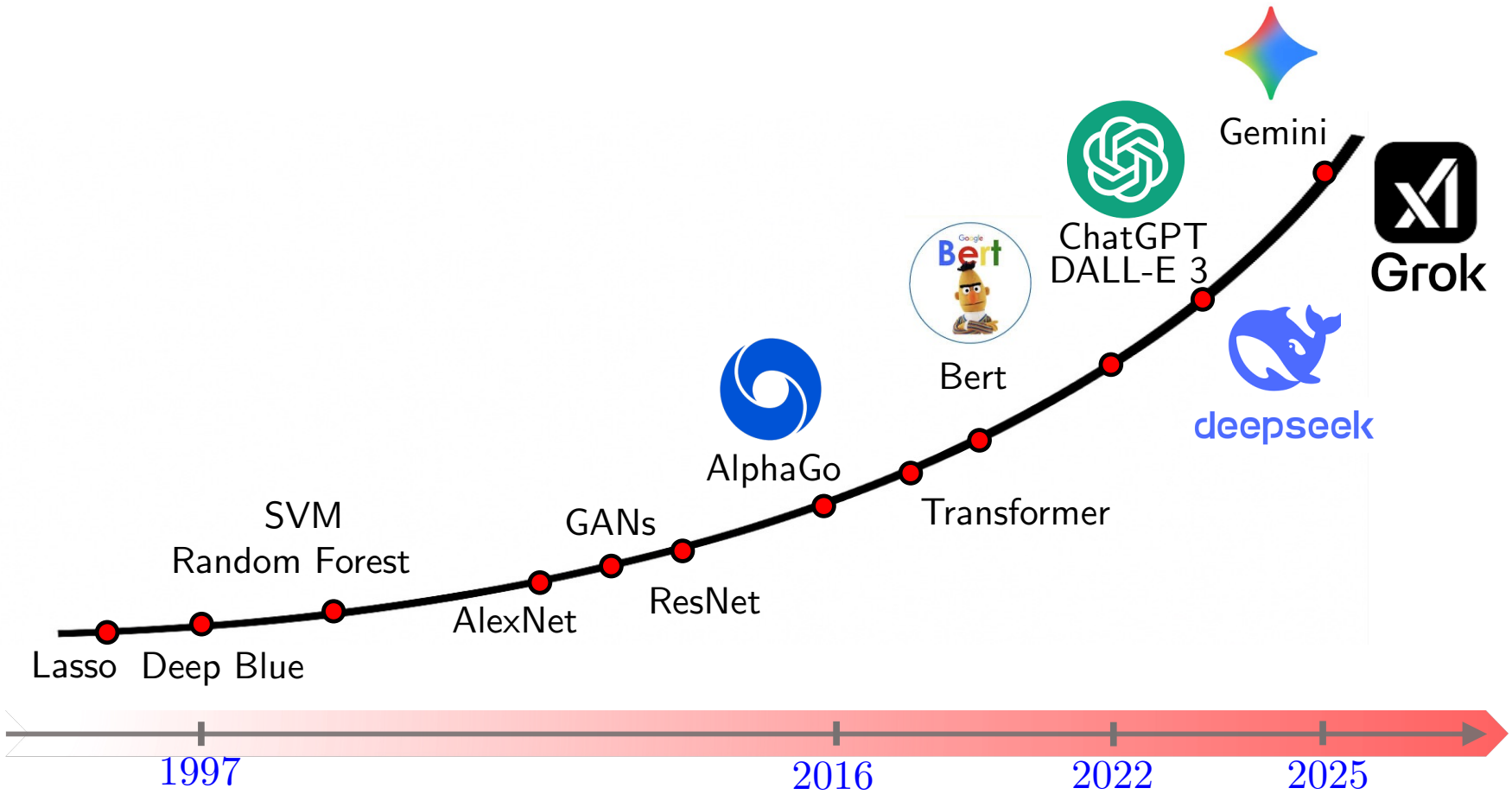
Yuxin Chen
UPenn



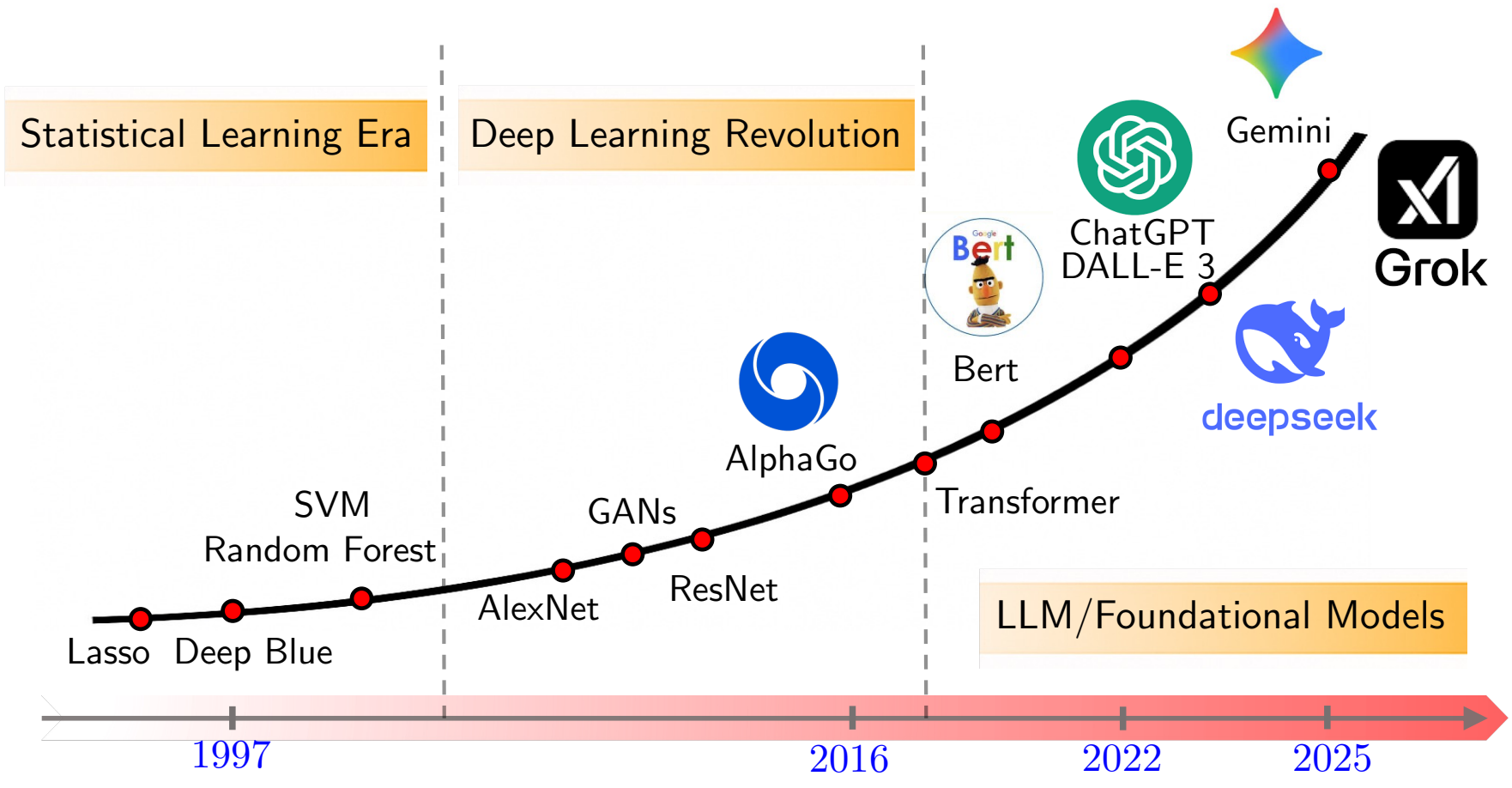
Gen Li
CUHK

Yuting Wei
UPenn

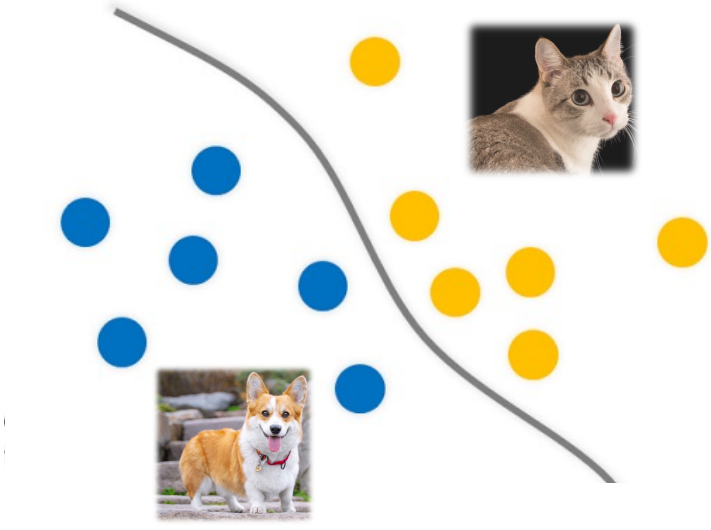
Evolution of AI models

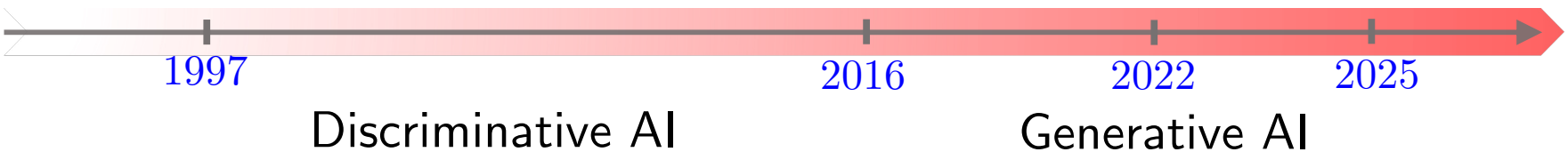


Evolution of AI models



Evolution of AI models





Diffusion models hold great promises

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

NeurIPS 2021, cited 10023

Diffusion models hold great promises

Diffusion Models Beat GANs on Image Synthesis

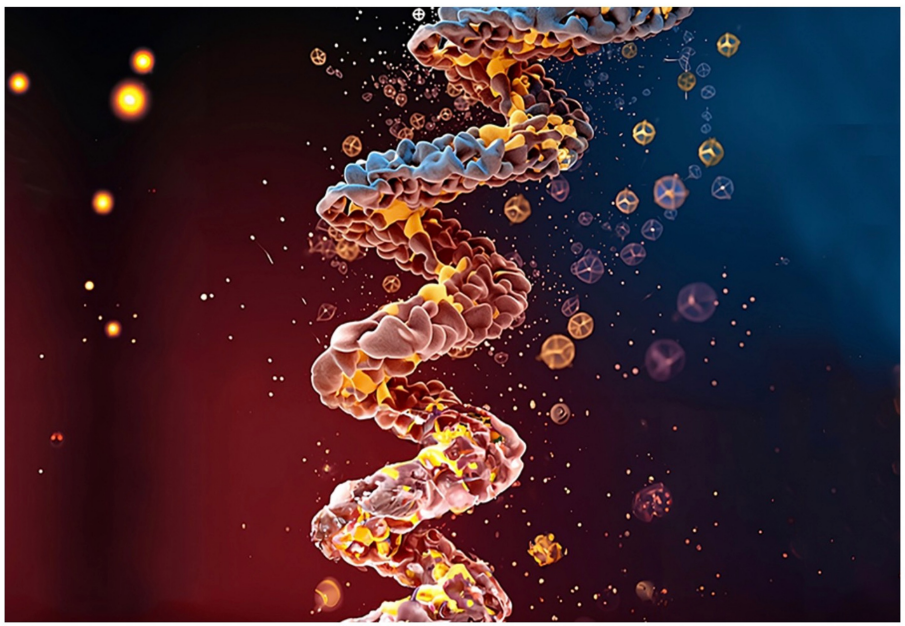
Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Generative AI imagines new protein structures

"FrameDiff" is a computational tool that uses generative AI to craft new protein structures, with the aim of accelerating drug development and improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



Diffusion models hold great promises

Diffusion Models Be

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

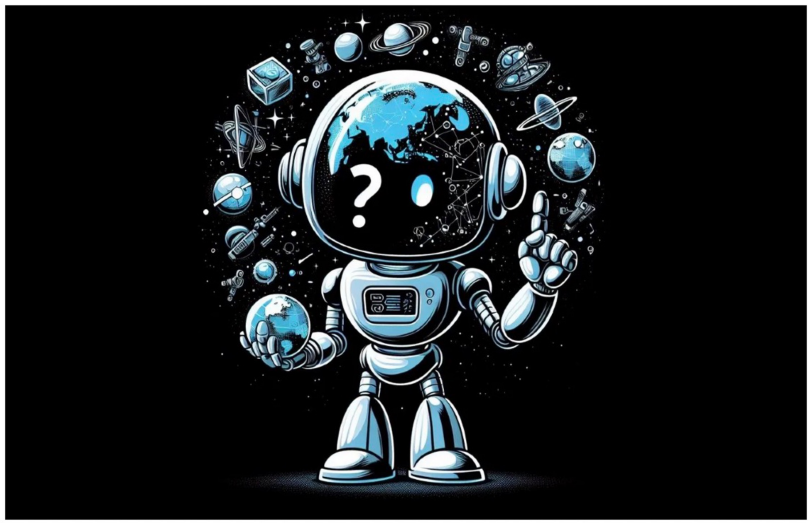
Diffusion models are now turbocharging reinforcement learning systems

By **Ben Dickson** - March 4, 2024

Generative AI imagine

"FrameDiff" is a computational to protein structures, with the aim o improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



Diffusion models hold great promises

Diffusion Models Be

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Diffusion models are now turbocharging reinforcement learning systems

By **Ben Dickson** - March 4, 2024

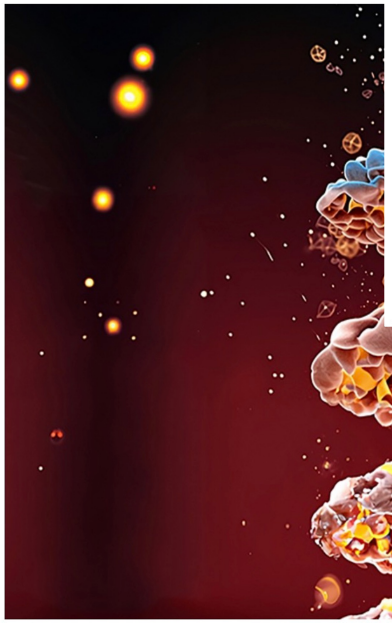
Like 75

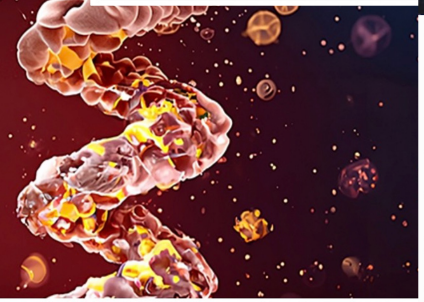
Facebook Twitter

Generative AI imagine

"FrameDiff" is a computational tc protein structures, with the aim o improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023





ScienceAdvances

HOME > SCIENCE ADVANCES > VOL. 10, NO. 13 > GENERATIVE EMULATION OF WEATHER FORECAST ENSEMBLES WITH DIFFUSION MODELS

RESEARCH ARTICLE | ATMOSPHERIC SCIENCE

Generative emulation of weather forecast ensembles with diffusion models

LIZAO LI, ROBERT CARVER, IGNACIO LOPEZ-GOMEZ, FEI SHA, AND JOHN ANDERSON Authors Info & Affiliations

SCIENCE ADVANCES • 29 Mar 2024 • Vol 10, Issue 13 • DOI: 10.1126/sciadv.adk4489

Stable Diffusion Based Solution for Medical Imaging

By NETSOL Technologies, on October 23, 2023

In this blog post, we'll deeply dive into the fascinating world of stable diffusion-based solutions and explore how they work alongside generative AI to take medical imaging to the next level.

INNOVATION > AI

Microsoft Team Uses Diffusion Model For Materials Science

By John Werner, Contributor. I am an MIT Senior Fellow & Lecturer, 5x-... Follow Author

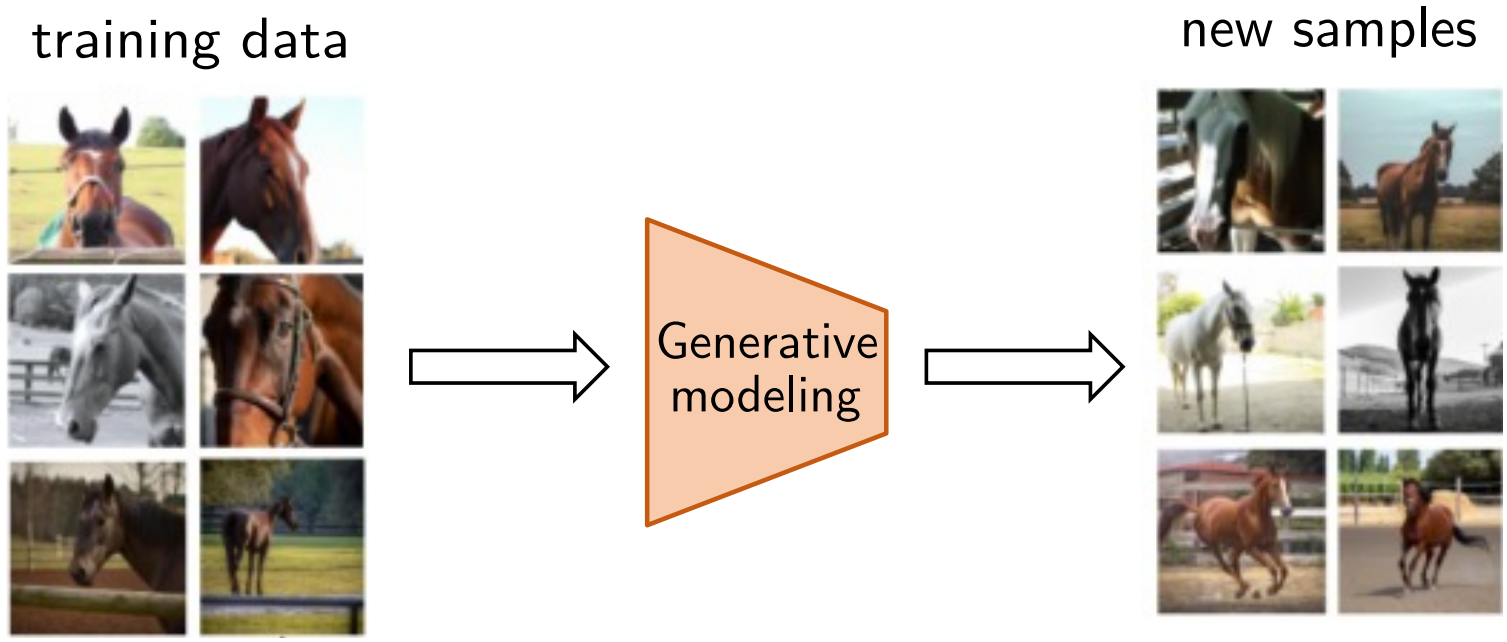
Published Jan 21, 2025, 03:40am EST

Generative modeling

training data

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

Generative modeling



- 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

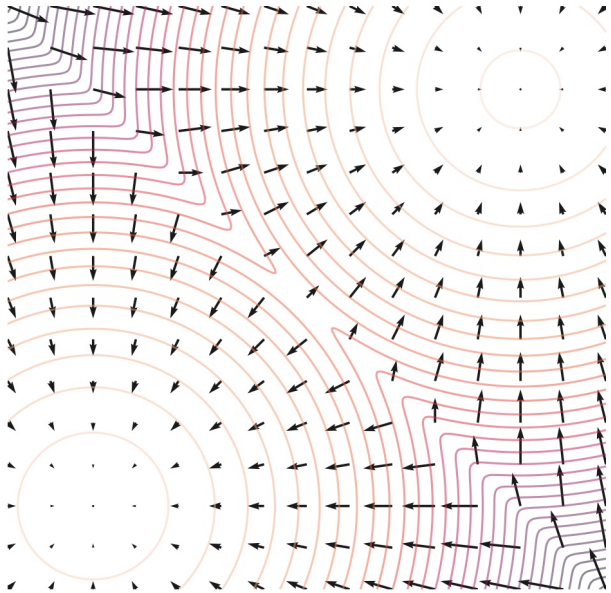
The **(Stein) score function** of a distribution $p(x)$ is defined as

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

Note that

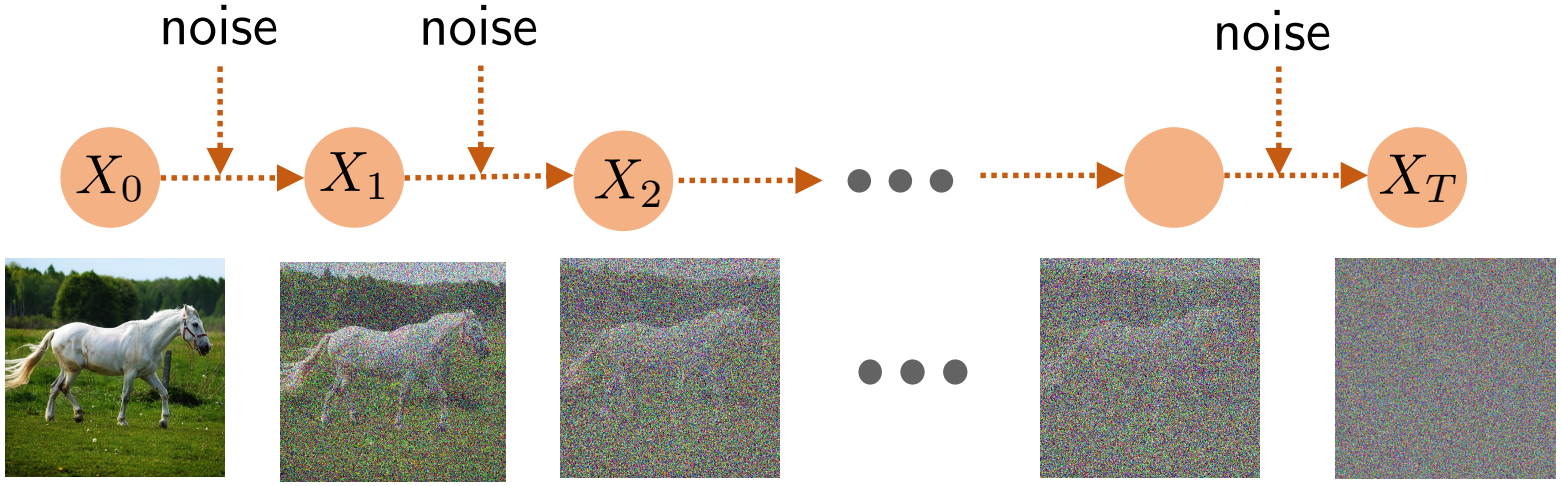
$$\begin{aligned}\nabla \log p(x \mid \theta) &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= -\nabla_x f_\theta(x)\end{aligned}$$

getting rid of the annoying Z_θ!

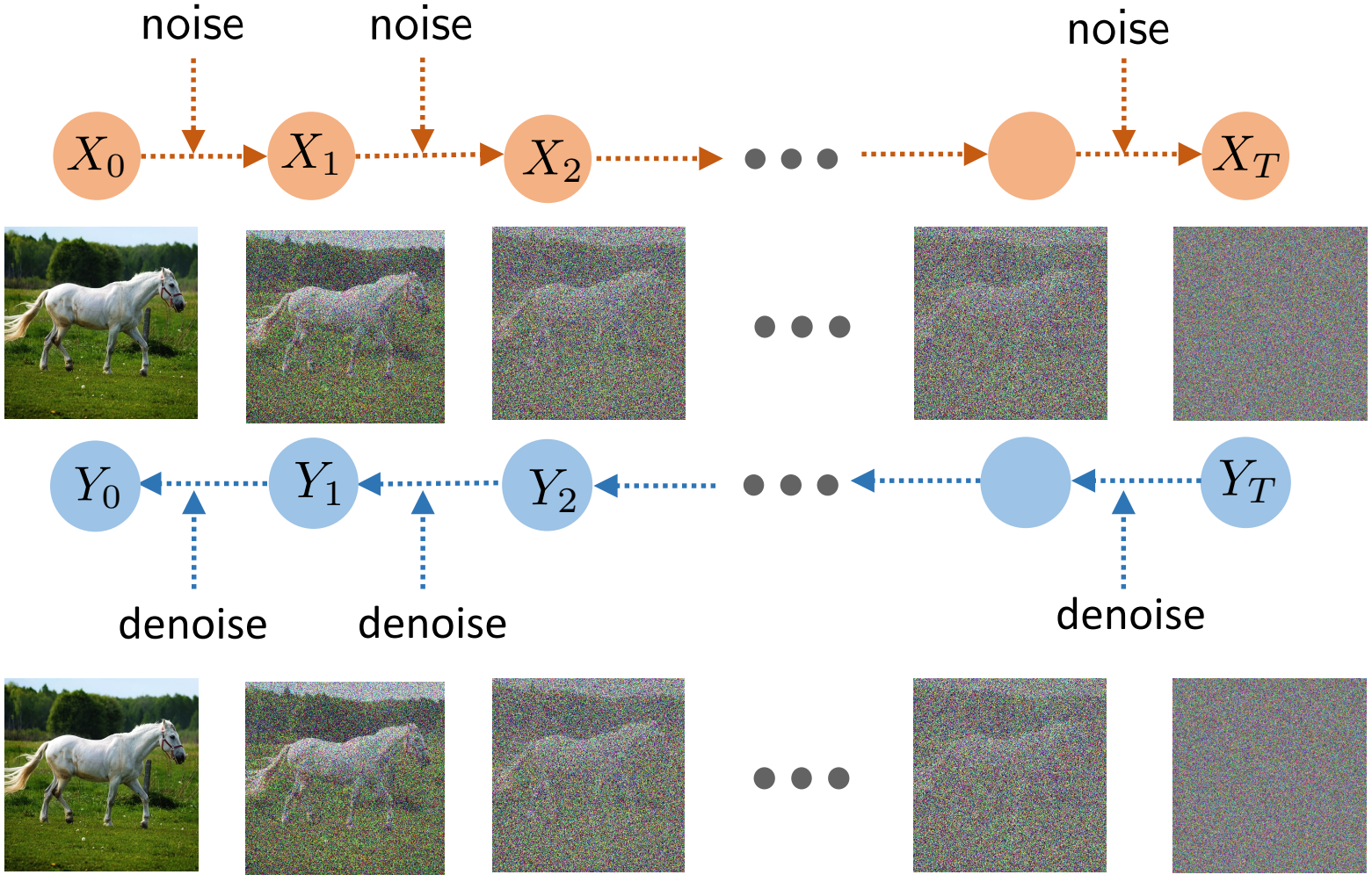


Score function of Gaussian mixtures

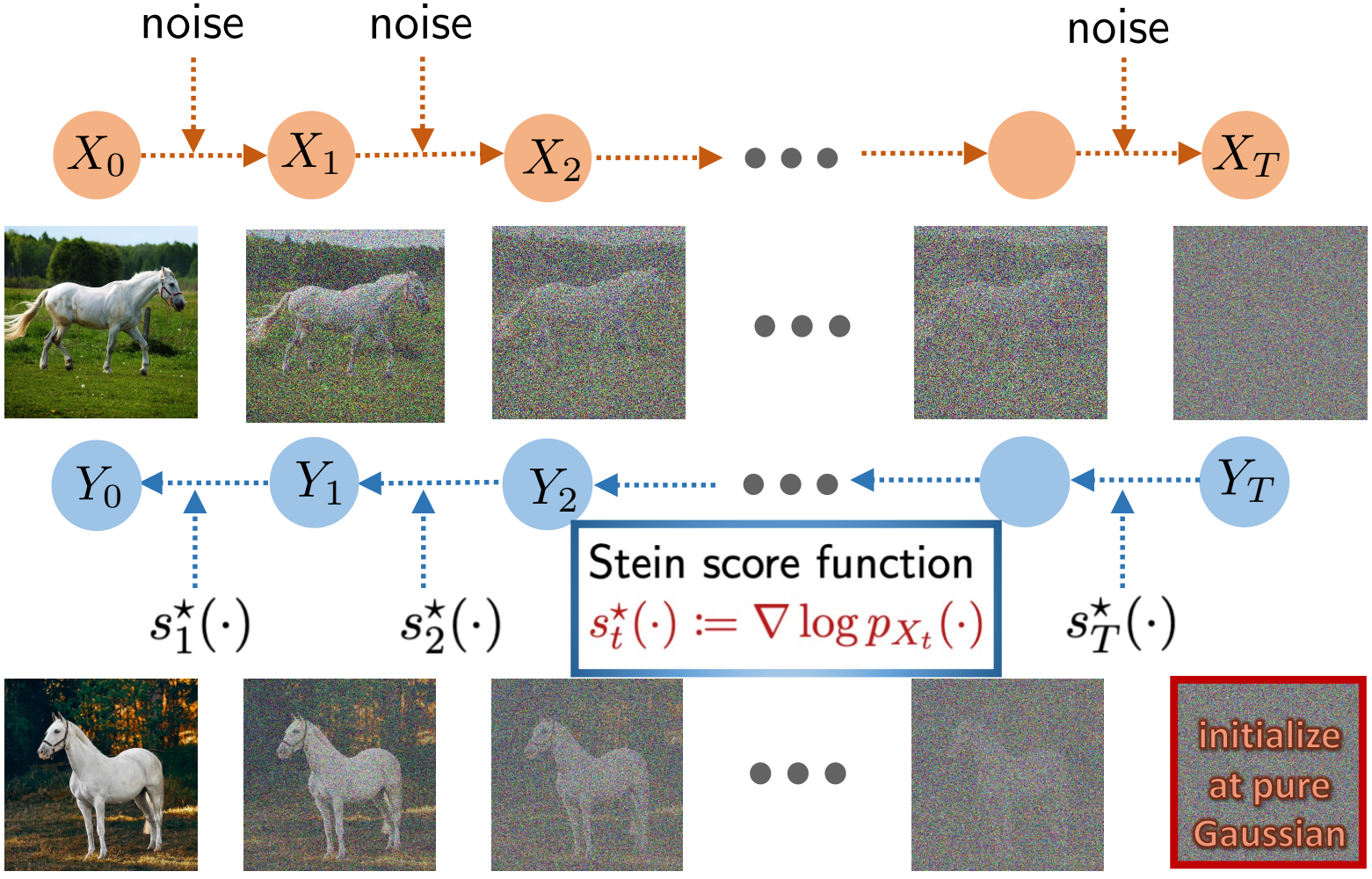
The score function points towards regions of higher probability.



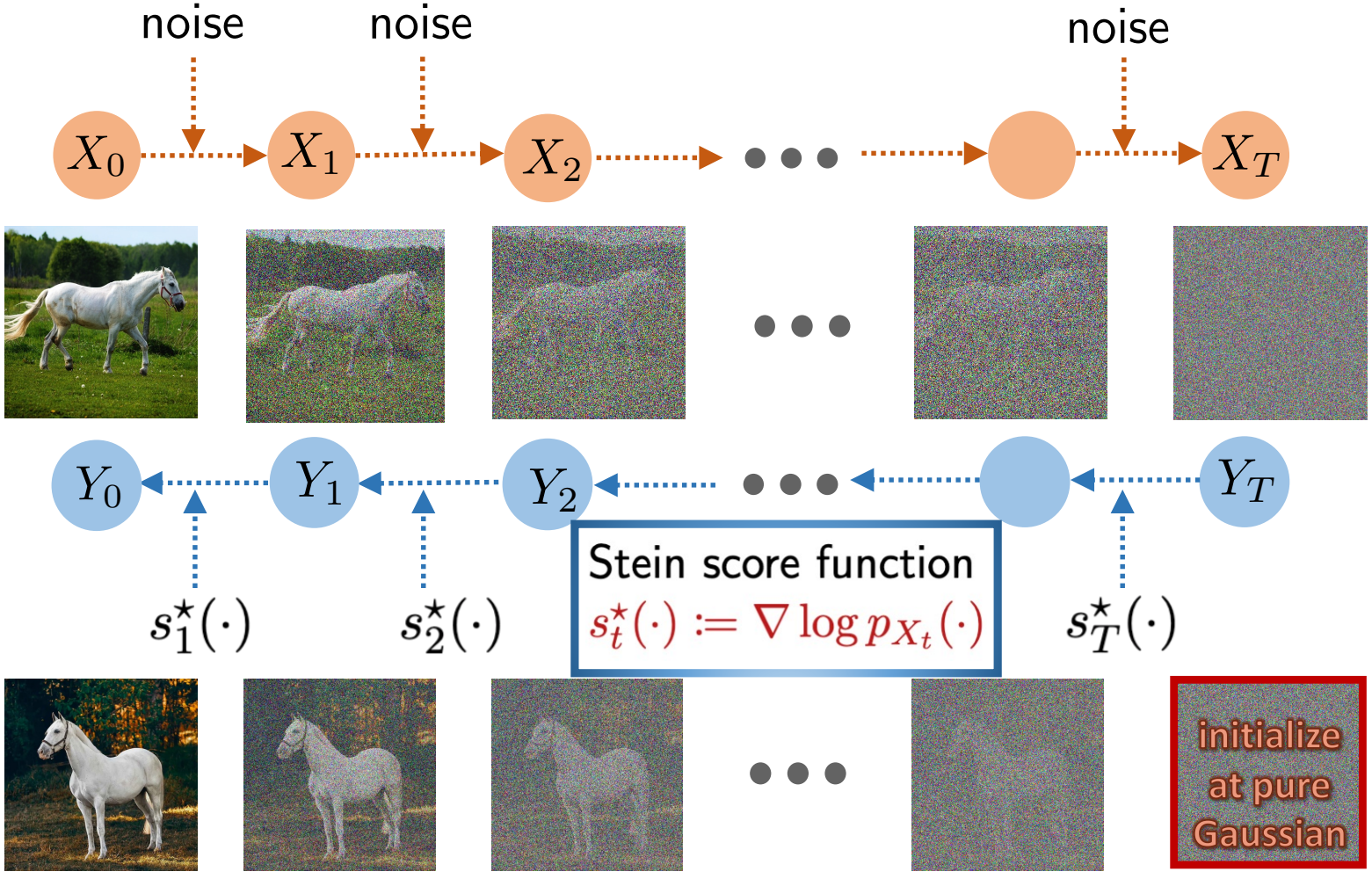
- **forward process:** diffuse data into noise



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



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



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

Goal: $Y_t \overset{\mathrm{d}}{\approx} X_t, \quad t = T, \cdots, 1$

Score is all you need

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

Score is all you need

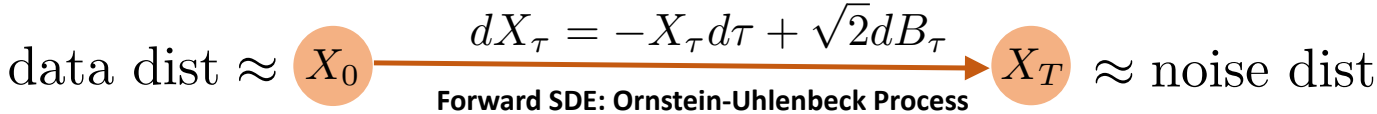
How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

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

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

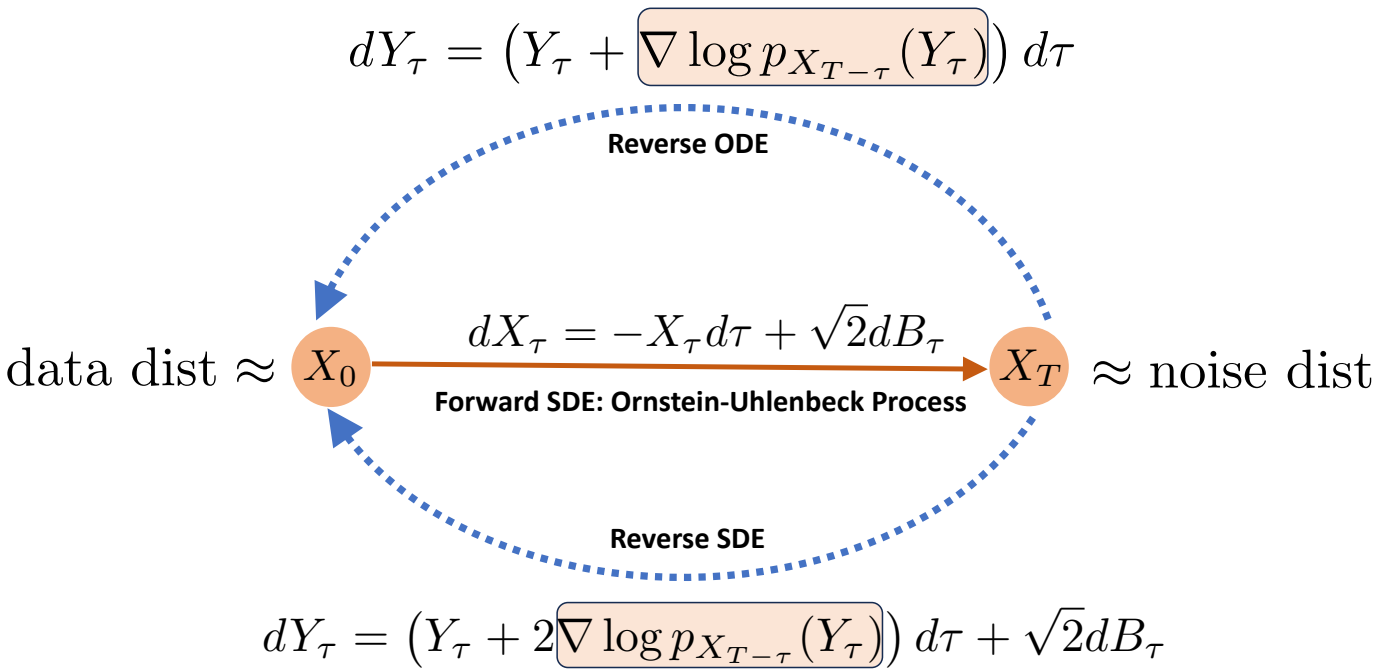


—*Anderson'82; Haussmann and Pardoux'86; Song et el.'20...*

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...*

Towards mathematical theory for diffusion models

- hard to develop full-fledged **end-to-end** theory

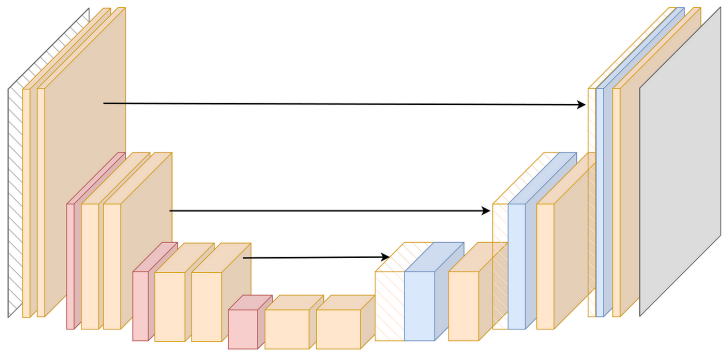
Towards mathematical theory for diffusion models

- hard to develop full-fledged **end-to-end** theory
- "divide-and-conquer": score learning $\overset{\text{decouple}}{\leftarrow \not\rightarrow}$ sampling

A divide-and-conquer approach

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

denoising network

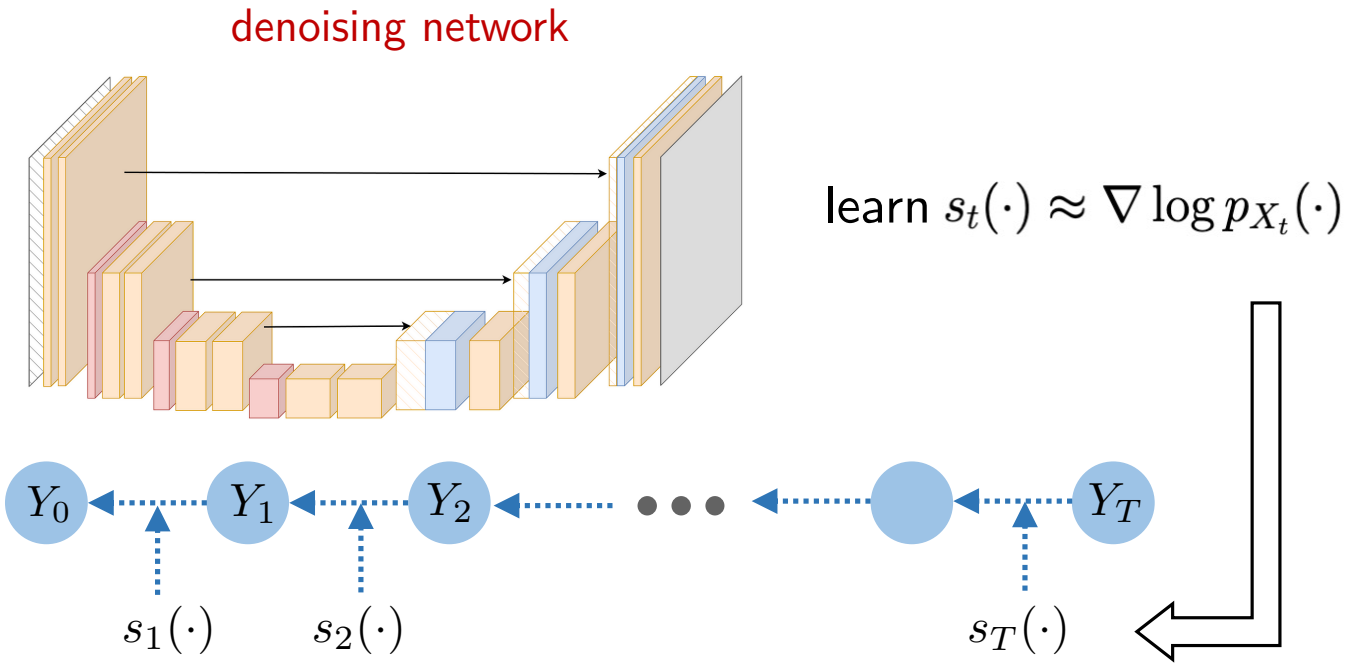


learn $s_t(\cdot) \approx \nabla \log p_{X_t}(\cdot)$

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

A divide-and-conquer approach

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



1. **score learning/matching:** learn estimates $s_t(\cdot)$ for $\nabla \log p_{X_t}(\cdot)$
2. **data generation:** sampling w/ 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

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

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 $W \sim \mathcal{N}(0, I_d), X_0 \sim p_{\mathsf{data}}$.

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):

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

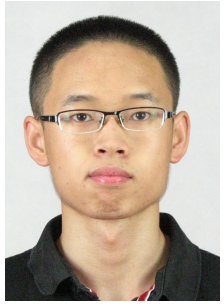
where the expectation is taken over $W \sim \mathcal{N}(0, I_d), X_0 \sim p_{\text{data}}$.

- 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

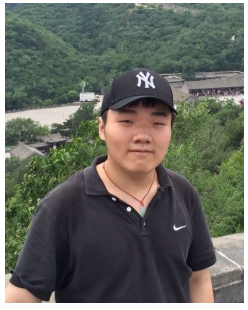
Agenda:

1. non-asymptotic convergence theory
2. adaptation to (unknown) low dimensionality
3. acceleration via higher-order approximation
4. provable benefits of diffusion guidance
5. diffusion models for inverse problems
6. discrete diffusion (or diffusion language models)

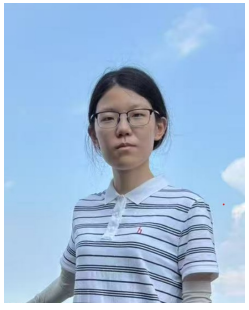
Part 1: nonasymptotic convergence theory



Gen Li
CUHK



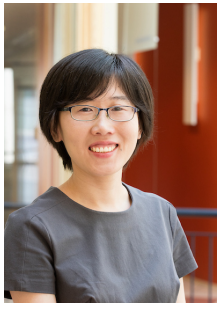
Yuling Yan
DE Shaw



Yuchen Jiao
CUHK

Yuchen Zhou
UIUC

Yuxin Chen
UPenn



Yuejie Chi
Yale

Yuting Wei
UPenn

Two mainstream approaches

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

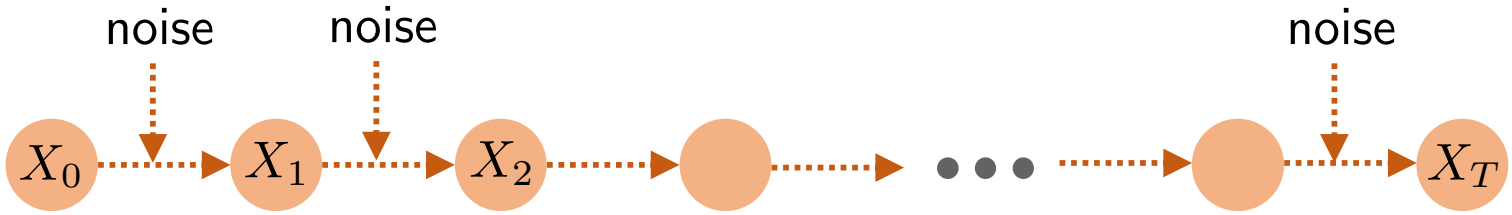
Ajay Jain
UC Berkeley
ajayj@berkeley.edu

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

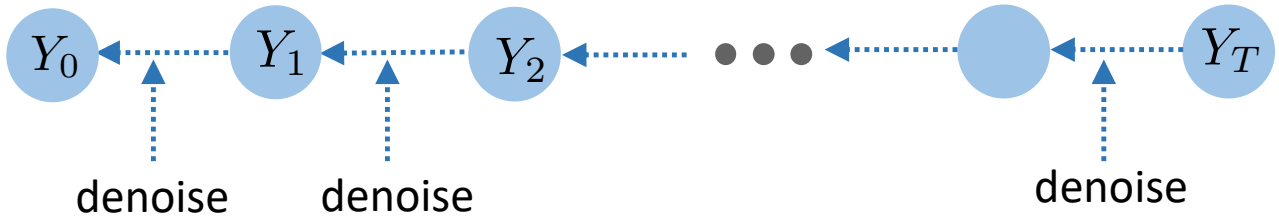


forward process: $X_0 \sim p_{\text{data}}$ (target distribution)

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

- $\beta_t := 1 - \alpha_t$ controls variance of injected noise

DDPM vs. DDIM

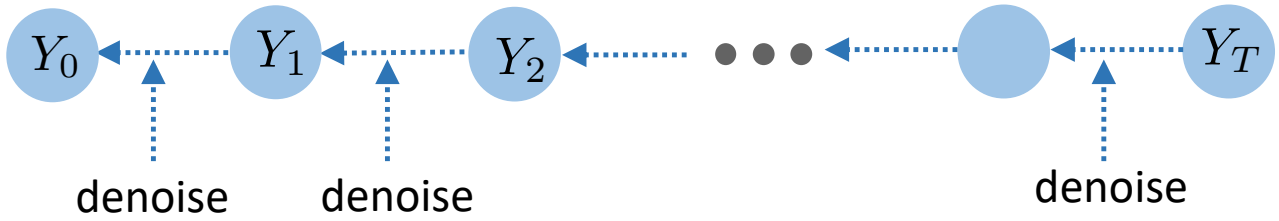


— *Ho, Jain, Abbeel '20*

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

DDPM

DDPM vs. DDIM



— *Ho, Jain, Abbeel '20*

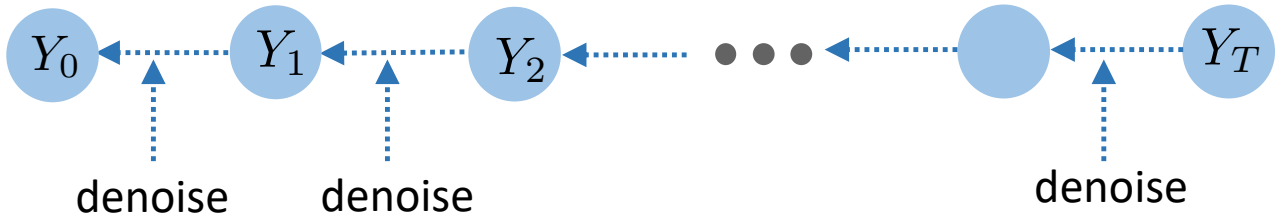
1. A stochastic sampler: **denoising diffusion probabilistic models**

DDPM

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

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

DDPM vs. DDIM



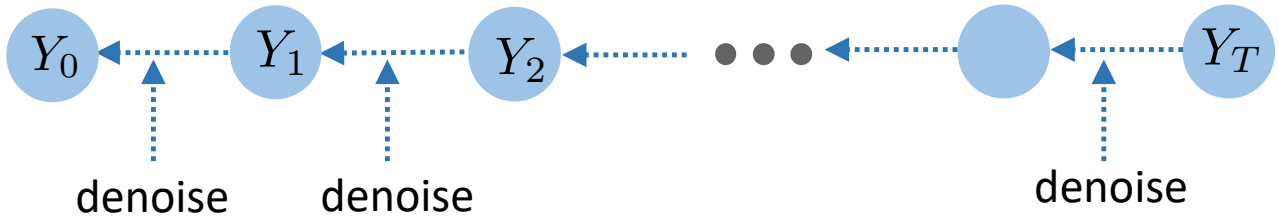
— *Song, Meng, Ermon '20*

— *Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole '20*

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

DDIM; or probability flow ODE

DDPM vs. DDIM



— *Song, Meng, Ermon '20*

— *Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole '20*

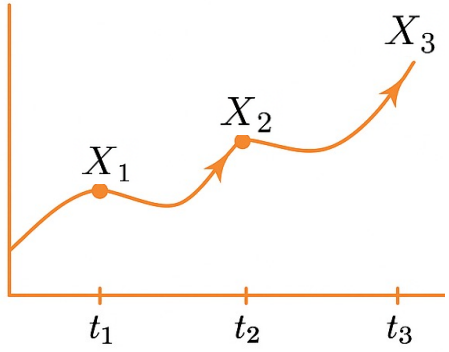
2. A <u>deterministic</u> sampler: **denoising diffusion implicit models**

DDIM; or probability flow ODE

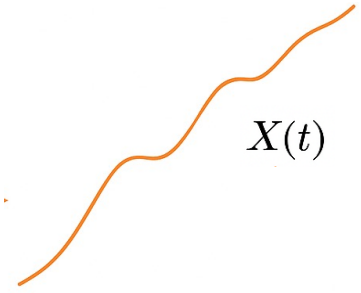
$$Y_T \sim \mathcal{N}(0, I_d)$$

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

Interpretations from lens of SDE/ODE

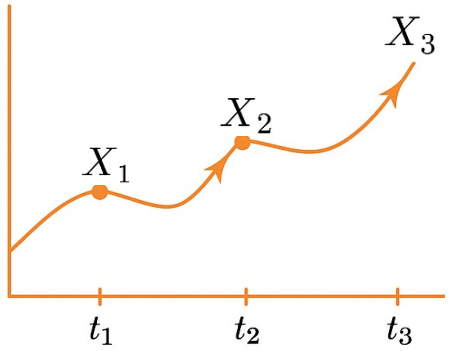


discrete-time

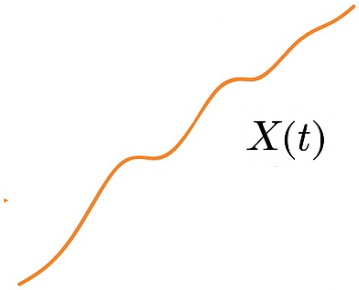


continuous-time

Interpretations from lens of SDE/ODE



discrete-time



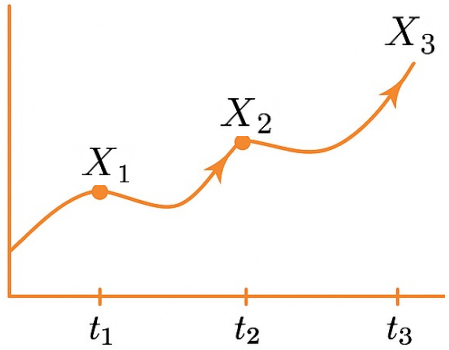
continuous-time

forward process

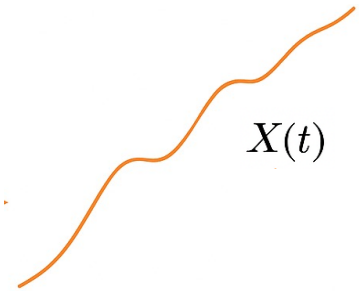
$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I_d)$$

$$\Longrightarrow \quad \mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t \quad \text{(SDE)}$$

Interpretations from lens of SDE/ODE



discrete-time



continuous-time

forward process

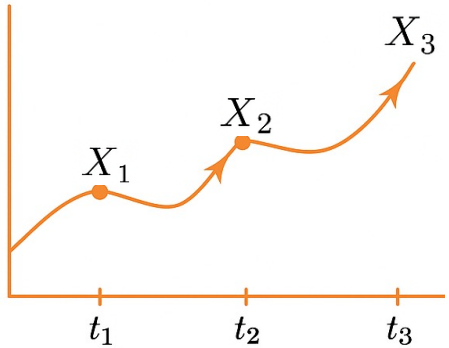
$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I_d)$$

$$\Longrightarrow \quad \mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t \quad \text{(SDE)}$$

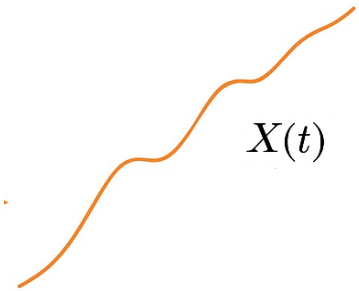
- $\exists$ reverse-time SDE w/ same path distribution (Anderson '82)

$$\mathrm{d}Y_t = \left(Y_t + 2\boldsymbol{s}^\star_{T-t}(Y_t)\right)\beta(T-t)\mathrm{d}t + \sqrt{2\beta(T-t)}\,\mathrm{d}W_t$$

Interpretations from lens of SDE/ODE



discrete-time



continuous-time

forward process

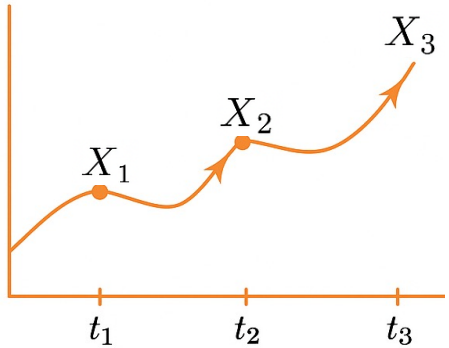
$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I_d)$$

$$\Longrightarrow \quad \mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t \quad \text{(SDE)}$$

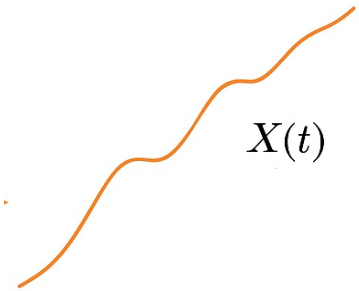
- $\exists$ reverse-time SDE w/ same path distribution (Anderson '82)

$$\xrightarrow{\text{time discretization}} \text{DDPM}$$

Interpretations from lens of SDE/ODE



discrete-time



continuous-time

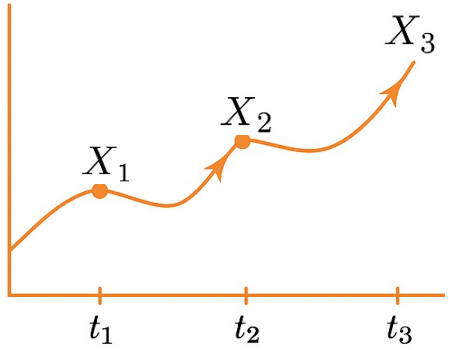
forward process

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I_d)$$
$$\Longrightarrow \quad \mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t \quad \text{(SDE)}$$

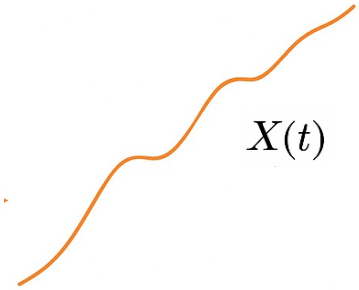
- $\exists$ reverse-time ODE w/ same *marginal* dist (Song et al. '20)

$$\mathrm{d}Y_t = \left(Y_t + s^\star_{T-t}(Y_t)\right)\beta(T-t)\mathrm{d}t$$

Interpretations from lens of SDE/ODE



discrete-time



continuous-time

forward process

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, I_d)$$
$$\Longrightarrow \quad \mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t \quad \text{(SDE)}$$

- $\exists$ reverse-time ODE w/ same *marginal* dist (Song et al. '20)

time discretization $\longrightarrow$ DDIM

Key takeaway: in continuous-time limits, sampling is feasible once exact score functions are available

— *almost no restriction on target data distributions*

Key takeaway: in continuous-time limits, sampling is feasible once exact score functions are available

— *almost no restriction on target data distributions*

Questions:

- what happens in discrete time? — effect of discretization error
- what if we only have imperfect scores? — effect of score error

An incomplete list of prior theory

- Lee, Lu, Tan '22
- Chen, Chewi, Li, Li, Salim, Zhang '22
- Chen, Lee, Lu '22
- Lee, Lu, Tan '23
- Chen, Daras, Dimakis '23
- Chen, Chewi, Lee, Li, Lu, Salim '23
- Benton, De Bortoli, Doucet, Deligiannidis '23
- Li, Wei, Chen, Chi '23
- Benton, Deligiannidis, Doucet '23
- Cheng, Lu, Tan, Xie '23
- Tang '23
- Li, Wei, Chi, Chen '24
- Li, Yan '24a, '24b
- Azangulov, Deligiannidis, Rousseau '24
- Potaptchik, Azangulov, Deligiannidis '24
- Huang, Wei, Chen '24
- Gao, Zhu '24
- Huang, Huang, Lin '24
- Li, Jiao '24
- Li, Di, Gu '24
- Liang, Ju, Liang, Shroff '24
- Tang, Zhao '24
- Liang, Huang, Chen '25
- Li, Cai, Wei '25
- Tang, Yan '25
- Yu, Yu '25
- Gentiloni-Silveri, Ocello '25
- Li, Zhou, Wei, Chen '25
- Jain, Zhang '25
- Jiao, Zhou, Li '25
- ...

Typical assumptions

- log-concavity of target distribution p_{data}
- smoothness of score function $s_t^\star = \nabla \log p_t(x)$
- ℓ_∞ / ℓ_2 loss of the score estimate

Our assumptions: target distribution p_{data}

$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1$ for arbitrarily large const $c_R > 0$

Our assumptions: target distribution p_{data}

$$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1 \text{ for arbitrarily large const } c_R > 0$$

- support size can be very large

Our assumptions: target distribution p_{data}

$$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1 \text{ for arbitrarily large const } c_R > 0$$

- support size can be very large
- very general: *no need of log-concavity, smoothness, etc*

Our assumptions: target distribution p_{data}

$$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1 \text{ for arbitrarily large const } c_R > 0$$

- support size can be very large
- very general: *no need of log-concavity, smoothness, etc*
- can also be replaced by $\mathbb{E}[\|X_0\|_2] \leq T^{c_M}$ for large const c_M

Our assumptions: score estimates $\{s_t(\cdot)\}$

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

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

Our assumptions: score estimates $\{s_t(\cdot)\}$

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

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

  - much weaker than *pointwise* score error assumption

Our assumptions: score estimates $\{s_t(\cdot)\}$

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

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

  - much weaker than *pointwise* score error assumption
  - suffices for DDPM but not DDIM (counterexample in Li et al. '24)

Our assumptions: score estimates $\{s_t(\cdot)\}$

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

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

  - much weaker than *pointwise* score error assumption
  - suffices for DDPM but not DDIM (counterexample in Li et al. '24)

- *Jacobian estimation error (for DDIM only):*

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

Assumptions: learning rates

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

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

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

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

A glimpse of convergence theory (up to log factor)

DDPM:

$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

A glimpse of convergence theory (up to log factor)

DDPM:
$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{KL} \le \varepsilon^2}$ d/ε (assuming accurate scores)
- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{TV} \le \varepsilon}$ d/ε (assuming accurate scores)

A glimpse of convergence theory (up to log factor)

DDPM:
$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

- **iteration complexity:** d/ε (assuming accurate scores)
  to yield $\mathsf{KL} \le \varepsilon^2$
- **iteration complexity:** d/ε (assuming accurate scores)
  to yield $\mathsf{TV} \le \varepsilon$
  - Pinsker inequality ($\mathsf{TV} \le \sqrt{\frac{1}{2}\mathsf{KL}}$) is loose when bounding TV

A glimpse of convergence theory (up to log factor)

DDPM:
$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{KL} \le \varepsilon^2}$ d/ε (assuming accurate scores)
- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{TV} \le \varepsilon}$ d/ε (assuming accurate scores)
- **stability:** degrades gracefully as $\varepsilon_{\mathsf{score}} \uparrow$

A glimpse of convergence theory (up to log factor)

DDPM:
$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

- **iteration complexity:** d/ε (assuming accurate scores)
  to yield $\mathsf{KL} \le \varepsilon^2$
- **iteration complexity:** d/ε (assuming accurate scores)
  to yield $\mathsf{TV} \le \varepsilon$
- **stability:** degrades gracefully as $\varepsilon_{\mathsf{score}} \uparrow$ and $\varepsilon_{\mathsf{Jacobi}} \uparrow$
- **general data distribution:** no need of smoothness, log-concavity

A glimpse of convergence theory (up to log factor)

DDPM:
$$\mathsf{KL}(p_{X_1} \,\|\, p_{Y_1}) \lesssim d^2/T^2 + \varepsilon_{\mathsf{score}}^2 \quad \text{(Jiao, Zhou, Li '25)}$$
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \varepsilon_{\mathsf{score}} \quad \text{(Li, Yan '24)}$$

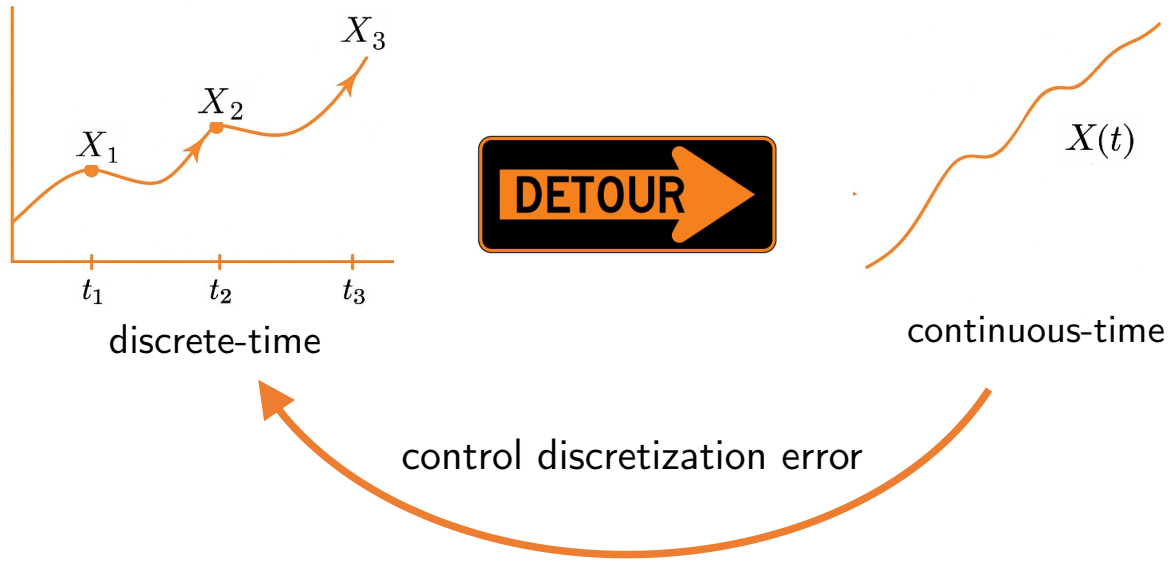
DDIM:
$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim d/T + \sqrt{d}\,\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}} \quad \text{(Li et al. '24)}$$

- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{KL} \le \varepsilon^2}$ d/ε (assuming accurate scores)
- $\underbrace{\textbf{iteration complexity:}}_{\text{to yield } \mathsf{TV} \le \varepsilon}$ d/ε (assuming accurate scores)
- **stability:** degrades gracefully as $\varepsilon_{\mathsf{score}} \uparrow$ and $\varepsilon_{\mathsf{Jacobi}} \uparrow$
- **general data distribution:** no need of smoothness, log-concavity

Analysis strategy # 1 (for DDPM)

— *Chen, Chewi, Li, Li, Salim, Zhang '22, Chen, Lee, Lu '22, Tang, Zhao '24*

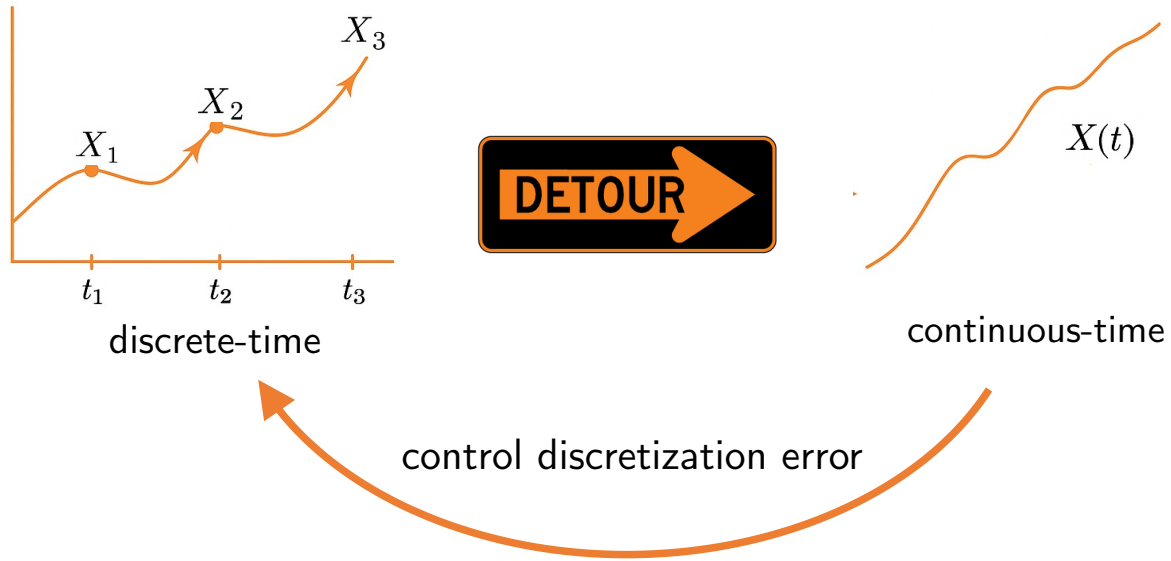
— *Benton, De Bortoli, Doucet, Deligiannidis '23, Huang, Wei, Chen '24*



Analysis strategy # 1 (for DDPM)

— *Chen, Chewi, Li, Li, Salim, Zhang '22, Chen, Lee, Lu '22, Tang, Zhao '24*

— *Benton, De Bortoli, Doucet, Deligiannidis '23, Huang, Wei, Chen '24*

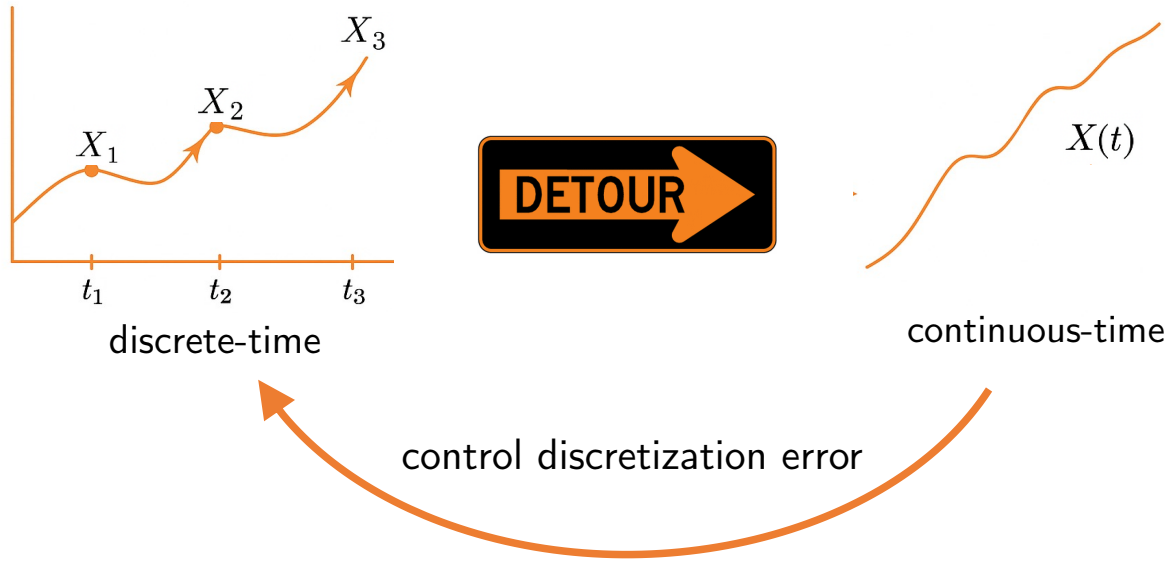


Analogy: (stochastic) gradient descent vs. gradient flow, TD learning via ODE

Analysis strategy # 1 (for DDPM)

— *Chen, Chewi, Li, Li, Salim, Zhang '22, Chen, Lee, Lu '22, Tang, Zhao '24*

— *Benton, De Bortoli, Doucet, Deligiannidis '23, Huang, Wei, Chen '24*



2 key steps:

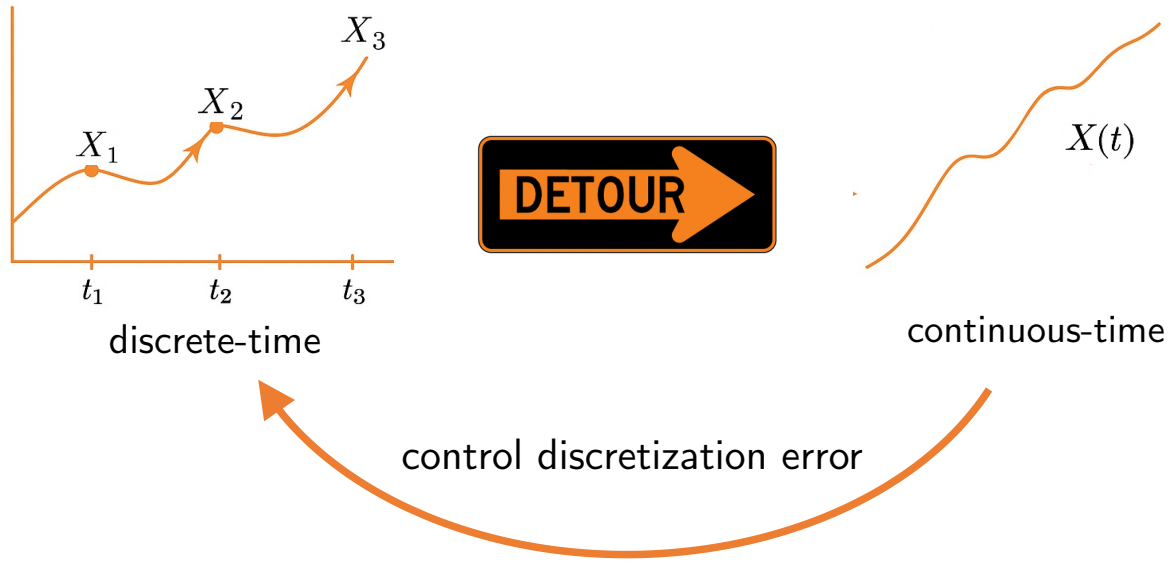
- apply change of measure (e.g. Girsanov thm) to show

$$\mathsf{KL}\big(P^{\mathsf{true}} \,\|\, P^{\mathsf{ddpm}}\big) \le \int w(t) \underbrace{\mathbb{E}\Big[\big\|\mathsf{drift}^{\mathsf{true}}(t) - \mathsf{drift}^{\mathsf{ddpm}}(t)\big\|\Big]^2}_{\text{score error + discretization error}} \mathrm{d}t + \mathsf{small\text{-}term}$$

Analysis strategy # 1 (for DDPM)

— *Chen, Chewi, Li, Li, Salim, Zhang '22,* *Chen, Lee, Lu '22,* *Tang, Zhao '24*

— *Benton, De Bortoli, Doucet, Deligiannidis '23, Huang, Wei, Chen '24*



2 key steps:

- leverage stochastic localization to chacracterize

$$\text{discretization error} \overset{\text{link}}{\longleftrightarrow} \mathbb{E}\big[\mathrm{Cov}\big(X_0 \,|\, X_t\big)\big]$$

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

yields state-of-the-art theory for DDIM & DDPM!

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

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

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

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

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

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

$t-1$ t T

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

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

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

$t-1$ t T

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

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

Analysis strategy # 2 (for DDIM & DDPM)

— Li, Wei, Chen, Chi '24, Li, Wei, Chi, Chen '24

— Li, Yan '24, Liang, Huang, Chen '25, Jiao, Zhou, Li '25

Tackle discrete-time process directly & track changes of, e.g., TV distance

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

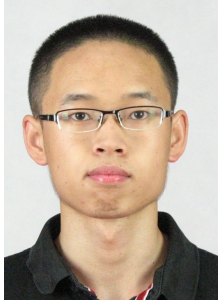
$t-1$ t T

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

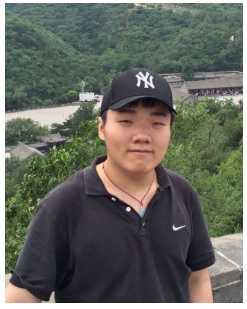
$$\frac{p_{\Phi_t(Y_t)}\big(\Phi_t(y_t)\big)}{p_{Y_t}(y_t)} = \det\left(\frac{\partial \Phi_t}{\partial y_t}\right)^{-1}$$

some concentration bounds

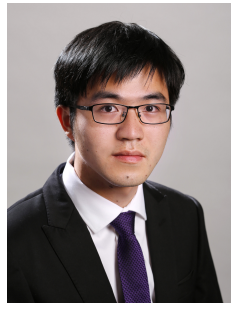
Part 2: adaptation to (unknown) low dimensionality



Gen Li
CUHK



Yuling Yan
DE Shaw



Changxiao Cai
UMich

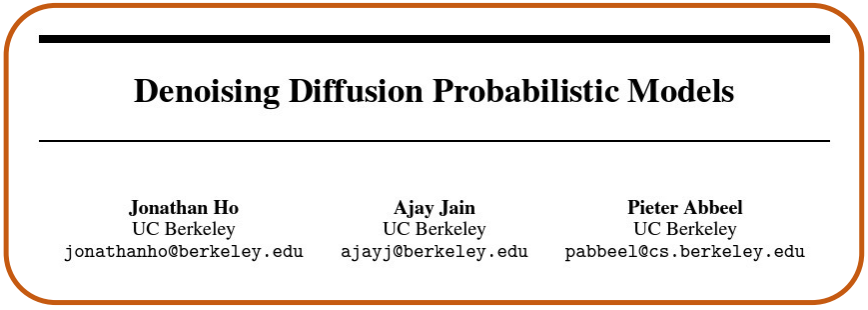
Jiadong Liang
UPenn

Zhihan Huang
UPenn

Yuxin Chen
UPenn

Yuting Wei
UPenn

Recap: theory for mainstream diffusion models



Theorem 1 (Li, Wei, Chi, Chen '24, Li, Yan '24)

With perfect scores, both DDIM & DDPM yield $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}(d/\varepsilon) \text{ iterations}$$

- d: *ambient dimension*

d/ε iterations are too slow ...

ImageNet: $d = 150,528$ pixels per image

d/ε iterations are too slow ...

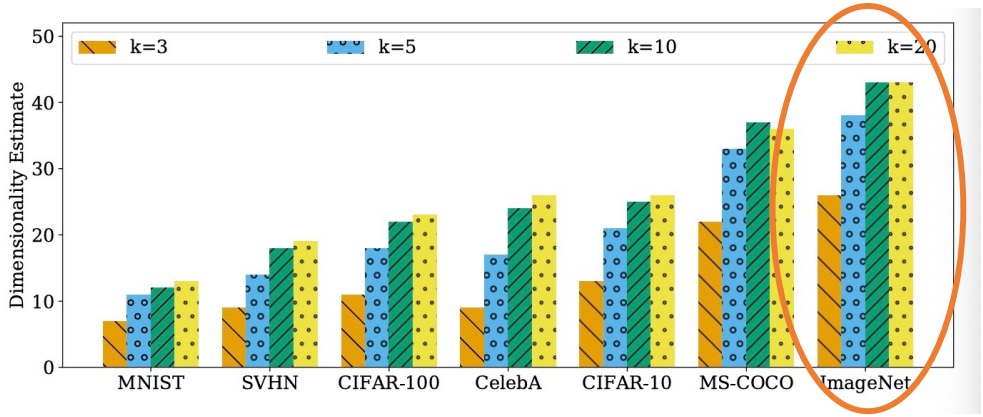
ImageNet: $d = 150,528$ pixels per image (so $\frac{d}{\varepsilon} > 10^6$ for moderate ε)

d/ε iterations are too slow ...

ImageNet: $d = 150,528$ pixels per image (so $\frac{d}{\varepsilon} > 10^6$ for moderate ε)

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

d/ε iterations are too slow ...



ImageNet: $d = 150,528$ pixels per image (so $\frac{d}{\varepsilon} > 10^6$ for moderate ε)
$k = 43$ 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 p_{data} is said to have intrinsic dimension k if

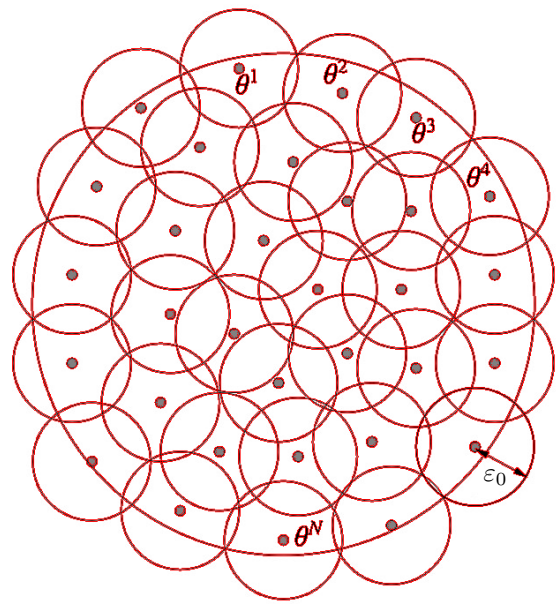
$$\log \underbrace{N^{\text{cover}}(\text{support}(p_{\text{data}}), \|\cdot\|_2, \varepsilon_0)}_{\text{covering number of support of } p_{\text{data}}} \lesssim k \log \frac{1}{\varepsilon_0}$$

Intrinsic dimension

The target distribution p_{data} is said to have intrinsic dimension k if

$$\log \underbrace{N^{\text{cover}}(\text{support}(p_{\text{data}}), \|\cdot\|_2, \varepsilon_0)}_{\text{covering number of support of } p_{\text{data}}} \lesssim k \log \frac{1}{\varepsilon_0}$$

- k-dimensional linear subspaces
- k-dimensional manifolds
- union of the above
- ...



— *see Li, Yan '24, Huang, Wei, Chen '24*

Assumptions

- **minimal data assumptions:**

$$\mathbb{P}(\|X_0\|_2 \leq \underbrace{T^{c_R}}_{\text{polynomially large diameter}}) = 1$$

  for arbitrarily large constant $c_R > 0$

- **perfect score estimates:** $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$
  $\longrightarrow$ not needed; only to simplify presentation

Convergence theory in KL divergence

Theorem 2 (Li and Yan '24; Huang, Wei, Chen '24)

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

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

— *related work Azangulov et al. '24, Potaptchik et al.'24*

Convergence theory in KL divergence

Theorem 2 (Li and Yan '24; Huang, Wei, Chen '24)

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

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

— *related work Azangulov et al. '24, Potaptchik et al.'24*

- optimal scaling in k

Convergence theory in total variation

Theorem 3 (Liang, Huang, Chen '25; Li, Yan '25; Tang, Yan '25)

Both DDPM & DDIM (their original form) yield $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

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

DDIM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + \eta_t^{\mathsf{ddim}} s_t(Y_t)\right)$$

DDPM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + \eta_t^{\mathsf{ddpm}} s_t(Y_t) + \sigma_t^{\mathsf{ddpm}} \mathcal{N}(0, I_d)\right)$$

Convergence theory in total variation

Theorem 3 (Liang, Huang, Chen '25; Li, Yan '25; Tang, Yan '25)

Both DDPM & DDIM (their original form) yield $\mathsf{TV}(p_{X_1}, p_{Y_1}) \le \varepsilon$ *in*

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

DDIM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + \frac{1-\alpha_t}{1+\sqrt{\frac{\alpha_t-\overline{\alpha}_t}{1-\overline{\alpha}_t}}} s_t(Y_t)\right)$$

DDPM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{\frac{(1-\alpha_t)(\alpha_t-\overline{\alpha}_t)}{1-\overline{\alpha}_t}}\mathcal{N}(0, I_d)\right)$$

where $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

Convergence theory in total variation

Theorem 3 (Liang, Huang, Chen '25; Li, Yan '25; Tang, Yan '25)

Both DDPM & DDIM (their original form) yield $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

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

DDIM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + \frac{1-\alpha_t}{1+\sqrt{\frac{\alpha_t-\overline{\alpha}_t}{1-\overline{\alpha}_t}}} s_t(Y_t)\right)$$

DDPM: $$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(Y_t + (1-\alpha_t)s_t(Y_t) + \sqrt{\frac{(1-\alpha_t)(\alpha_t-\overline{\alpha}_t)}{1-\overline{\alpha}_t}}\mathcal{N}(0, I_d)\right)$$

where $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

- *originally derived to optimize variational lower bounds!*

Interpretation from lens of SDE/ODE

reverse-time SDE (same distribution as X_t):

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

Interpretation from lens of SDE/ODE

reverse-time SDE (same distribution as X_t):

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

Tweedie's formula $\Downarrow$

$$\mathrm{d}Y_t = \big(\underbrace{c_{1,t}Y_t}_{\text{linear drift}} + c_{2,t}\underbrace{\mathbb{E}[X_0 \mid X_{T-t} = Y_t]}_{\text{cond. mean of } X_0}\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

- **key enabler:** $\mathbb{E}[X_0 \mid X_t]$ is "projection" onto low-dimensional structure

Interpretation from lens of SDE/ODE

reverse-time SDE (same distribution as X_t):

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

Tweedie's formula $\Downarrow$

$$\mathrm{d}Y_t = \big(\underbrace{c_{1,t}Y_t}_{\text{linear drift}} + c_{2,t}\underbrace{\mathbb{E}[X_0 \mid X_{T-t} = Y_t]}_{\text{cond. mean of } X_0}\big)\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

time-discretize $\Downarrow$

DDIM or DDPM

- **key enabler:** $\mathbb{E}[X_0 \mid X_t]$ is "projection" onto low-dimensional structure
- **discretization scheme matters:** time-discretize carefully to retain low-dimensional adaptation

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'95
- 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

Dimension-free convergence of DDPM for GMMs

Gaussian mixture models (Pearson'94)

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

Theorem 4 (Li, Cai, Wei '25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, DDPM yields

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

Dimension-free convergence of DDPM for GMMs

Gaussian mixture models (Pearson'94)

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

Theorem 4 (Li, Cai, Wei '25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, DDPM yields

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

- to yield ε-accuracy, it requires $\underbrace{\widetilde{O}(1/\varepsilon)}_{\text{dimension-free}}$ iterations

Dimension-free convergence of DDPM for GMMs

Gaussian mixture models (Pearson'94)

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

Theorem 4 (Li, Cai, Wei '25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, DDPM yields

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

- to yield ε-accuracy, it requires $\underbrace{\widetilde{O}(1/\varepsilon)}_{\text{dimension-free}}$ iterations
- stable vis-a-vis score errors

Dimension-free convergence of DDPM for GMMs

Gaussian mixture models (Pearson'94)

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

Theorem 4 (Li, Cai, Wei '25)

For spherical Gaussian mixture with $\Sigma = \sigma^2 I_d$, DDPM yields

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

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

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

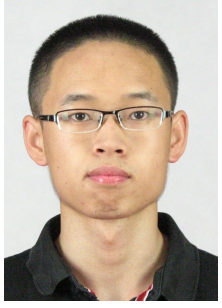
- Liang, Shi, Song, Zhou'24: $\underbrace{\text{provides } \widetilde{O}(d/\varepsilon^2) \text{ complexity bound}}_{\text{show no adaptation phenomenon}}$

Comparison w/ prior theory of GMMs

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

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

Part 3: acceleration via higher-order approximation

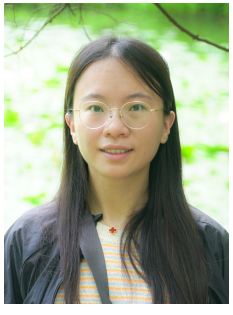


Gen Li
CUHK

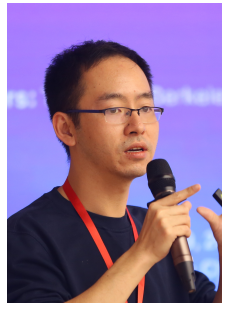
Yuchen Zhou
UIUC

Yu Huang
UPenn

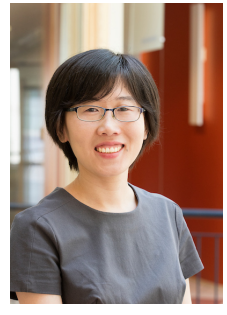
Timofey Efimov
CMU



Yuchen Wu
Cornell



Yuxin Chen
UPenn



Yuejie Chi
Yale

Yuting Wei
UPenn

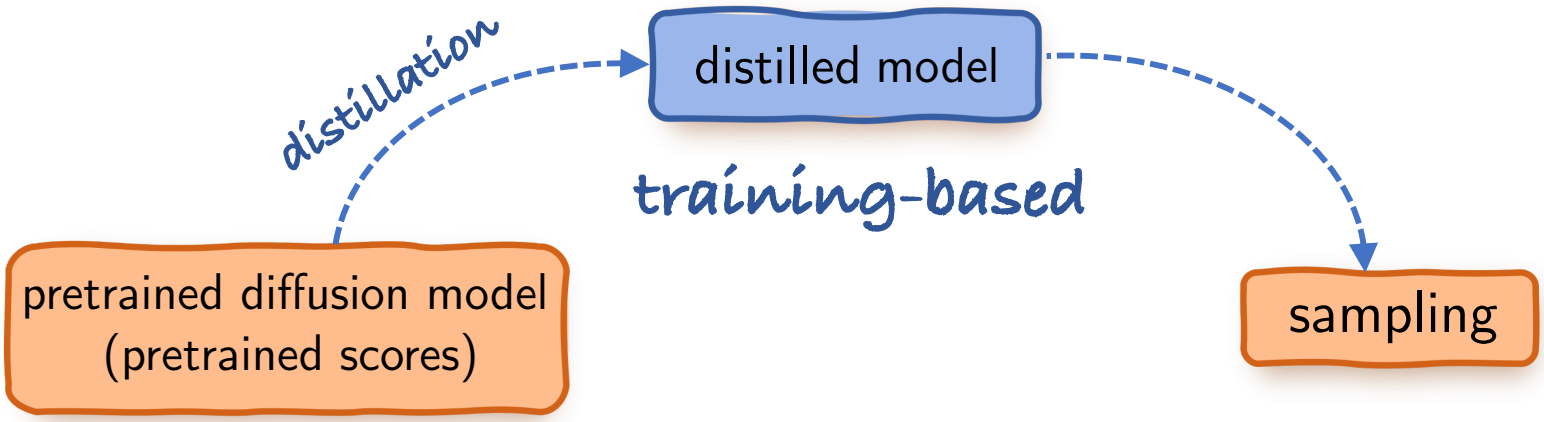
DDPM and DDIM are still slow . . .

— *Song, Meng, Ermon '20*

DDPM and DDIM are still slow . . .

50K 32×32 images: DDPM (20h) *vs.* single-step GANs ($<$ 1min)

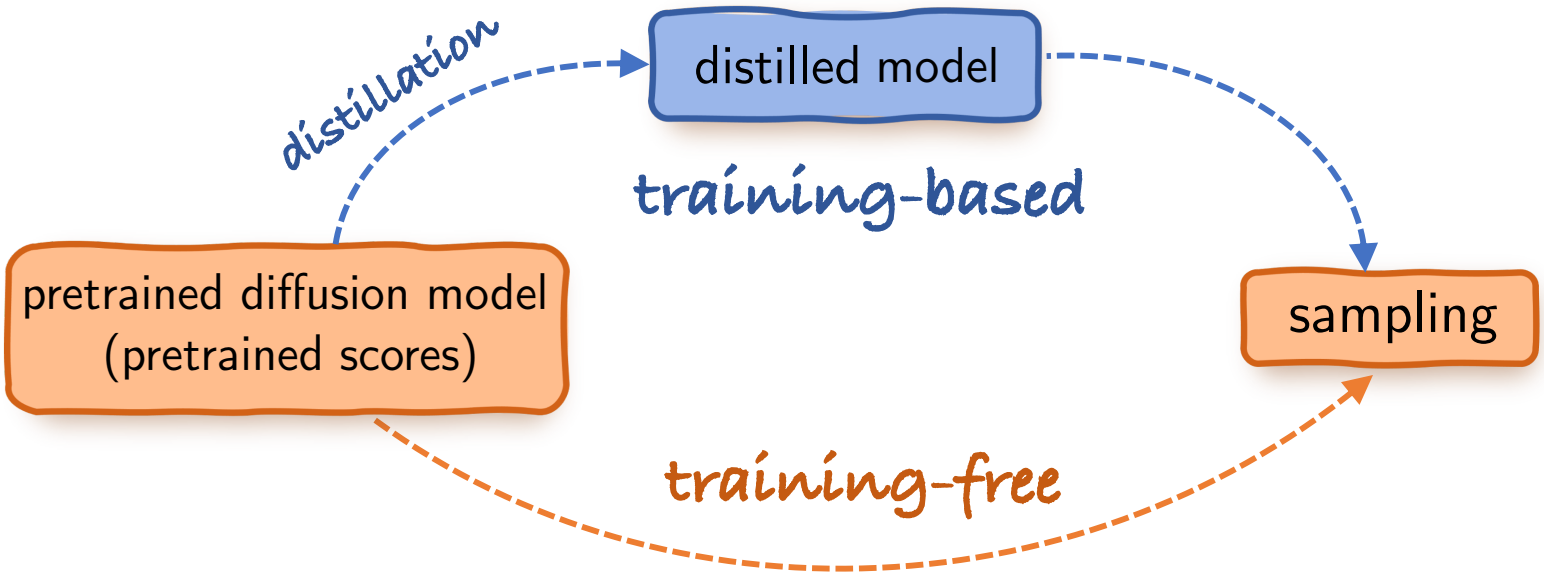
— *Song, Meng, Ermon '20*



- **training-based:** distill pre-trained diffusion model into another model that can be executed rapidly

  requires additional training

  - e.g., progressive distillation, consistency model



- **training-free:** directly invoke pre-trained score estimates for sampling w/o additional training
  - e.g., DPM-Solver/++ (Lu et al. '22), UniPC (Zhao et al. '23), ...

Can we design a training-free sampler that is provably faster than DDIM/DDPM?

Discretization $\longleftrightarrow$ approximation

A starting point: equiv solution to probability flow ODE

$$\underbrace{Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}}}_{\text{represent } Y_{t-1}} = \frac{1}{\sqrt{\alpha_t}} \underbrace{Y^{\mathsf{ode}}_{\overline{\alpha}_t}}_{\text{represent } Y_t} + \underbrace{\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s^\star_\gamma(Y^{\mathsf{ode}}_\gamma) \mathrm{d}\gamma}$$

where $s^\star_\gamma(x) := \nabla \log p_{\sqrt{\gamma}X_0 + \sqrt{1-\gamma}\mathcal{N}(0,I_d)}(x)$

Discretization $\longleftrightarrow$ approximation

A starting point: equiv solution to probability flow ODE

$$\underbrace{Y^{\text{ode}}_{\overline{\alpha}_{t-1}}}_{\text{represent } Y_{t-1}} = \frac{1}{\sqrt{\alpha_t}} \underbrace{Y^{\text{ode}}_{\overline{\alpha}_t}}_{\text{represent } Y_t} + \underbrace{\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s^\star_\gamma(Y^{\text{ode}}_\gamma) \mathrm{d}\gamma}_{\text{infinitely many score evaluations!}}$$

where $s^\star_\gamma(x) := \nabla \log p_{\sqrt{\gamma}X_0 + \sqrt{1-\gamma}\mathcal{N}(0,I_d)}(x)$

Discretization $\longleftrightarrow$ approximation

A starting point: equiv solution to probability flow ODE

$$\underbrace{Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}}}_{\text{represent } Y_{t-1}} = \frac{1}{\sqrt{\alpha_t}} \underbrace{Y^{\mathsf{ode}}_{\overline{\alpha}_t}}_{\text{represent } Y_t} + \underbrace{\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \mathrm{d}\gamma}_{\text{infinitely many score evaluations!}}$$

where $s^\star_\gamma(x) := \nabla \log p_{\sqrt{\gamma} X_0 + \sqrt{1-\gamma}\mathcal{N}(0, I_d)}(x)$

- can we approximate the integral by a few score evals?

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\left(Y^{\mathsf{ode}}_\gamma\right)}_{\text{approximate by?}} \mathrm{d}\gamma$$

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\left(Y^{\mathsf{ode}}_\gamma\right)}_{\text{approximate by?}} \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\left(Y^{\mathsf{ode}}_\gamma\right) \approx s^\star_{\overline{\alpha}_t}\left(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\right) \approx s_t(Y_t)$

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\left(Y^{\mathsf{ode}}_\gamma\right)}_{\text{approximate by?}} \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\left(Y^{\mathsf{ode}}_\gamma\right) \approx s^\star_{\overline{\alpha}_t}\left(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\right) \approx s_t(Y_t)$

$$\Longrightarrow \quad Y_{t-1} \approx \frac{1}{\sqrt{\alpha_t}}\left(Y_t + \frac{1-\alpha_t}{2} s_t(Y_t)\right) \qquad \text{original DDIM}$$

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; $\quad 1$ score eval per iteration $\qquad$ (DDIM)

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; $\quad 1$ score eval per iteration $\qquad$ (DDIM)

refined approximation?

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; 1 score eval per iteration (DDIM)

refined approximation?

$$s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s_t(Y_t) + \frac{\gamma - \overline{\alpha}_t}{\overline{\alpha}_t - \overline{\alpha}_{t+1}} \Big(s_t(Y_t) - s_{t+1}(Y_{t+1}) \Big)$$

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; 1 score eval per iteration (DDIM)

2nd order approx: (Li, Huang, Efimov, Wei, Chi, Chen '24)

$$\sqrt{\alpha_t} Y_{t-1} \approx Y_t + \frac{1-\alpha_t}{2} s_t(Y_t) + \frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})}\Big(s_t(Y_t) - \sqrt{\alpha_{t+1}} s_{t+1}(Y_{t+1})\Big)$$

— *similar in spirit to DPM-Solver-2 (Lu et al '22)*

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; 1 score eval per iteration (DDIM)

2nd order approx: (Li, Huang, Efimov, Wei, Chi, Chen '24)

$\frac{\mathrm{poly}(d)}{\sqrt{\varepsilon}}$ iterations; 2 score evals per iteration

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\mathsf{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; $\quad 1$ score eval per iteration $\qquad$ (DDIM)

even higher-order approximation?

$$Y^{\text{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\text{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\text{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\text{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\text{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

$\frac{d}{\varepsilon}$ iterations; 1 score eval per iteration (DDIM)

even higher-order approximation? for order K:

$$\frac{1}{\gamma^{3/2}} s^\star_\gamma(Y^{\text{ode}}_\gamma) \approx \sum\nolimits_{0 \le i < K} \psi_i(\gamma) \frac{s_{\gamma_{t,i}}(Y^{\text{ode}}_{\gamma_{t,i}})}{(\gamma_{t,i})^{3/2}}$$

- K anchor points: $\gamma_{t,0}, \ldots, \gamma_{t,K-1}$

$$Y^{\text{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\text{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\text{ode}}_\gamma\big)}_{\text{approximate by?}} \, \mathrm{d}\gamma$$

1st order approx: $s^\star_\gamma\big(Y^{\text{ode}}_\gamma\big) \approx s^\star_{\overline{\alpha}_t}\big(Y^{\text{ode}}_{\overline{\alpha}_t}\big) \approx s_t(Y_t)$

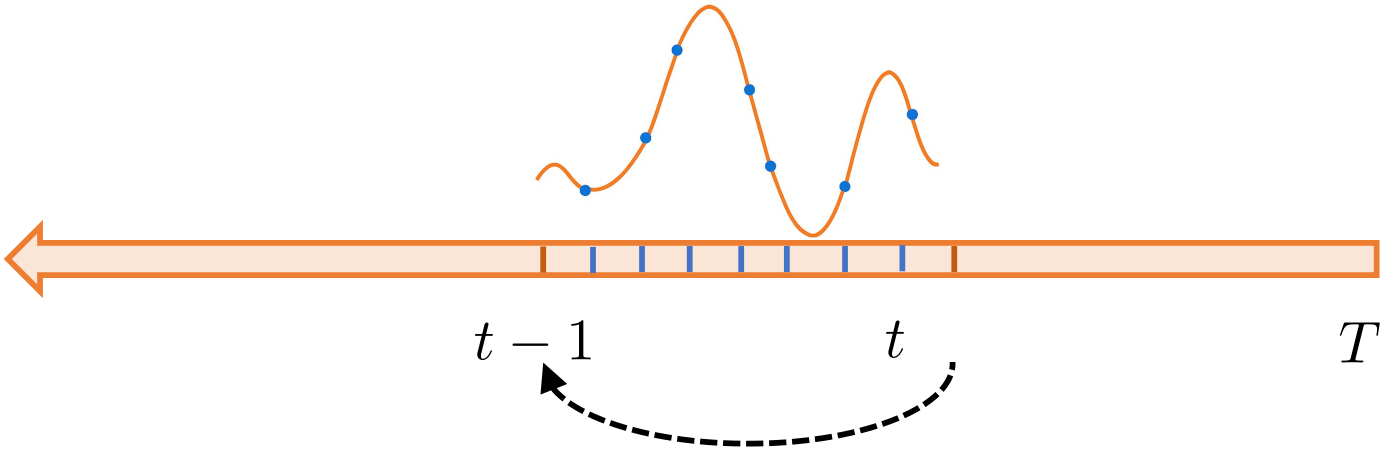
$\frac{d}{\varepsilon}$ iterations; 1 score eval per iteration (DDIM)

even higher-order approximation? for order K:

$$\frac{1}{\gamma^{3/2}} s^\star_\gamma(Y^{\text{ode}}_\gamma) \approx \sum\nolimits_{0 \le i < K} \psi_i(\gamma) \frac{s_{\gamma_{t,i}}(Y^{\text{ode}}_{\gamma_{t,i}})}{(\gamma_{t,i})^{3/2}}$$

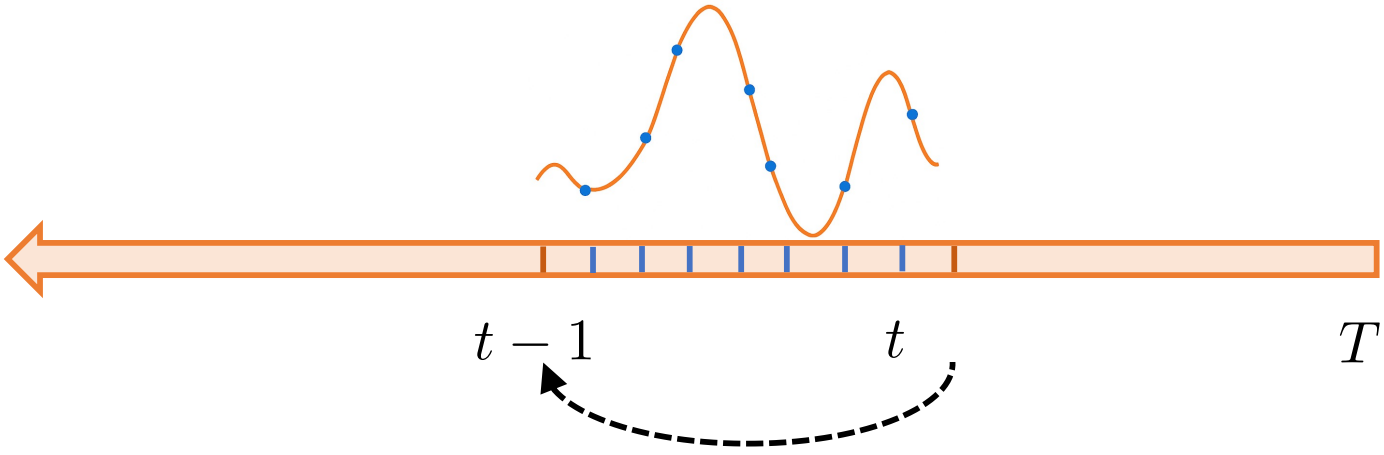
- K anchor points: $\gamma_{t,0}, \ldots, \gamma_{t,K-1}$
- Lagrange basis polynomial: $\psi_i(\gamma) := \frac{\prod_{i':i' \neq i} (\gamma - \gamma_{t,i'})}{\prod_{i':i' \neq i} (\gamma_{t,i} - \gamma_{t,i'})}$

Proposed K-th order sampler (Li et al. '25)



$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approx by deg-}(K-1)\text{ Lagrange polynomials}} \mathrm{d}\gamma$$

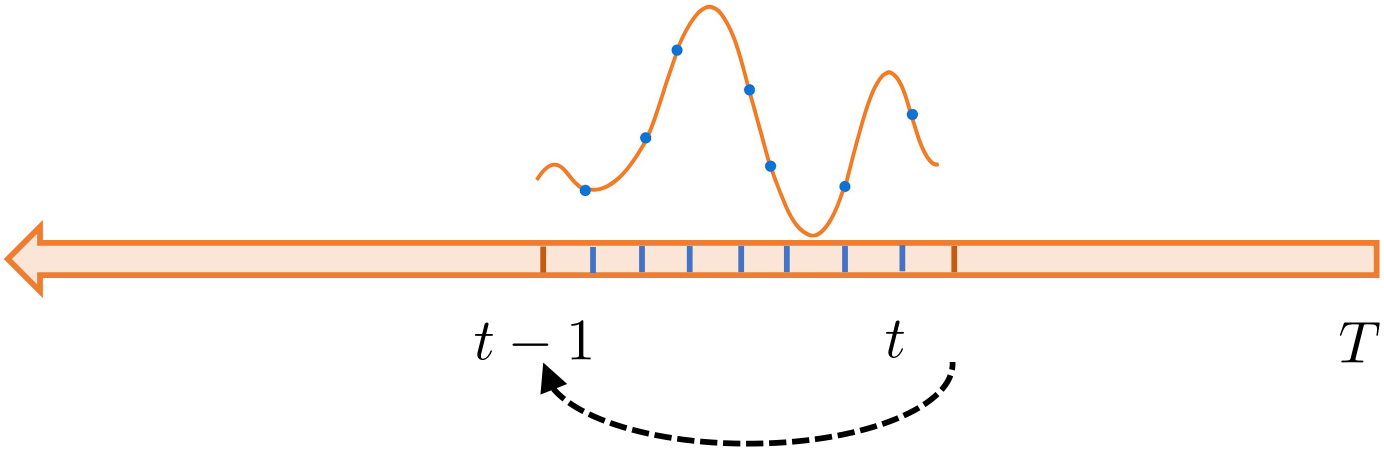
Proposed K-th order sampler (Li et al. '25)



$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approx by deg-}(K-1)\text{ Lagrange polynomials}} \mathrm{d}\gamma$$

- successively, alternately refine $Y^{\mathsf{ode}}_{\gamma_{t,i}}$ and $s_{\gamma_{t,i}}\big(Y^{\mathsf{ode}}_{\gamma_{t,i}}\big)$

Proposed K-th order sampler (Li et al. '25)



$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(Y^{\mathsf{ode}}_\gamma\big)}_{\text{approx by deg-}(K-1)\text{ Lagrange polynomials}} \mathrm{d}\gamma$$

- successively, alternately refine $Y^{\mathsf{ode}}_{\gamma_{t,i}}$ and $s_{\gamma_{t,i}}\big(Y^{\mathsf{ode}}_{\gamma_{t,i}}\big)$

K score evals per iteration; $\widetilde{O}(1)$ rounds of refinements

Convergence theory for our accelerated sampler

Theorem 5 (Li, Zhou, Wei, Chen '25)

Consider any $K = O(1)$. With perfect scores, our accelerated deterministic sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \le \varepsilon$ *in*

$$\widetilde{O}\big(d^{1+2/K}/\varepsilon^{1/K}\big) \text{ iterations}$$

Convergence theory for our accelerated sampler

Theorem 5 (Li, Zhou, Wei, Chen '25)

Consider any $K = O(1)$. With perfect scores, our accelerated deterministic sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}\big(d^{1+2/K}/\varepsilon^{1/K}\big) \text{ iterations}$$

- **# score function evaluations**: $\frac{d^{1+o(1)}}{\varepsilon^{1/K}}$

Convergence theory for our accelerated sampler

Theorem 5 (Li, Zhou, Wei, Chen '25)

Consider any $K = O(1)$. With perfect scores, our accelerated deterministic sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}\big(d^{1+2/K}/\varepsilon^{1/K}\big) \text{ iterations}$$

- **# score function evaluations**: $\frac{d^{1+o(1)}}{\varepsilon^{1/K}}$
- outperforms vanilla DDIM (d/ε)
  - substantially improved ε-dependency
  - almost no loss in d-dependency;

Convergence theory for our accelerated sampler

Theorem 5 (Li, Zhou, Wei, Chen '25)

Consider any $K = O(1)$. With perfect scores, our accelerated deterministic sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \le \varepsilon$ *in*

$$\widetilde{O}\big(d^{1+2/K}/\varepsilon^{1/K}\big) \text{ iterations}$$

- **# score function evaluations**: $\frac{d^{1+o(1)}}{\varepsilon^{1/K}}$
- outperforms vanilla DDIM (d/ε)
  - substantially improved ε-dependency
  - almost no loss in d-dependency;
- **minimal assumptions** on data distributions
  - see also Huang et al. '24, '25 (Runge-Kutta; stronger assumptions)

Can we design algorithms to improve the dependency on dimension d?

Non-uniform Lipschitz property

Non-uniform Lipschitz constant $L \geq 1$: for every $\gamma \in (0,1)$,

$$\mathbb{P}\left((1-\gamma)\|\nabla s^\star_\gamma(X_\gamma)\|_2 \leq L\right) \geq 1 - \frac{1}{d^4}.$$

Non-uniform Lipschitz property

Non-uniform Lipschitz constant $L \geq 1$: for every $\gamma \in (0,1)$,

$$\mathbb{P}\left((1-\gamma)\|\nabla s_\gamma^\star(X_\gamma)\|_2 \leq L\right) \geq 1 - \frac{1}{d^4}.$$

- much weaker than the global $\|\nabla s_\gamma^\star(x)\|_2 \leq \widetilde{L}$ for all $x \in \mathbb{R}^d$ and τ

Non-uniform Lipschitz property

Non-uniform Lipschitz constant $L \geq 1$: for every $\gamma \in (0,1)$,

$$\mathbb{P}\left((1-\gamma)\|\nabla s_\gamma^\star(X_\gamma)\|_2 \leq L\right) \geq 1 - \frac{1}{d^4}.$$

- much weaker than the global $\|\nabla s_\gamma^\star(x)\|_2 \leq \widetilde{L}$ for all $x \in \mathbb{R}^d$ and τ
- Examples that L is small but $\widetilde{L}$ is large
  - $X_0 \sim N(\mu, \Sigma) \Rightarrow L = 1$, $\widetilde{L} = \infty$ for singular Σ
  - $X_0 \sim \sum_{h=1}^H \pi_h \mathcal{N}(\mu_h, \sigma_h^2 I_d) \Rightarrow L \lesssim \log(d)\log(H)$, $\widetilde{L}$ can be $\Omega(\|\mu\|_2^2) \approx \mathbb{E}[\|X_0\|_2^2]$ even when $\sigma_h^2 = 1$
  - X_0 has indep. entries with $\mathbb{E}[|X_{0,i}|] \leq d^{c_R} \Rightarrow L \lesssim \log d$, $\widetilde{L}$ can be infinite

Convergence theory for DDPM

Theorem 6 (Jiao, Zhou, Li '25)

DDPM yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \le \varepsilon$ *and* $\mathsf{KL}(p_{X_1} \| p_{Y_1}) \le \varepsilon^2$ *in*

$$\frac{\min\{L\sqrt{d}, d\}}{\varepsilon} \text{ iterations}$$

Convergence theory for DDPM

Theorem 6 (Jiao, Zhou, Li '25)

DDPM yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *and* $\mathsf{KL}(p_{X_1} \| p_{Y_1}) \leq \varepsilon^2$ *in*

$$\frac{\min\{L\sqrt{d}, d\}}{\varepsilon} \text{ iterations}$$

- When $L = \widetilde{O}(1)$, iteration complexity: $\sqrt{d}/\varepsilon$

Convergence theory for DDPM

Theorem 6 (Jiao, Zhou, Li '25)

DDPM yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *and* $\mathsf{KL}(p_{X_1} \| p_{Y_1}) \leq \varepsilon^2$ *in*

$$\frac{\min\{L\sqrt{d}, d\}}{\varepsilon} \text{ iterations}$$

- When $L = \widetilde{O}(1)$, iteration complexity: $\sqrt{d}/\varepsilon$
- When $L = O(\sqrt{d})$, sublinear dependency on d

Convergence theory for DDPM

Theorem 6 (Jiao, Zhou, Li '25)

DDPM yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \le \varepsilon$ *and* $\mathsf{KL}(p_{X_1} \| p_{Y_1}) \le \varepsilon^2$ *in*

$$\frac{\min\{L\sqrt{d}, d\}}{\varepsilon} \text{ iterations}$$

- When $L = \widetilde{O}(1)$, iteration complexity: $\sqrt{d}/\varepsilon$
- When $L = O(\sqrt{d})$, sublinear dependency on d
- When $L = \widetilde{O}(1)$, the lower bound is $\widetilde{\Omega}(\sqrt{d}/\varepsilon)$

Sampler based on randomized midpoint

For integral $\int_0^1 f(x)\mathrm{d}x$, interval width δ:

- Fixed endpoints: error $\propto \|f'\|\delta$
- Randomized midpoints: error $\propto \|f'\|\delta^{3/2}$

— *Gupta et al '24, Shen and Lee '19*

Sampler based on randomized midpoint

For integral $\int_0^1 f(x)\mathrm{d}x$, interval width δ:

- Fixed endpoints: error $\propto \|f'\|\delta$
- Randomized midpoints: error $\propto \|f'\|\delta^{3/2}$

— *Gupta et al '24, Shen and Lee '19*

For ODE in diffusion models:

$$Y^{\mathsf{ode}}_{\overline{\alpha}_{t-1}} = \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma(Y^{\mathsf{ode}}_\gamma)}_{\text{approximate by?}} \mathrm{d}\gamma$$

$$\approx \frac{1}{\sqrt{\alpha_t}} Y^{\mathsf{ode}}_{\overline{\alpha}_t} + \frac{\overline{\alpha}_{t-1} - \overline{\alpha}_t}{\sqrt{\gamma_t^3}} s^\star_{\gamma_t}(Y^{\mathsf{ode}}_{\gamma_t}), \qquad \gamma_t \sim \mathsf{Unif}([\overline{\alpha}_t, \overline{\alpha}_{t-1}])$$

Convergence theory for our accelerated sampler

Theorem 7 (Jiao & Li '25)

Our accelerated sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}\Big(\frac{\min\{d, d^{2/3}L^{1/3}, d^{1/3}L\}}{\varepsilon^{2/3}}\Big) \text{ iterations}$$

Convergence theory for our accelerated sampler

Theorem 7 (Jiao & Li '25)

Our accelerated sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

$$\widetilde{O}\Big(\frac{\min\{d, d^{2/3}L^{1/3}, d^{1/3}L\}}{\varepsilon^{2/3}}\Big) \text{ iterations}$$

- To achieve sublinear dependency on d, our accelerated sampler requires $L = O(d)$ vs. DDPM requires $L = O(\sqrt{d})$

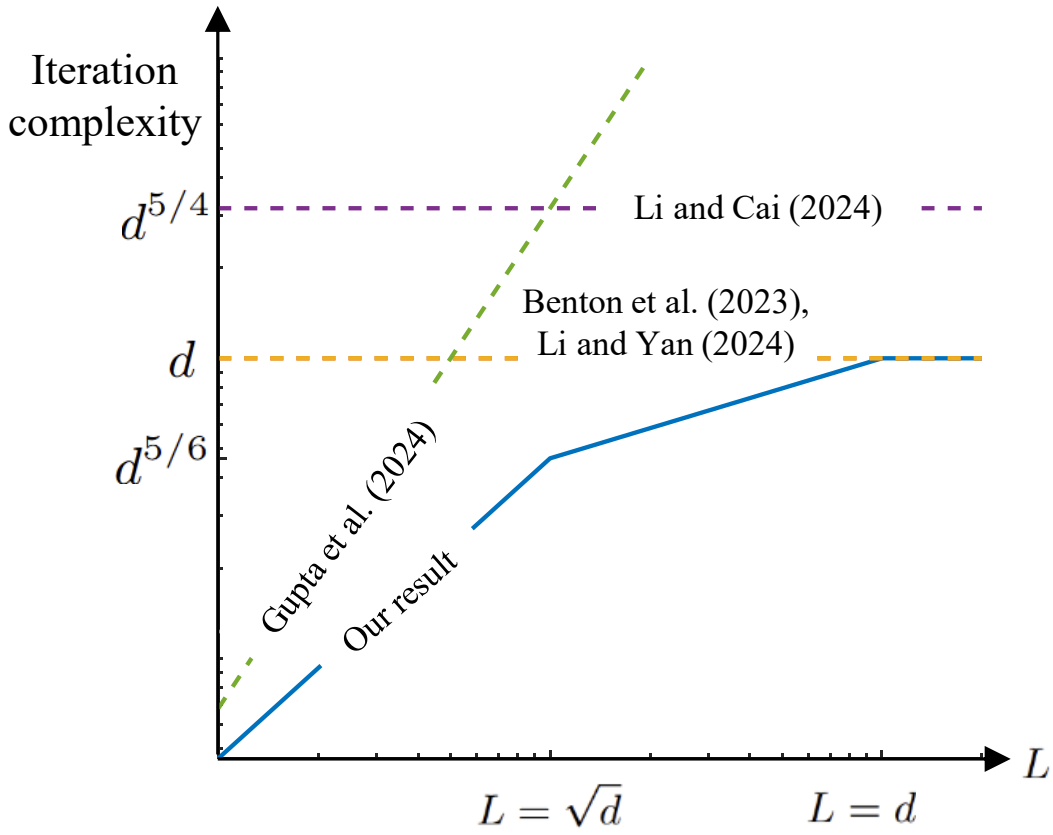
Convergence theory for our accelerated sampler

Theorem 7 (Jiao & Li '25)

Our accelerated sampler yields $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$ *in*

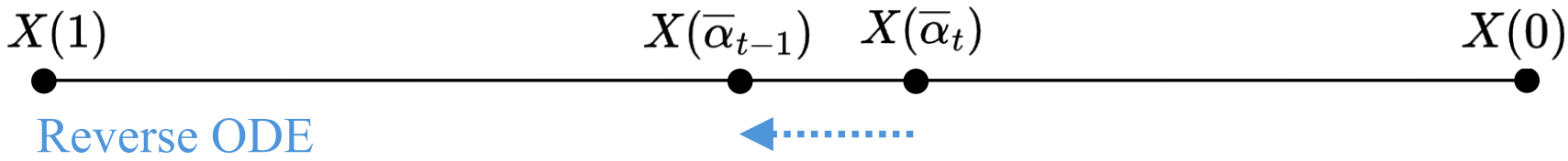
$$\widetilde{O}\Big(\frac{\min\{d, d^{2/3}L^{1/3}, d^{1/3}L\}}{\varepsilon^{2/3}}\Big) \text{ iterations}$$

- When $L = \widetilde{O}(1)$, $O(d^{1/3})$ dependency vs. $O(d^{1/2})$ dependency of DDPM



achieve improvements over a fulll range of L

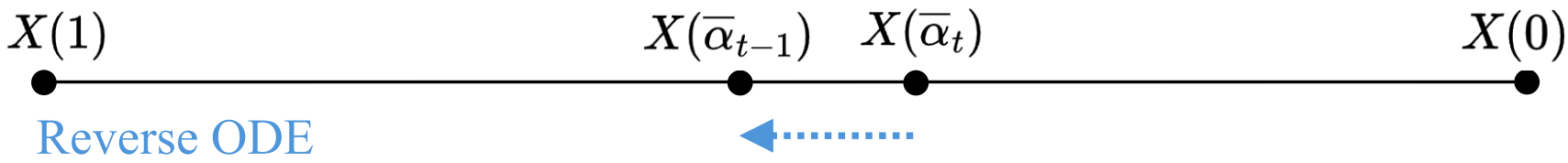
Forward-value discretization of diffusion ODE



$$X(\overline{\alpha}_{t-1}) = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} X(\overline{\alpha}_t) + \sqrt{1-\overline{\alpha}_{t-1}} \int_{\lambda(\overline{\alpha}_{t-1})}^{\lambda(\overline{\alpha}_t)} \mathrm{e}^{\lambda} \underbrace{\mu\big(X(\overline{\alpha}(\lambda)), \overline{\alpha}(\lambda)\big)}_{\text{data predictor}} \mathrm{d}\lambda$$

$$\lambda = \frac{1}{2} \log \frac{\overline{\alpha}}{1-\overline{\alpha}}, \ \mu(x, \overline{\alpha}) = \frac{x + (1-\overline{\alpha}) s_{\overline{\alpha}}(x)}{\sqrt{\overline{\alpha}}}$$

Forward-value discretization of diffusion ODE



$$X(\overline{\alpha}_{t-1}) = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} X(\overline{\alpha}_t) + \sqrt{1-\overline{\alpha}_{t-1}} \int_{\lambda(\overline{\alpha}_{t-1})}^{\lambda(\overline{\alpha}_t)} \mathrm{e}^{\lambda} \underbrace{\mu\big(X(\overline{\alpha}(\lambda)), \overline{\alpha}(\lambda)\big)}_{\text{data predictor}} \mathrm{d}\lambda$$

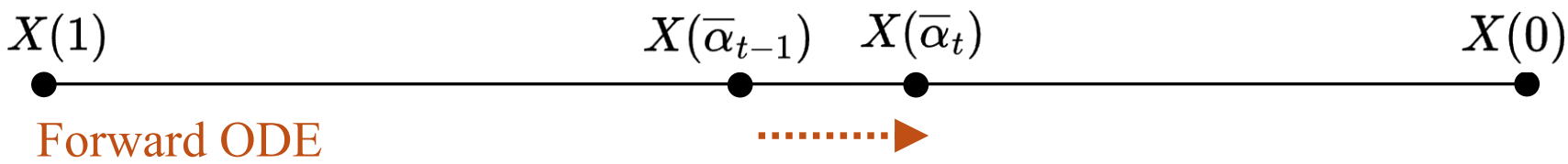
Reverse-value discretization of diffusion ODE:

Approx. integrand $\mu\big(X(\overline{\alpha}(\lambda)), \overline{\alpha}(\lambda)\big)$ with reverse value at $\overline{\alpha}_t$

$$Y_{t-1} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_t, \overline{\alpha}_t\big),$$

$$t = T, \cdots, 2.$$

Forward-value discretization of diffusion ODE

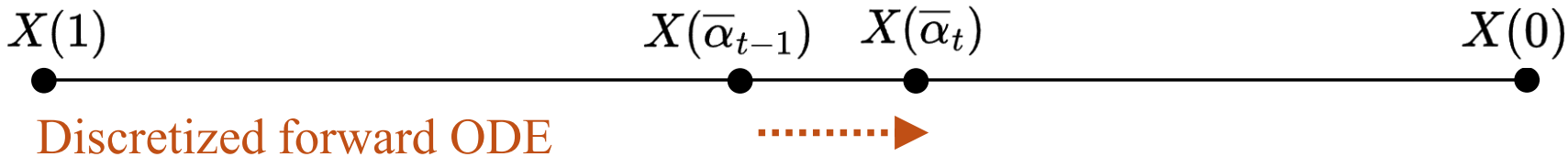


Forward-value discretization of diffusion ODE:

Approx. integrand $\mu\big(X(\overline{\alpha}(\lambda)), \overline{\alpha}(\lambda)\big)$ with forward value at $\overline{\alpha}_{t-1}$

$$Y_t^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_t}{1-\overline{\alpha}_{t-1}}}\, Y_{t-1}^{\mathsf{fw}} - \left(\sqrt{\overline{\alpha}_t} - \sqrt{\frac{(1-\overline{\alpha}_t)\overline{\alpha}_{t-1}}{1-\overline{\alpha}_{t-1}}}\right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big),$$
$$t = T, \cdots, 2.$$

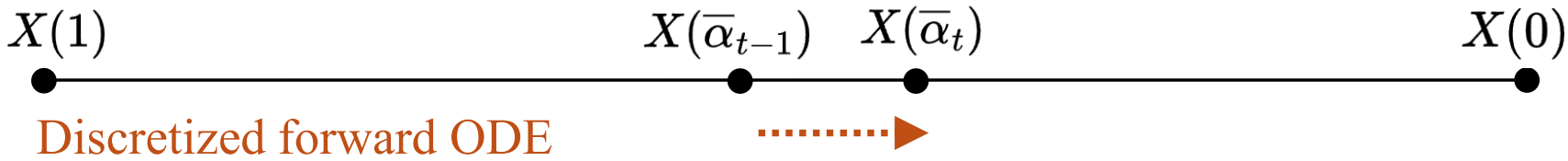
Motivation: diffusion ODE



$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big),$$

$$t = T, \cdots, 2.$$

Motivation: diffusion ODE



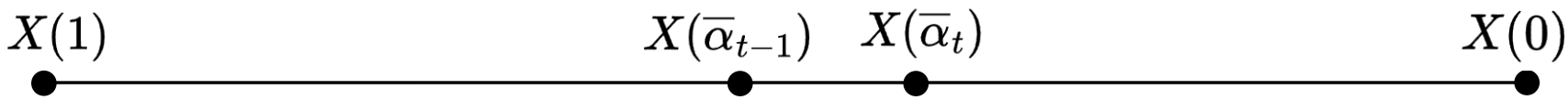
$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big),$$

$$t = T, \cdots, 2.$$

Valid in two extreme cases:

- $T = 2$: $Y_1^{\mathsf{fw}} \approx \mu\big(Y_1^{\mathsf{fw}}, \overline{\alpha}_1\big) \approx X_1$ $(\overline{\alpha}_1 \approx 1,\ \overline{\alpha}_2 \approx 0)$

Motivation: diffusion ODE



Discretized forward ODE

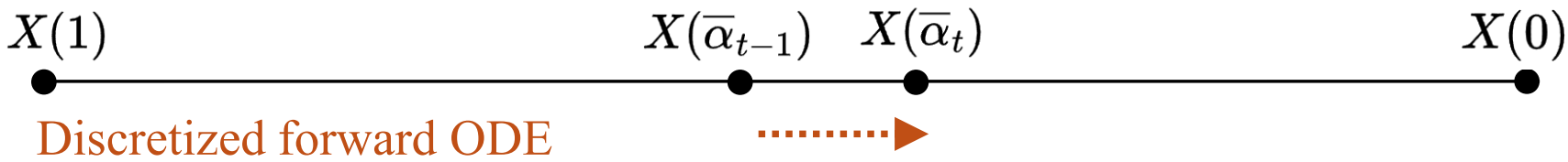
$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big),$$

$$t = T, \cdots, 2.$$

Valid in two extreme cases:

- $T = 2$: $Y_1^{\mathsf{fw}} \approx \mu\big(Y_1^{\mathsf{fw}}, \overline{\alpha}_1\big) \approx X_1$ $(\overline{\alpha}_1 \approx 1,\ \overline{\alpha}_2 \approx 0)$
- $T \to \infty$: $\mu\big(Y^{\mathsf{fw}}, \overline{\alpha}\big) \approx \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big) \Rightarrow$ solve the ODE

Motivation: diffusion ODE



$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big),$$

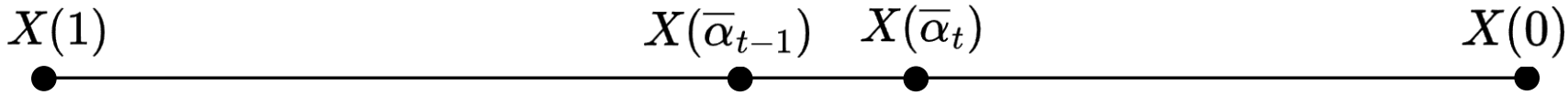
$$t = T, \cdots, 2.$$

Valid in two extreme cases:

- $T = 2$: $Y_1^{\mathsf{fw}} \approx \mu\big(Y_1^{\mathsf{fw}}, \overline{\alpha}_1\big) \approx X_1$ $(\overline{\alpha}_1 \approx 1,\ \overline{\alpha}_2 \approx 0)$
- $T \to \infty$: $\mu\big(Y^{\mathsf{fw}}, \overline{\alpha}\big) \approx \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big) \Rightarrow$ solve the ODE

Conjecture: forward-value discretized ODE $Y_1^{\mathsf{fw}} \overset{\mathsf{d}}{\approx} X(\overline{\alpha}_1)$ for any $T \geq 2$

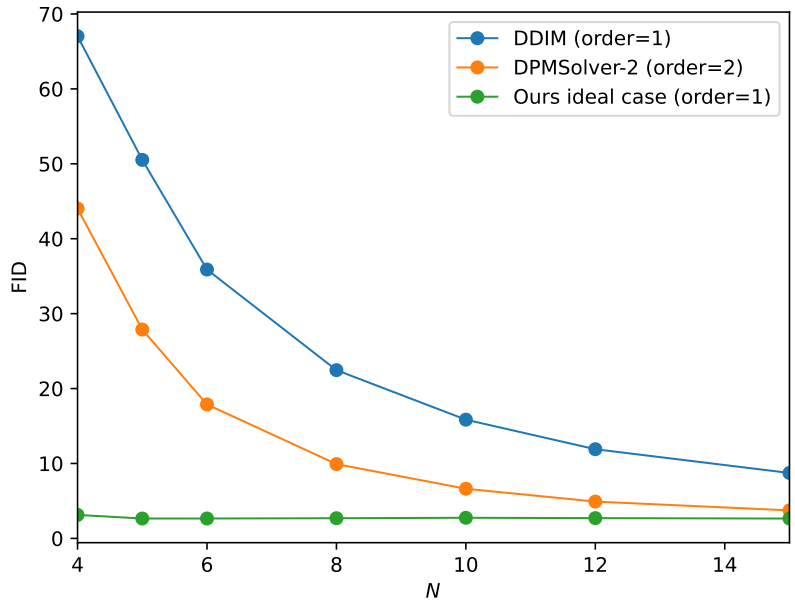
Motivation: diffusion ODE

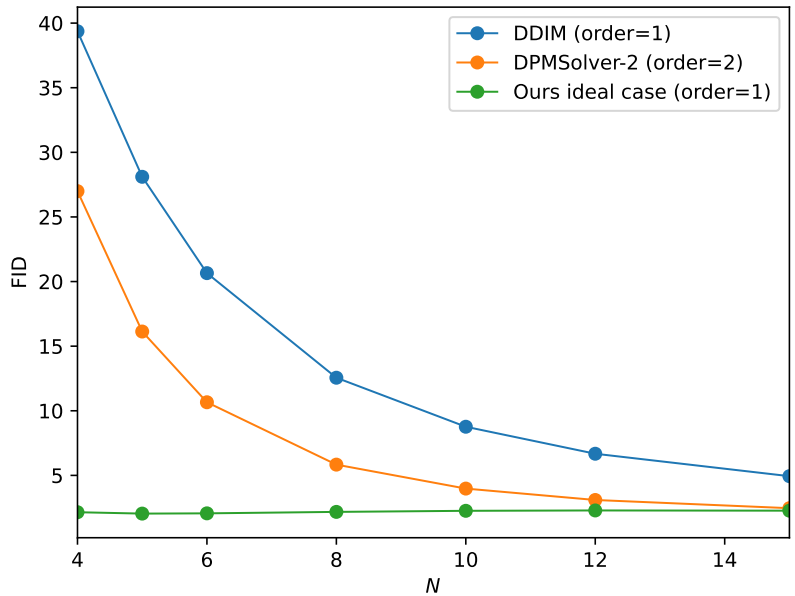


Discretized forward ODE

$$Y_{t-1}^{\text{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\text{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\left(Y_{t-1}^{\text{fw}}, \overline{\alpha}_{t-1}\right),$$

$$t = T, \cdots, 2.$$





Order of our sampler

Recall forward-value discretization of diffusion ODE: for $t = T, \ldots, 2$,

$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big).$$

Theorem 8 (Jiao, Li, Cai, Li. '25)

The forward-value discretization of diffusion ODE satisfies

$$\|Y_1^{\mathsf{fw}} - X(\overline{\alpha}_1) + Y_1^{\mathsf{ddim}} - X(\overline{\alpha}_1)\|_2 = \widetilde{O}(1/T^2).$$

Order of our sampler

Recall forward-value discretization of diffusion ODE: for $t = T, \ldots, 2$,

$$Y_{t-1}^{\mathsf{fw}} = \sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}} Y_t^{\mathsf{fw}} + \left(\sqrt{\overline{\alpha}_{t-1}} - \sqrt{\frac{(1-\overline{\alpha}_{t-1})\overline{\alpha}_t}{1-\overline{\alpha}_t}} \right) \mu\big(Y_{t-1}^{\mathsf{fw}}, \overline{\alpha}_{t-1}\big).$$

Theorem 8 (Jiao, Li, Cai, Li. '25)

The forward-value discretization of diffusion ODE satisfies

$$\|Y_1^{\mathsf{fw}} - X(\overline{\alpha}_1) + Y_1^{\mathsf{ddim}} - X(\overline{\alpha}_1)\|_2 = \widetilde{O}(1/T^2).$$

Our proposed sampler is first-order

(DDIM error $\|Y_1^{\mathsf{ddim}} - X(\overline{\alpha}_1)\|_2 = \widetilde{O}(1/T)$)

Experiments: quantitative comparisons

FIDs↓ on ImageNet64 dataset (EDM2-S):

NFE	First-order Algorithms		High-order Algorithms		
	Ours	DDIM	DPMSolver-2	DPMSolver-3	UniPC-3
4	**22.35**	43.86	29.91	23.66	50.00
5	**11.98**	31.41	18.16	12.57	26.93
6	**7.20**	23.22	11.89	7.63	15.26
8	**3.64**	14.20	6.40	3.92	5.78
10	**2.51**	9.85	4.26	2.70	2.71

Experiments: quantitative comparisons

FIDs↓ on ImageNet64 dataset (EDM2-S):

NFE	First-order Algorithms		High-order Algorithms		
	Ours	DDIM	DPMSolver-2	DPMSolver-3	UniPC-3
4	**22.35**	43.86	29.91	23.66	50.00
5	**11.98**	31.41	18.16	12.57	26.93
6	**7.20**	23.22	11.89	7.63	15.26
8	**3.64**	14.20	6.40	3.92	5.78
10	**2.51**	9.85	4.26	2.70	2.71

- Better than first- and second-order algorithms
- Comparable with third-order algorithms

Experiments: quantitative comparisons

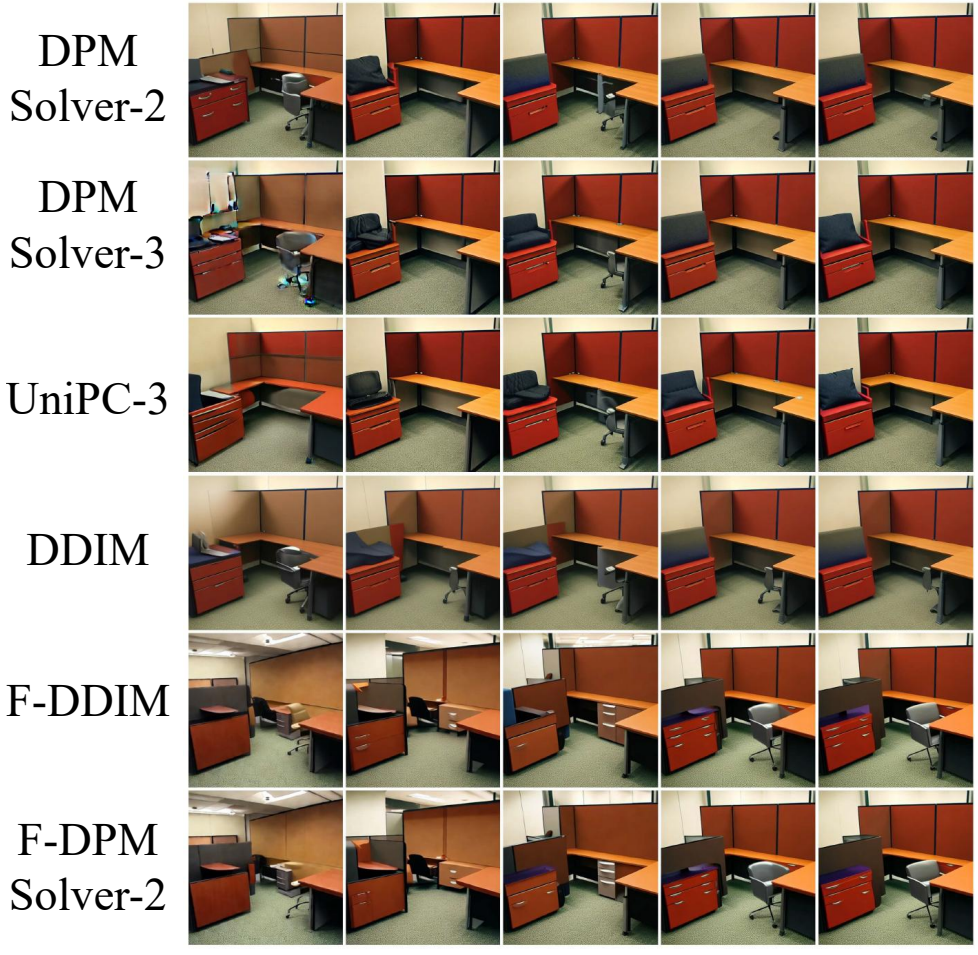
FIDs↓ on ImageNet512 dataset (EDM2-XXL):

NFE	First-order Algorithms		High-order Algorithms		
	Ours	DDIM	DPMSolver-2	DPMSolver-3	UniPC-3
4	**50.96**	87.78	66.21	59.34	217.22
5	**22.69**	61.62	38.02	28.05	94.36
6	**14.33**	53.88	28.23	16.11	30.65
8	**6.18**	31.16	12.18	7.01	8.51
10	**4.57**	20.97	7.44	4.93	4.90

- Better than first- and second-order algorithms
- Comparable with third-order algorithms

Experiments: qualitative comparisons

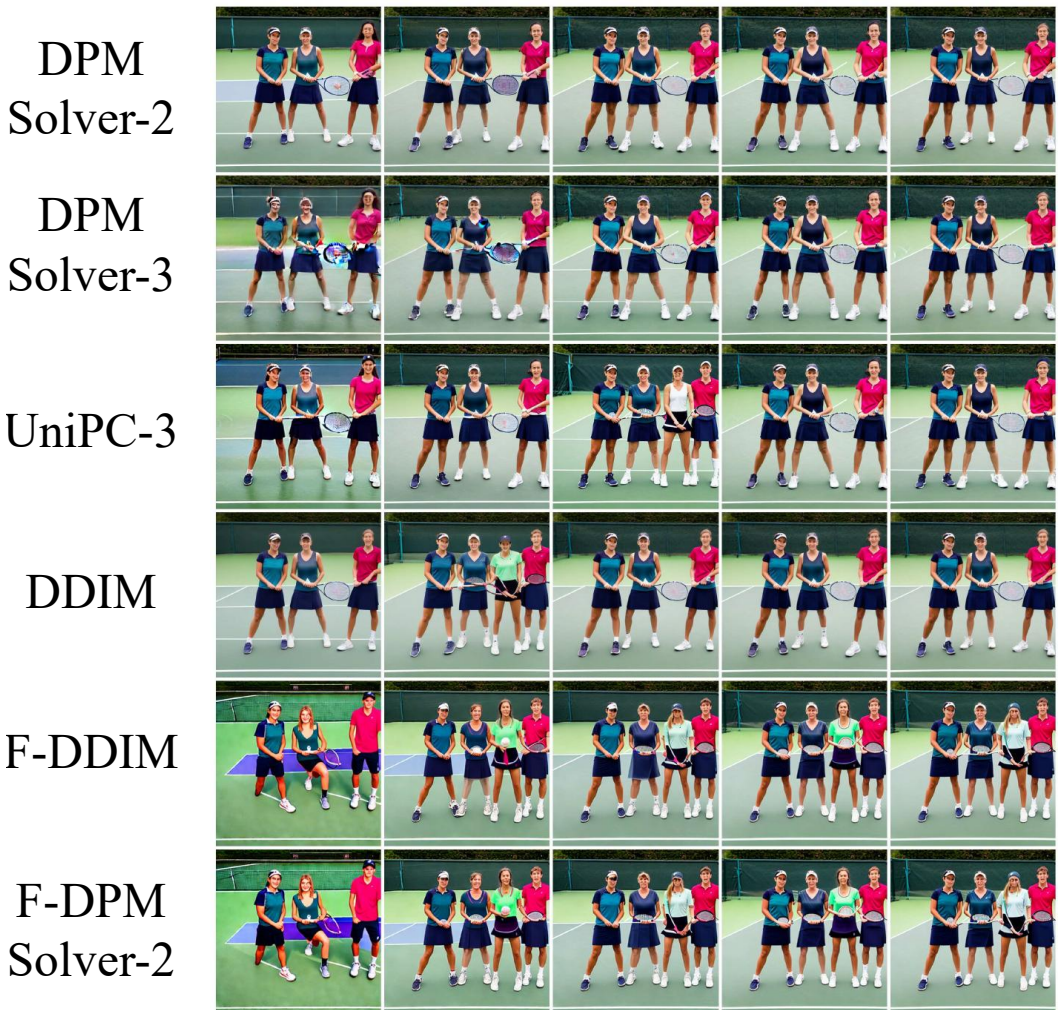
Stable diffusion: a desk and chair in an office cubicle



Only our sampler can generate chair

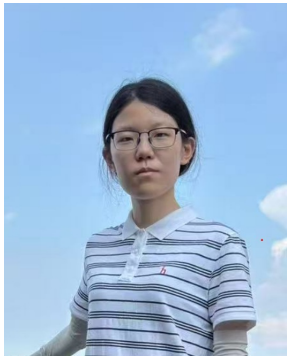
Experiments: qualitative comparisons

Stable diffusion: four tennis players with rackets on a court



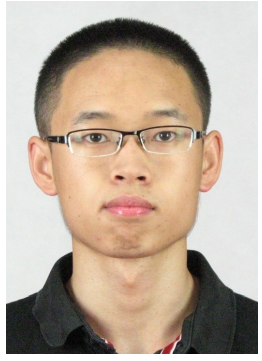
Only our sampler can generate four players

Part 4: provable benefits of diffusion guidance



Yuchen Jiao
CUHK

Yuxin Chen
UPenn



Gen Li
CUHK

Guided/controlled data generation

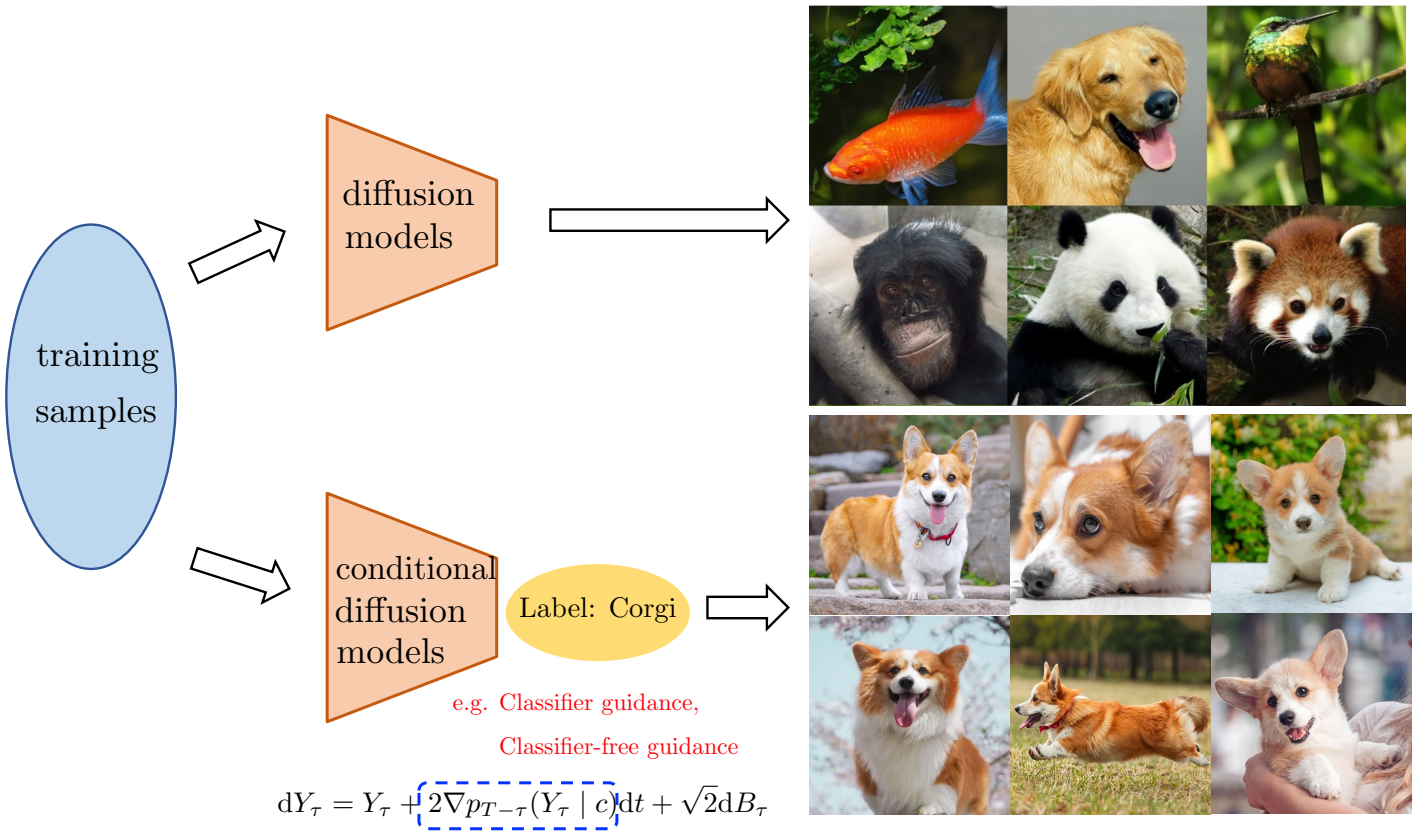
external conditions / specific instructions (via, e.g., prompts)

$\longrightarrow$ steer generation toward specific prompts

Guided/controlled data generation

external conditions / specific instructions (via, e.g., prompts)

$\longrightarrow$ steer generation toward specific prompts



class-conditional sampling: generate samples for a specified class c

sample from $p_{\text{data}|c}$

Conditional diffusion models?

class-conditional sampling: $\underbrace{\text{generate samples for a specified class } c}_{\text{sample from } p_{\text{data}|c}}$

a natural conditioning approach: replace scores w/ cond. scores

Conditional diffusion models?

class-conditional sampling: generate samples for a specified class c

sample from $p_{\text{data}|c}$

a natural conditioning approach: replace scores w/ cond. scores

$$\text{(unguided)} \qquad \mathrm{d}Y_t = \Big(\frac{1}{2}Y_t + \nabla \log p_{X_{1-t}}(Y_t)\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

Conditional diffusion models?

class-conditional sampling: generate samples for a specified class c

sample from $p_{\text{data}|c}$

a natural conditioning approach: replace scores w/ cond. scores

$$\text{(guided)} \qquad \mathrm{d}Y_t = \Big(\frac{1}{2}Y_t + \nabla \log p_{X_{1-t}\,|\,c}(Y_t \,|\, c)\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

Conditional diffusion models?

class-conditional sampling: generate samples for a specified class c

sample from $p_{\text{data}|c}$

a natural conditioning approach: replace scores w/ cond. scores

$$\text{(guided)} \qquad \mathrm{d}Y_t = \Big(\frac{1}{2}Y_t + \nabla \log p_{X_{1-t}\,|\,c}(Y_t \,|\, c)\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- mathematically sound

Conditional diffusion models?

class-conditional sampling: generate samples for a specified class c

sample from $p_{\text{data}|c}$

a natural conditioning approach: replace scores w/ cond. scores

$$\text{(guided)} \qquad \mathrm{d}Y_t = \Big(\frac{1}{2}Y_t + \nabla \log p_{X_{1-t}\,|\,c}(Y_t \,|\, c)\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- mathematically sound
- *unsatisfactory perceptual quality . . .*

sampling from this model via a standard diffusion sampler (e.g. DDPM). Interestingly, this standard way of conditioning usually does not perform well for diffusion models, for reasons that are unclear. In the text-to-image case for example, the generated samples tend to be visually incoherent and not faithful to the prompt, even for large-scale models (Ho and Salimans, 2022; Rombach et al., 2022).

Practically more appealing: diffusion guidance

Add classifier probability to the drift to amplify guidance

$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}\,|\,c}(Y_t^w \,|\, c)\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

Practically more appealing: diffusion guidance

Add classifier probability to the drift to amplify guidance

$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}\,|\,c}(Y_t^w \,|\, c) + \underbrace{w\nabla \log p_{c\,|\,X_{1-t}}(c \,|\, Y_t^w)}_{\text{guidance}}\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

Practically more appealing: diffusion guidance

Add classifier probability to the drift to amplify guidance

$$\mathrm{d}Y_t^w = \left(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}\,|\,c}(Y_t^w \,|\, c) + \underbrace{w\nabla \log p_{c\,|\,X_{1-t}}(c \,|\, Y_t^w)}_{\text{guidance}}\right)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

classifier guidance

CLASSIFIER-FREE DIFFUSION GUIDANCE

Jonathan Ho & Tim Salimans
Google Research, Brain team
{jonathanho,salimans}@google.com

Mystery of diffusion guidance

$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}\,|\,c}(Y_t^w \,|\, c) + \underbrace{w\nabla \log p_{c\,|\,X_{1-t}}(c \,|\, Y_t^w)}_{\text{guidance}}\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- Bayes' rule $\nabla \log p_{X_{1-t}\,|\,c} + w\nabla \log p_{c\,|\,X_{1-t}} = \nabla \log \left(p_{X_{1-t}} \cdot p_{c\,|\,X_{1-t}}^{1+w}\right)$

Mystery of diffusion guidance

$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}\,|\,c}(Y_t^w \,|\, c) + \underbrace{w\nabla \log p_{c\,|\,X_{1-t}}(c \,|\, Y_t^w)}_{\text{guidance}}\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- Bayes' rule $\nabla \log p_{X_{1-t}\,|\,c} + w\nabla \log p_{c\,|\,X_{1-t}} = \nabla \log \left(p_{X_{1-t}} \cdot p_{c\,|\,X_{1-t}}^{1+w}\right)$
- *appears to sample from*

$$p_{X_0} p_{c\,|\,X_0}^{1+w}$$

Mystery of diffusion guidance

- appears to sample from

$$p_{X_0} p_{c\,|\,X_0}^{1+w}$$

- contrasts w/ Bayes

$$p_{X_0\,|\,c} \propto p_{X_0} p_{c\,|\,X_0}$$

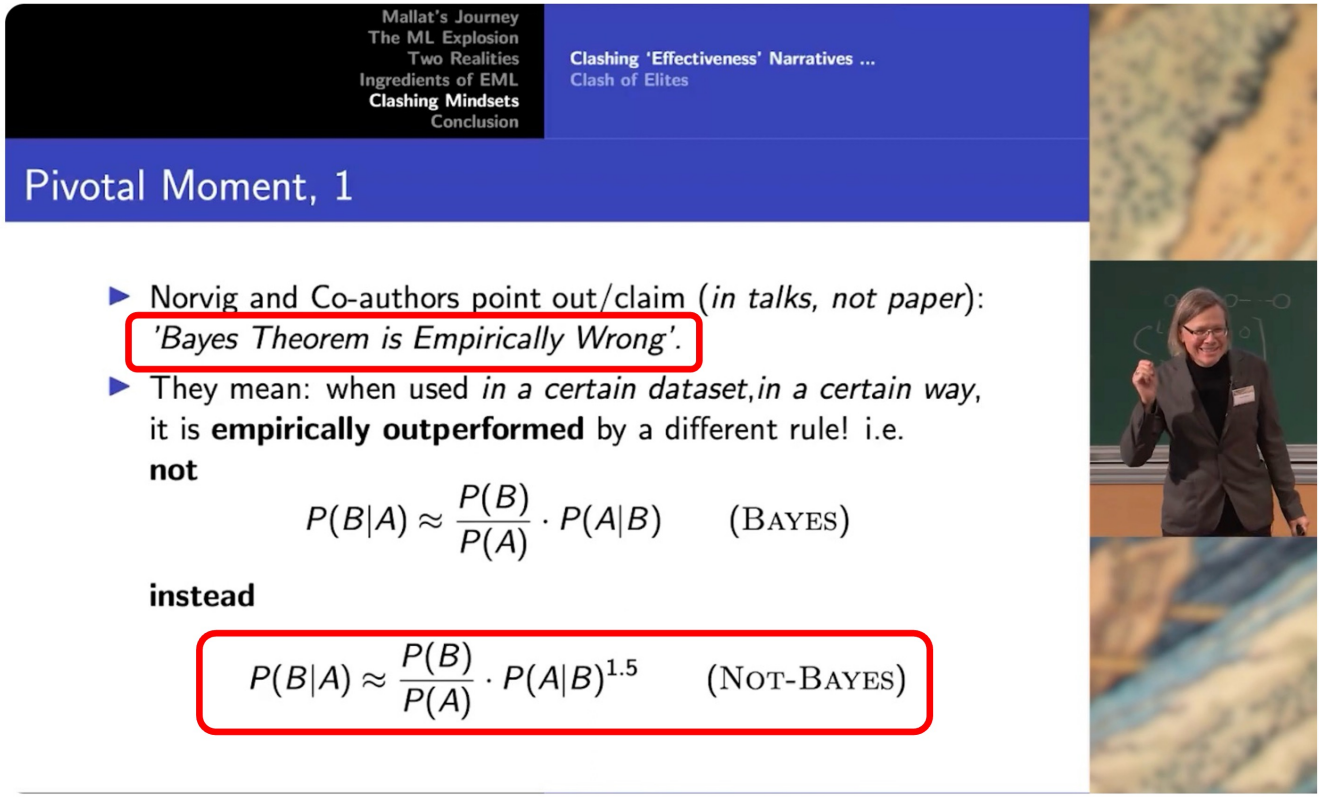
Mystery of diffusion guidance

- appears to sample from

$$p_{X_0} p_{c\,|\,X_0}^{1+w}$$

- contrasts w/ Bayes

$$p_{X_0\,|\,c} \propto p_{X_0} p_{c\,|\,X_0}$$



David Donoho - The Bridge from Mathematical to Digital, and Back

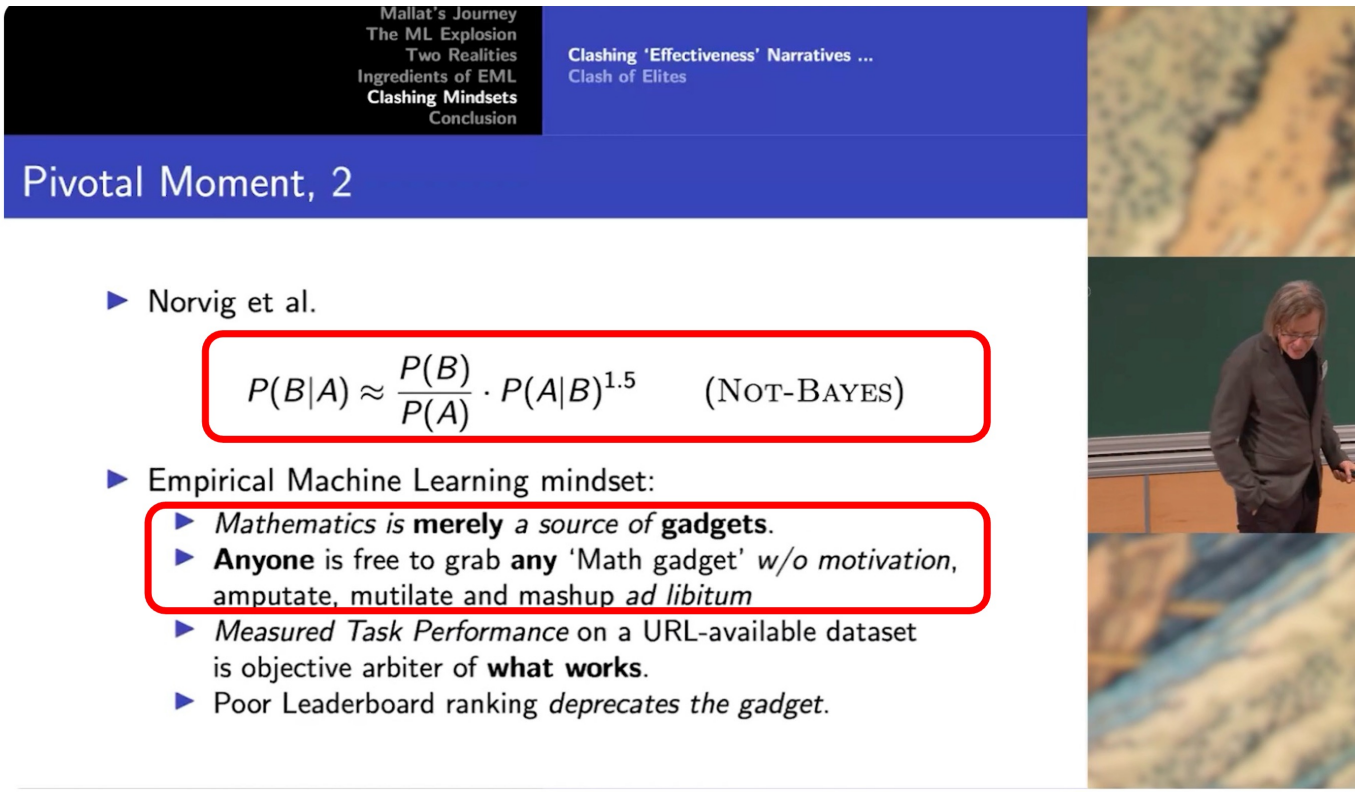
Mystery of diffusion guidance

- appears to sample from

$$p_{X_0} p_{c\,|\,X_0}^{1+w}$$

- contrasts w/ Bayes

$$p_{X_0\,|\,c} \propto p_{X_0} p_{c\,|\,X_0}$$



David Donoho - The Bridge from Mathematical to Digital, and Back

Mystery of diffusion guidance

- appears to sample from

$$p_{X_0} p_{c\,|\,X_0}^{1+w}$$

- contrasts w/ Bayes

$$p_{X_0\,|\,c} \propto p_{X_0} p_{c\,|\,X_0}$$

- *actually, even more complicated than* $p_{X_0} p_{c\,|\,X_0}^{1+w}$ (Bradley et al. '24)

Recent progress

Some recent progress for special distributions

- boosts classification confidence, diminishes diversity in Gaussian mixtures (Wu, Chen, Li, Wang, Wei '24)

Recent progress

Some recent progress for special distributions

- boosts classification confidence, diminishes diversity in Gaussian mixtures (Wu, Chen, Li, Wang, Wei '24)
- samples more heavily from boundary of support of cond. dist (Chidambaram, Gatmiry, Chen, Lee, Lu '24)

Recent progress

Some recent progress for special distributions

- boosts classification confidence, diminishes diversity in Gaussian mixtures (Wu, Chen, Li, Wang, Wei '24)
- samples more heavily from boundary of support of cond. dist (Chidambaram, Gatmiry, Chen, Lee, Lu '24)

Can we clarify benefits of guidance for general distributions?

Our results: effectiveness of diffusion guidance

Theorem 9 (Jiao, Chen, Li '25)

Guided diffusion w/ strength w yields

$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})}\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_0}(c\,|\,Y^{w}_{\text{output}})}\right]}_{\text{guided SDE w/ strength } w}$$

Our results: effectiveness of diffusion guidance

Theorem 9 (Jiao, Chen, Li '25)

Guided diffusion w/ strength w yields

$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})}\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_0}(c\,|\,Y^{w}_{\text{output}})}\right]}_{\text{guided SDE w/ strength } w}$$

- guidance improves $\underbrace{\text{avg. reciprocal of classifier probability}}_{\text{related to Inception Score } \mathbb{E}[\log p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})]}$

Our results: effectiveness of diffusion guidance

Theorem 9 (Jiao, Chen, Li '25)

For any $\underbrace{\delta \in (0,1)}_{\text{early stopping}}$*, guided diffusion w/ strength* w *yields*

$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y_{1-\delta})}\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y^w_{1-\delta})}\right]}_{\text{guided SDE w/ strength } w}$$

- guidance improves $\underbrace{\text{avg. reciprocal of classifier probability}}_{\text{related to Inception Score } \mathbb{E}[\log p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})]}$

Our results: effectiveness of diffusion guidance

Theorem 9 (Jiao, Chen, Li '25)

For any $\underbrace{\delta \in (0,1)}_{\text{early stopping}}$ *and init* y*, guided diffusion w/ strength* w *yields*

$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y_{1-\delta})}\,\middle|\,Y_0 = y\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y^w_{1-\delta})}\,\middle|\,Y^w_0 = y\right]}_{\text{guided SDE w/ strength } w}$$

- guidance improves $\underbrace{\text{avg. reciprocal of classifier probability}}_{\text{related to Inception Score } \mathbb{E}[\log p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})]}$

Our results: effectiveness of diffusion guidance

Theorem 9 (Jiao, Chen, Li '25)

For any $\underbrace{\delta \in (0,1)}_{\text{early stopping}}$ *and init* y*, guided diffusion w/ strength* w *yields*

$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y_{1-\delta})}\,\middle|\,Y_0 = y\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y^w_{1-\delta})}\,\middle|\,Y^w_0 = y\right]}_{\text{guided SDE w/ strength } w}$$

- guidance improves $\underbrace{\text{avg. reciprocal of classifier probability}}_{\text{related to Inception Score } \mathbb{E}[\log p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})]}$
- clarifies (rigorously) which metric guidance can improve

Our results: effectiveness of diffusion guidance

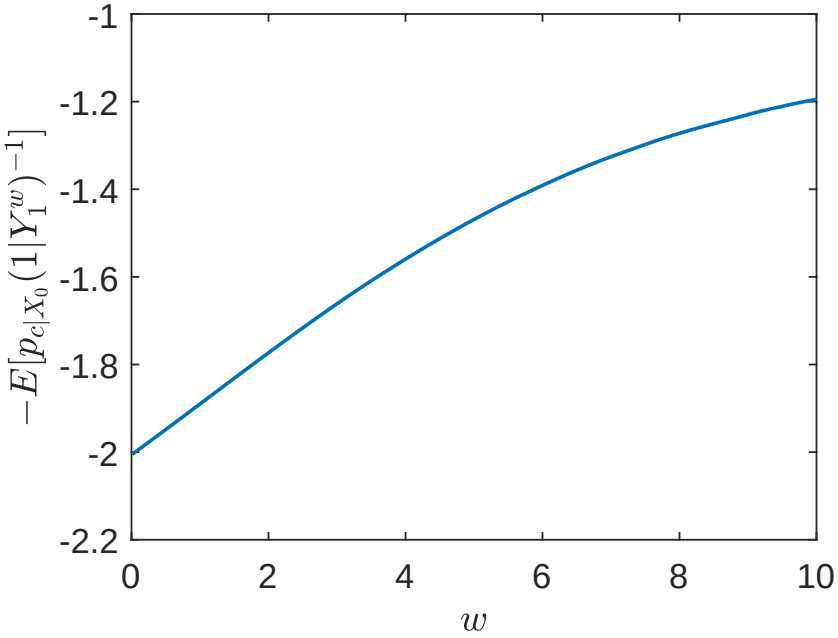
Theorem 9 (Jiao, Chen, Li '25)

For any $\underbrace{\delta \in (0,1)}_{\text{early stopping}}$ *and init* y*, guided diffusion w/ strength* w *yields*

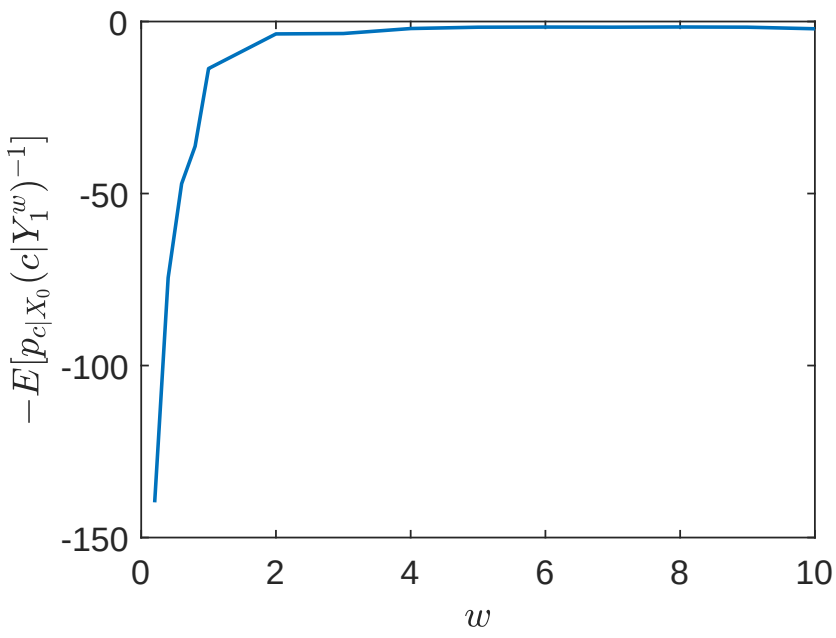
$$\underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y_{1-\delta})}\,\middle|\,Y_0 = y\right]}_{\text{unguided SDE}} \geq \underbrace{\mathbb{E}\left[\frac{1}{p_{c\,|\,X_\delta}(c\,|\,Y^w_{1-\delta})}\,\middle|\,Y^w_0 = y\right]}_{\text{guided SDE w/ strength } w}$$

- guidance improves $\underbrace{\text{avg. reciprocal of classifier probability}}_{\text{related to Inception Score } \mathbb{E}[\log p_{c\,|\,X_0}(c\,|\,Y_{\text{output}})]}$
- clarifies (rigorously) which metric guidance can improve
- holds for most distributions

Experiments



Gaussian Mixture Models



ImageNet

More generally: reward-guided diffusion

improve reward $\mathbb{E}\big[r(Y^{\mathsf{sample}})\big]$ by fine-tuning diffusion model $\mathcal{D}$

More generally: reward-guided diffusion

improve reward $\mathbb{E}\big[r(Y^{\text{sample}})\big]$ by fine-tuning diffusion model $\mathcal{D}$

$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}}(Y_t^w) + w \cdot \text{guidance}\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- guidance term: $\underbrace{\nabla \log p_{X_{1-t}^{r\text{-wt}}}(Y_t^w)}_{\text{reward-reweighted score}} - \nabla \log p_{X_{1-t}}(Y_t^w)$

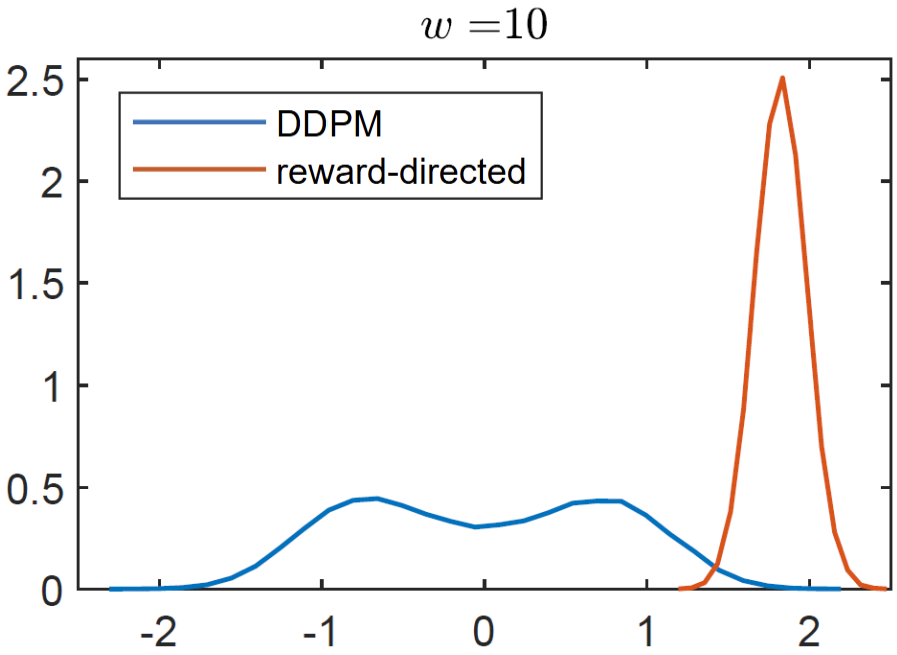
More generally: reward-guided diffusion

improve reward $\mathbb{E}\big[r(Y^{\text{sample}})\big]$ by fine-tuning diffusion model $\mathcal{D}$

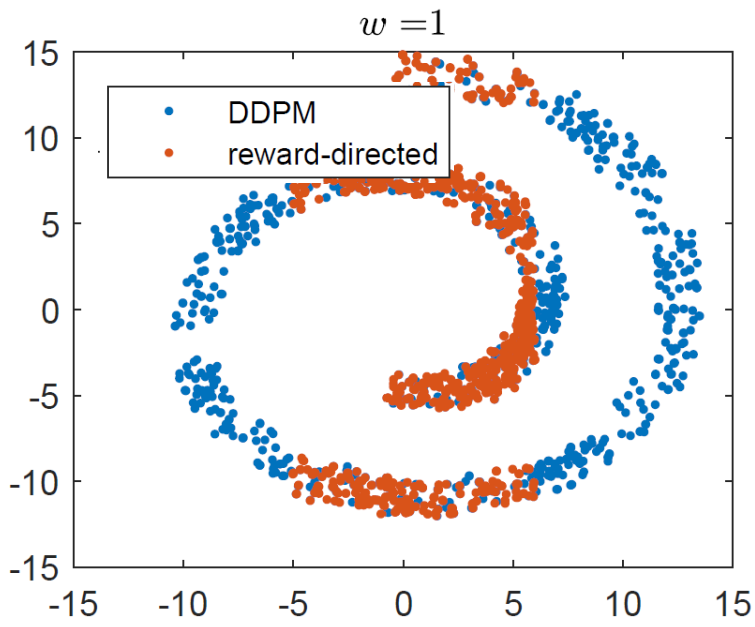
$$\mathrm{d}Y_t^w = \Big(\frac{1}{2}Y_t^w + \nabla \log p_{X_{1-t}}(Y_t^w) + w \cdot \text{guidance}\Big)\frac{\mathrm{d}t}{t} + \frac{1}{\sqrt{t}}\mathrm{d}B_t$$

- guidance term: $\underbrace{\nabla \log p_{X_{1-t}^{r\text{-wt}}}(Y_t^w)}_{\text{reward-reweighted score}} - \nabla \log p_{X_{1-t}}(Y_t^w)$
- supported by theory; easy to train

Experiments

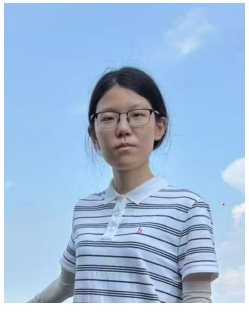


$r(x) = -(x-2)^2$

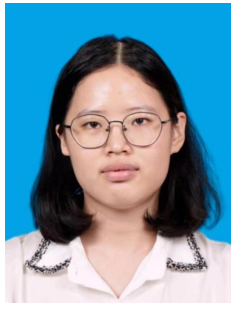


$r(x) = 10\mathbb{1}(x_1 \in [-5, 6])$

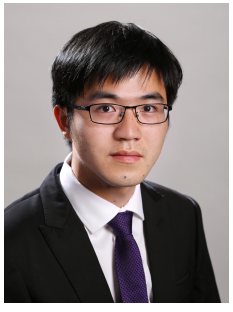
Part 5: diffusion models for inverse problems



Yuchen Jiao
CUHK

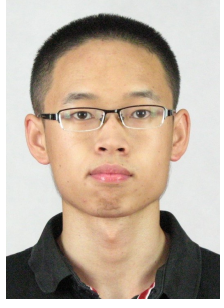


Na Li
ZJU



Changxiao Cai
UMich

Yuxin Chen
UPenn

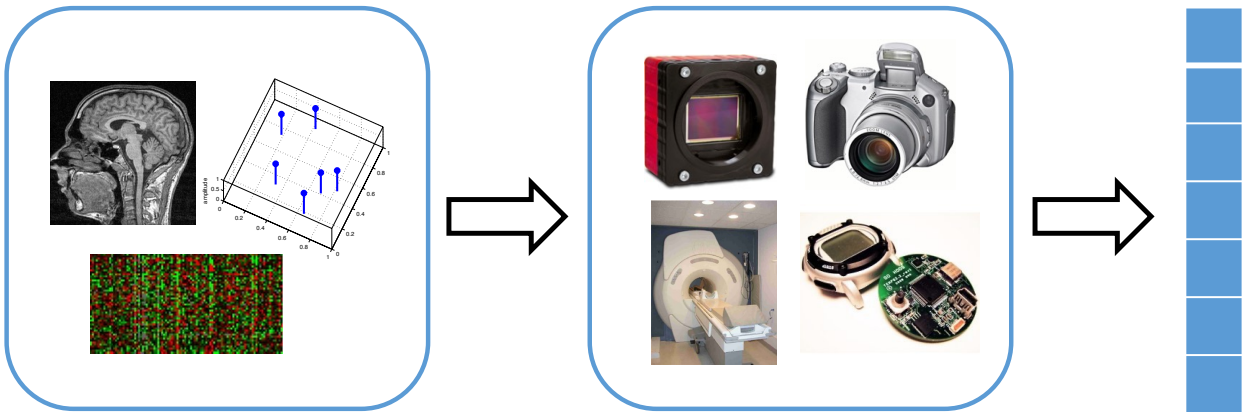


Gen Li
CUHK

Inverse problems

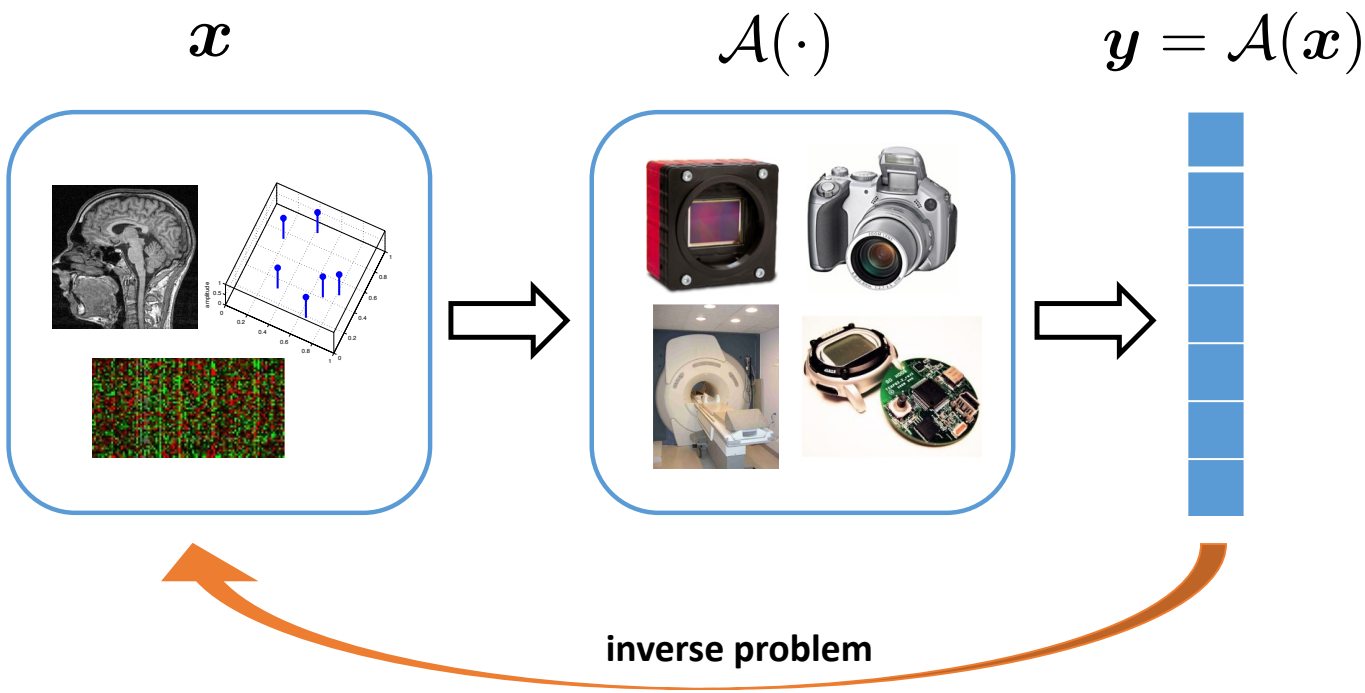
Forward model: we interrogate the signal of interest $\boldsymbol{x}$ through forward model $\mathcal{A}$ and make measurements $\boldsymbol{y}$.

$\boldsymbol{x}$ $\quad$ $\mathcal{A}(\cdot)$ $\quad$ $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x})$



Inverse problems

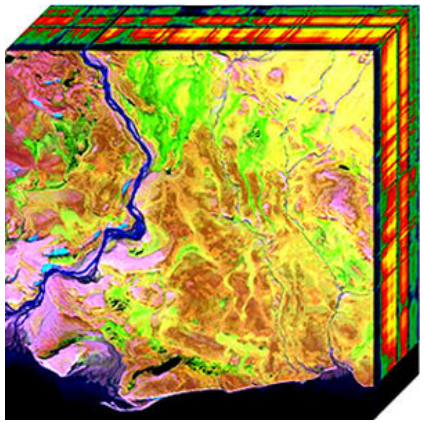
Forward model: we interrogate the signal of interest x through forward model $\mathcal{A}$ and make measurements y.



Inverse problem: recover the signal of interest x from y.

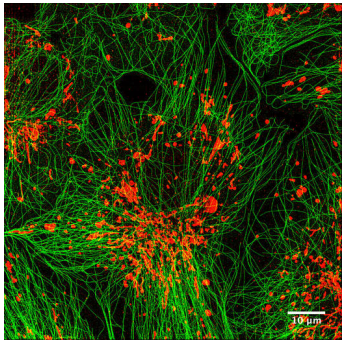
Ubiquitous, but often ill-posed

healthcare



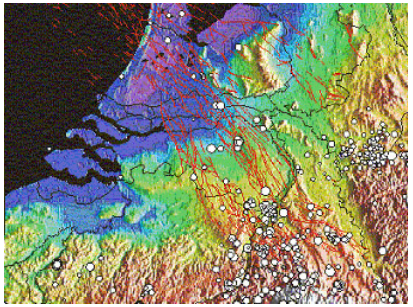
hyperspectral

Internet traffic



microscopy

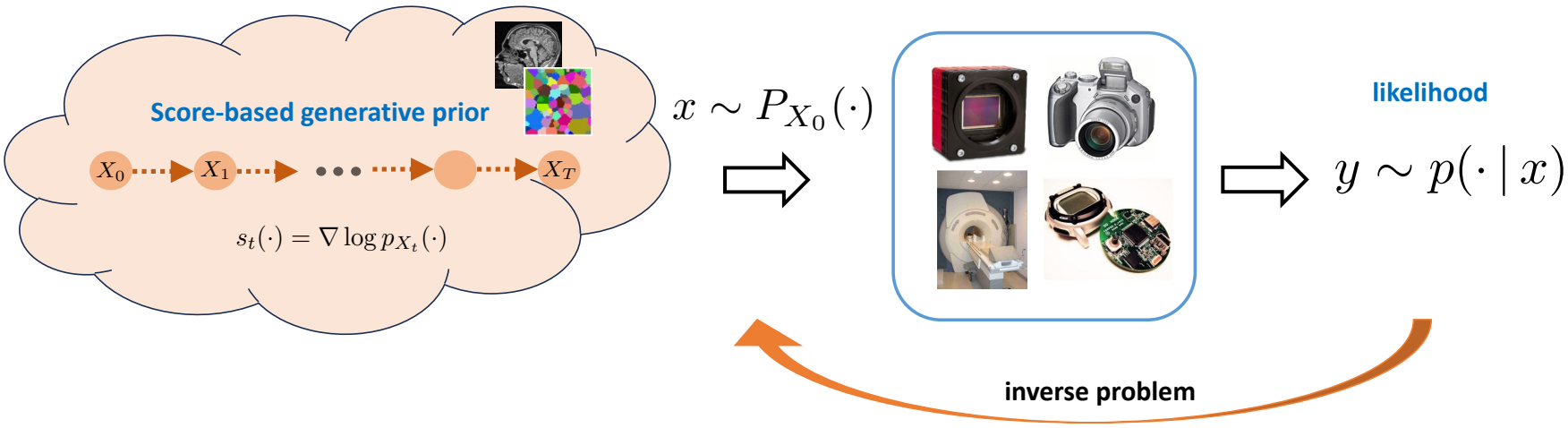
Radio astronomy



seismic imaging

Can we exploit flexible / expressive data priors prescribed by diffusion models?

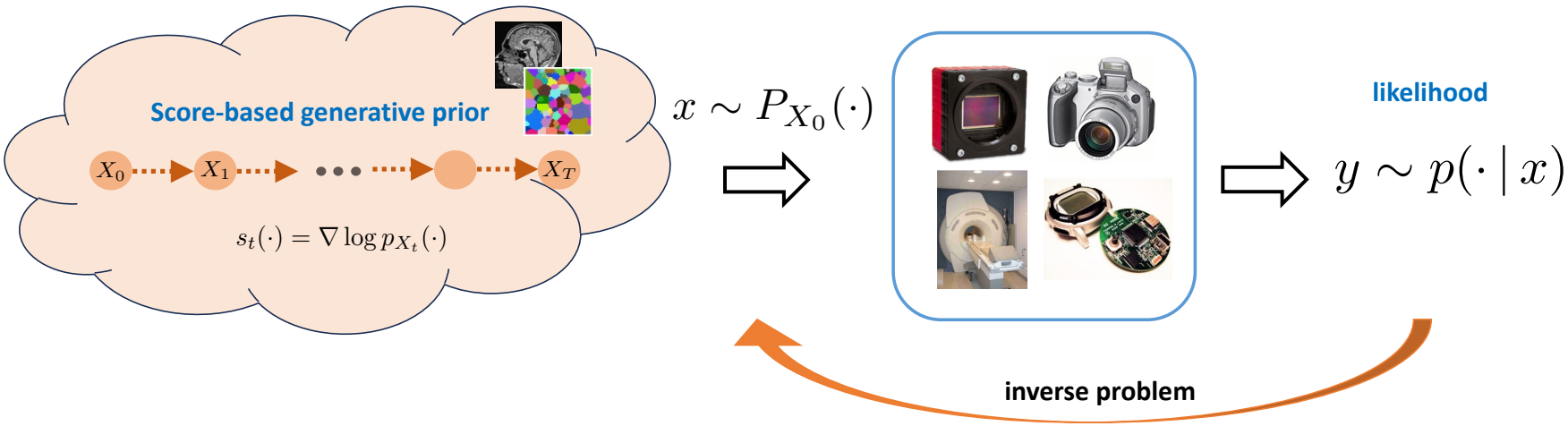
Diffusion model for inverse problems



Posterior sampling: sample from

$$p(\cdot|y) \propto p(\cdot)\, p(y \mid x) = \underbrace{p(\cdot)}_{\texttt{prior}} \exp \underbrace{(\mathcal{L}(\cdot\,;\, y))}_{\texttt{log-likelihood}}$$

Diffusion model for inverse problems



Posterior sampling: sample from

$$p(\cdot|y) \propto p(\cdot)\, p(y \mid x) = \underbrace{p(\cdot)}_{\texttt{prior}} \exp \underbrace{(\mathcal{L}(\cdot\,;\, y))}_{\texttt{log-likelihood}}$$

Score-based implicit prior: the data prior $p(\cdot)$ is accessed through its score functions $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$.

Recall: Tweedie's formula

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

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

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

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

Recall: Tweedie's formula

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

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

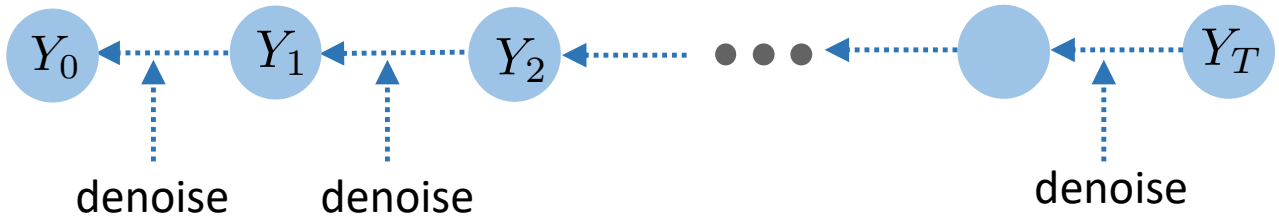
$$s_t^\star(x) = \frac{1}{\sqrt{1-\overline{\alpha}_t}} \mathbb{E}\left[W \;\middle|\; \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t} W = x\right],$$

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

Data predictor (considered in inverse problems):

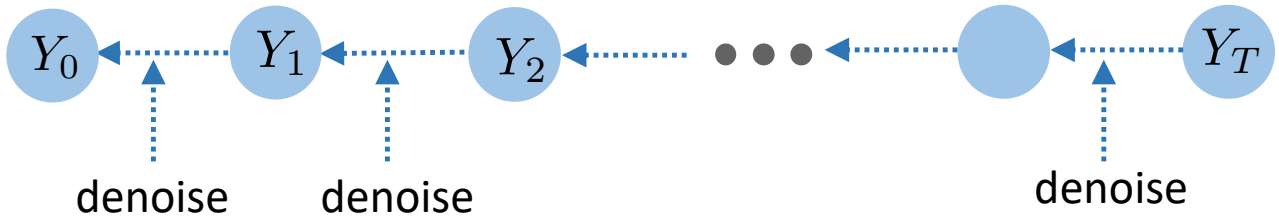
$$\begin{aligned}\mu_t(x) &:= \mathbb{E}\left[X_0 \;\middle|\; \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t} W = x\right] \\ &= \frac{1}{\sqrt{\overline{\alpha}_t}}\left(x - (1-\overline{\alpha}_t) s_t^\star(x)\right)\end{aligned}$$

Recall: DDIM-type Sampler



— *Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole '20*

Recall: DDIM-type Sampler

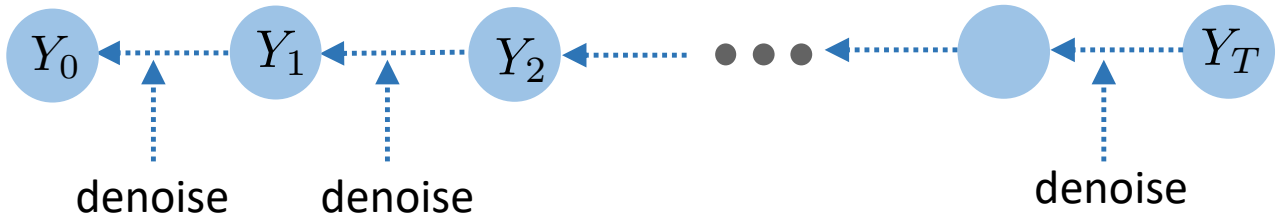


— *Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole '20*

$$\frac{\widehat{X}_{t-1}}{\sigma_{t-1}} = \frac{\widehat{X}_t}{\sigma_t}\sqrt{1-\eta\,(1-e^{-2\delta_{t-1}})} + e^{\lambda_{t-1}}\left(1-e^{-\delta_{t-1}}\sqrt{1-\eta\,(1-e^{-2\delta_{t-1}})}\right)$$
$$\cdot\,\mu_t(\widehat{X}_t) + \sqrt{\eta\,(1-e^{-2\delta_{t-1}})}\,\mathcal{N}(0, I_d), \qquad 0 \le \eta \le 1$$

- $\delta_{t-1} := \lambda_{t-1} - \lambda_t$ where $\lambda_t := \log(\alpha_t/\sigma_t)$ denotes signal-to-noise ratio

Recall: DDIM-type Sampler



— *Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole '20*

$$\frac{\widehat{X}_{t-1}}{\sigma_{t-1}} = \frac{\widehat{X}_t}{\sigma_t}\sqrt{1-\eta\,(1-e^{-2\delta_{t-1}})} + e^{\lambda_{t-1}}\left(1-e^{-\delta_{t-1}}\sqrt{1-\eta\,(1-e^{-2\delta_{t-1}})}\right)$$
$$\cdot\,\mu_t(\widehat{X}_t) + \sqrt{\eta\,(1-e^{-2\delta_{t-1}})}\,\mathcal{N}(0, I_d), \qquad 0 \le \eta \le 1$$

- $\delta_{t-1} := \lambda_{t-1} - \lambda_t$ where $\lambda_t := \log(\alpha_t/\sigma_t)$ denotes signal-to-noise ratio
- Common choices with $\eta \in [0, 1]$
  - $\eta = 1$: stochastic sampler (DDPM, Ho, Jain, Abbeel '20)
  - $\eta = 0$: deterministic sampler (probability flow ODE, Song, Meng, Ermon '20)

Denoising with linear observations

Setting: linear inverse problem:

$$y = Ax + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Goal: sample from posterior $p_{X_0|Y=y}$

Denoising with linear observations

Setting: linear inverse problem:

$$y = Ax + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Goal: sample from posterior $p_{X_0|Y=y}$

Basic idea: replace μ_t with $\mu_{t,y}$ in a DDIM-type sampler

$$\mu_{t,y}(x) := \mathbb{E}[X_0 \mid X_t = x,\ AX_0 + \varepsilon = y]$$

immediate trial: $\min_{\mu} \ \|\mu - \mu_t(x)\|_2^2 + \gamma \|y - A\mu\|_2^2.$

Prior algorithms

- Diffusion Posterior Sampling (DPS, Chung et al'22)
- Denoising Diffusion Restoration Models (DDRM, Kawar et al'22)
- Regularization by Denoising Diffusion Process (RED-diff, Mardani et al'23)
- Denoising Diffusion Null-space Model (DDNM/DDNM+, Wang et al'22)
- Projected Diffusion (ProjDiff, Zhang et al'24)

Prior algorithms

- Diffusion Posterior Sampling (DPS, Chung et al'22)
- Denoising Diffusion Restoration Models (DDRM, Kawar et al'22)
- Regularization by Denoising Diffusion Process (RED-diff, Mardani et al'23)
- Denoising Diffusion Null-space Model (DDNM/DDNM+, Wang et al'22)
- Projected Diffusion (ProjDiff, Zhang et al'24)

Limitations

- implementation-complicated
- lacks theoretical guarantees
- unsatisfactory performance

Observation: two sources of uncertainty

1. Measurement uncertainty: Apply SVD to $A = U\Sigma V^\top \in \mathbb{R}^{k\times d}$:

$$y = Ax + \varepsilon \iff \Sigma_{ss}^{-1} u_s^\top y = v_s^\top x_0 + \mathcal{N}\left(0, \frac{\sigma^2}{\Sigma_{ss}^2}\right), \quad \Sigma_{ss} > 0$$

2. Diffusion uncertainty

$$v_s^\top X_t = \alpha_t v_s^\top X_0 + \sigma_t \mathcal{N}(0,1)$$

Observation: two sources of uncertainty

1. Measurement uncertainty: Apply SVD to $A = U\Sigma V^\top \in \mathbb{R}^{k\times d}$:

$$y = Ax + \varepsilon \iff \Sigma_{ss}^{-1} u_s^\top y = v_s^\top x_0 + \mathcal{N}\left(0, \frac{\sigma^2}{\Sigma_{ss}^2}\right), \quad \Sigma_{ss} > 0$$

2. Diffusion uncertainty

$$v_s^\top X_t = \alpha_t v_s^\top X_0 + \sigma_t \mathcal{N}(0,1)$$

Key insight: divide based on SNR

Direction sets:

$$\mathcal{S}_t := \{\, s : \alpha_t \sigma < \Sigma_{ss}\sigma_t \,\} \quad \text{(observation-dominated at time } t\text{)}$$
$$\mathcal{S}_t^c := \mathcal{S} \setminus \mathcal{S}_t \quad \text{(prior-dominated at time } t\text{)}$$
$$\mathcal{S}^c := [k] \setminus \mathcal{S} \quad \text{(no observations)}$$

Define critical time τ_s s.t. $\alpha_{\tau_s}/\sigma_{\tau_s} = \Sigma_{ss}/\sigma$.

Our sampler

$$\frac{v_s^\top \widehat{X}_{t-1}}{\sigma_{t-1}} = \frac{v_s^\top \widehat{X}_t}{\sigma_t}\sqrt{1-\eta(1-e^{-2\delta_{t-1}})} + e^{\lambda_{t-1}}\left(1-e^{-\delta_{t-1}}\sqrt{1-\eta(1-e^{-2\delta_{t-1}})}\right)$$
$$\cdot\, v_s^\top \mu_t(\widehat{X}_t) + \sqrt{\eta(1-e^{-2\delta_{t-1}})}\, v_s^\top \mathcal{N}(0, I_d)$$

- **Observation-dominated** ($s \in \mathcal{S}_t$): apply DDIM-type update with $\eta = 0$ to the modified forward process

$$\xi_{t,s} := \alpha_t \xi_{0,s} + \sqrt{\sigma_t^2 - \alpha_t^2 \sigma^2 \Sigma_{ss}^{-2}}\,\mathcal{N}(0,1), \quad \xi_{0,s} := \Sigma_{ss}^{-1} u_s^\top y$$

- **Prior-dominated directions** ($s \in \mathcal{S}_t^c$): apply DDIM-type update with $\eta = 1$ (corresponding to DDPM)
- **Unobserved directions** ($s \in \mathcal{S}^{\mathrm{c}}$): apply DDIM-type update with sufficiently large $\eta = \eta_{\mathcal{S}^{\mathrm{c}}}$.

Theoretical guarantee

Theorem 10 (Jiao, Na, Cai, Chen, Li '26)

Assume X_0 *has bounded support. Let* $\delta = \max \delta_t$. *If* $\eta_{\mathcal{S}^c}^3 \delta^2 \to 0$ *and* $\eta_{\mathcal{S}^c}^2 \delta / \log \frac{1}{\delta} \to \infty$, *then*

$$\mathsf{law}(\widehat{X}_t) \to \mathsf{law}(X_0 \mid Y = y) \quad \textit{as} \quad \lambda_t \to \infty.$$

Theoretical guarantee

Theorem 10 (Jiao, Na, Cai, Chen, Li '26)

Assume X_0 *has bounded support. Let* $\delta = \max \delta_t$. *If* $\eta_{\mathcal{S}^c}^3 \delta^2 \to 0$ *and* $\eta_{\mathcal{S}^c}^2 \delta / \log \frac{1}{\delta} \to \infty$, *then*

$$\mathsf{law}(\widehat{X}_t) \to \mathsf{law}(X_0 \mid Y = y) \quad \textit{as} \quad \lambda_t \to \infty.$$

- provably solving inverse problems

Theoretical guarantee

Theorem 10 (Jiao, Na, Cai, Chen, Li '26)

Assume X_0 *has bounded support. Let* $\delta = \max \delta_t$. *If* $\eta_{\mathcal{S}^c}^3 \delta^2 \to 0$ *and* $\eta_{\mathcal{S}^c}^2 \delta / \log \frac{1}{\delta} \to \infty$, *then*

$$\mathsf{law}(\widehat{X}_t) \to \mathsf{law}(X_0 \mid Y = y) \quad \textit{as} \quad \lambda_t \to \infty.$$

- provably solving inverse problems
- providing guidance for parameter choices η

Experiments: Choices of η

Inpainting task

$\eta_{\mathcal{S}_t^c}$	$\eta_{\mathcal{S}^c}$	CelebA				ImageNet			
		PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
0	0	27.22	0.79	27.11	61.46	21.24	0.54	46.72	89.56
	1	31.32	0.88	16.97	30.24	26.13	0.78	26.17	37.76
	2	32.55	0.90	15.44	26.62	28.76	0.86	16.74	18.60
	4	33.16	**0.91**	15.07	25.42	30.72	**0.89**	14.45	**15.74**
	8	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.86
	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
	32	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0.5		33.03	0.90	15.66	26.60	30.86	0.88	15.33	17.89
1		32.84	0.90	16.36	28.42	30.67	0.88	16.61	20.85
2		32.51	0.89	17.89	33.21	30.26	0.86	19.52	28.71

Experiments: Choices of η

Inpainting task

$\eta_{\mathcal{S}_t^c}$	$\eta_{\mathcal{S}^c}$	CelebA				ImageNet			
		PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
0	0	27.22	0.79	27.11	61.46	21.24	0.54	46.72	89.56
	1	31.32	0.88	16.97	30.24	26.13	0.78	26.17	37.76
	2	32.55	0.90	15.44	26.62	28.76	0.86	16.74	18.60
	4	33.16	**0.91**	15.07	25.42	30.72	**0.89**	14.45	**15.74**
	8	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.86
	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
	32	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0.5		33.03	0.90	15.66	26.60	30.86	0.88	15.33	17.89
1		32.84	0.90	16.36	28.42	30.67	0.88	16.61	20.85
2		32.51	0.89	17.89	33.21	30.26	0.86	19.52	28.71

- Larger $\eta_{\mathcal{S}^c}$ improves performance, aligning with our theory

Experiments: Choices of η

Inpainting task

$\eta_{\mathcal{S}_t^c}$	$\eta_{\mathcal{S}^c}$	CelebA				ImageNet			
		PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
0	0	27.22	0.79	27.11	61.46	21.24	0.54	46.72	89.56
	1	31.32	0.88	16.97	30.24	26.13	0.78	26.17	37.76
	2	32.55	0.90	15.44	26.62	28.76	0.86	16.74	18.60
	4	33.16	**0.91**	15.07	25.42	30.72	**0.89**	14.45	**15.74**
	8	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.86
	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
	32	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0	16	**33.23**	**0.91**	**15.06**	**25.33**	**31.06**	**0.89**	**14.31**	15.85
0.5		33.03	0.90	15.66	26.60	30.86	0.88	15.33	17.89
1		32.84	0.90	16.36	28.42	30.67	0.88	16.61	20.85
2		32.51	0.89	17.89	33.21	30.26	0.86	19.52	28.71

- Larger $\eta_{\mathcal{S}^c}$ improves performance, aligning with our theory
- $\eta_{\mathcal{S}_t^c} = 0$ achieves the best performance

Experiments: Comparison with references

Super-resolution task

	Method	CelebA				ImageNet			
		PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Single-sample	$A^\dagger y$	23.64	0.49	68.72	147.89	21.85	0.45	65.34	183.32
	DPS	27.98	0.78	23.10	39.91	24.44	0.67	31.81	**36.17**
	DDRM	29.20	0.82	21.92	40.14	25.66	0.72	34.88	55.71
	RED-diff	24.98	0.55	50.59	73.89	22.74	0.49	53.24	96.26
	DDNM+	29.20	0.82	21.91	39.96	25.62	0.72	34.39	53.78
	ProjDiff	29.49	0.83	**20.89**	36.61	25.73	0.72	33.03	49.70
	Ours	29.84	0.84	22.02	**34.16**	25.90	0.73	**32.84**	49.99
Posterior mean	DDNM+	30.22	**0.85**	21.97	43.54	26.11	0.73	34.38	54.27
	ProjDiff	30.31	**0.85**	22.55	38.34	26.10	**0.74**	33.13	50.27
	Ours	**30.35**	**0.85**	22.55	40.62	**26.13**	**0.74**	33.35	52.91

Experiments: Comparison with references

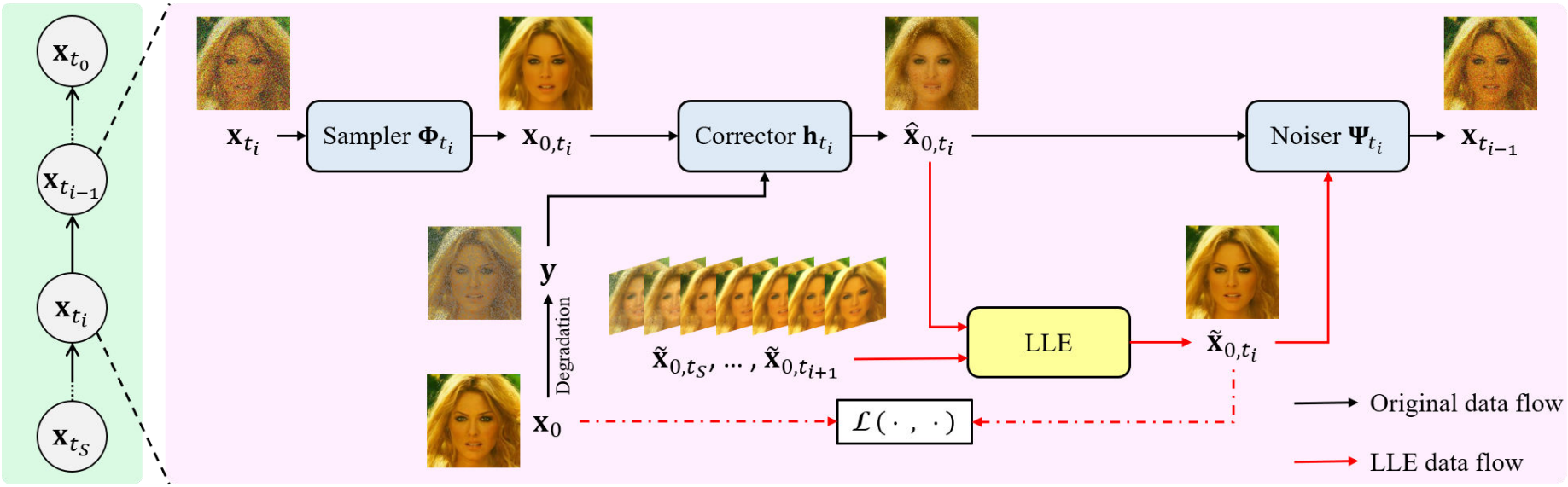
Super-resolution task

	Method	CelebA				ImageNet			
		PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Single-sample	$A^\dagger y$	23.64	0.49	68.72	147.89	21.85	0.45	65.34	183.32
	DPS	27.98	0.78	23.10	39.91	24.44	0.67	31.81	**36.17**
	DDRM	29.20	0.82	21.92	40.14	25.66	0.72	34.88	55.71
	RED-diff	24.98	0.55	50.59	73.89	22.74	0.49	53.24	96.26
	DDNM+	29.20	0.82	21.91	39.96	25.62	0.72	34.39	53.78
	ProjDiff	29.49	0.83	**20.89**	36.61	25.73	0.72	33.03	49.70
	Ours	29.84	0.84	22.02	**34.16**	25.90	0.73	**32.84**	49.99
Posterior mean	DDNM+	30.22	**0.85**	21.97	43.54	26.11	0.73	34.38	54.27
	ProjDiff	30.31	**0.85**	22.55	38.34	26.10	**0.74**	33.13	50.27
	Ours	**30.35**	**0.85**	22.55	40.62	**26.13**	**0.74**	33.35	52.91

- Ours: best on three metrics
- Others: best on at most one or two metrics

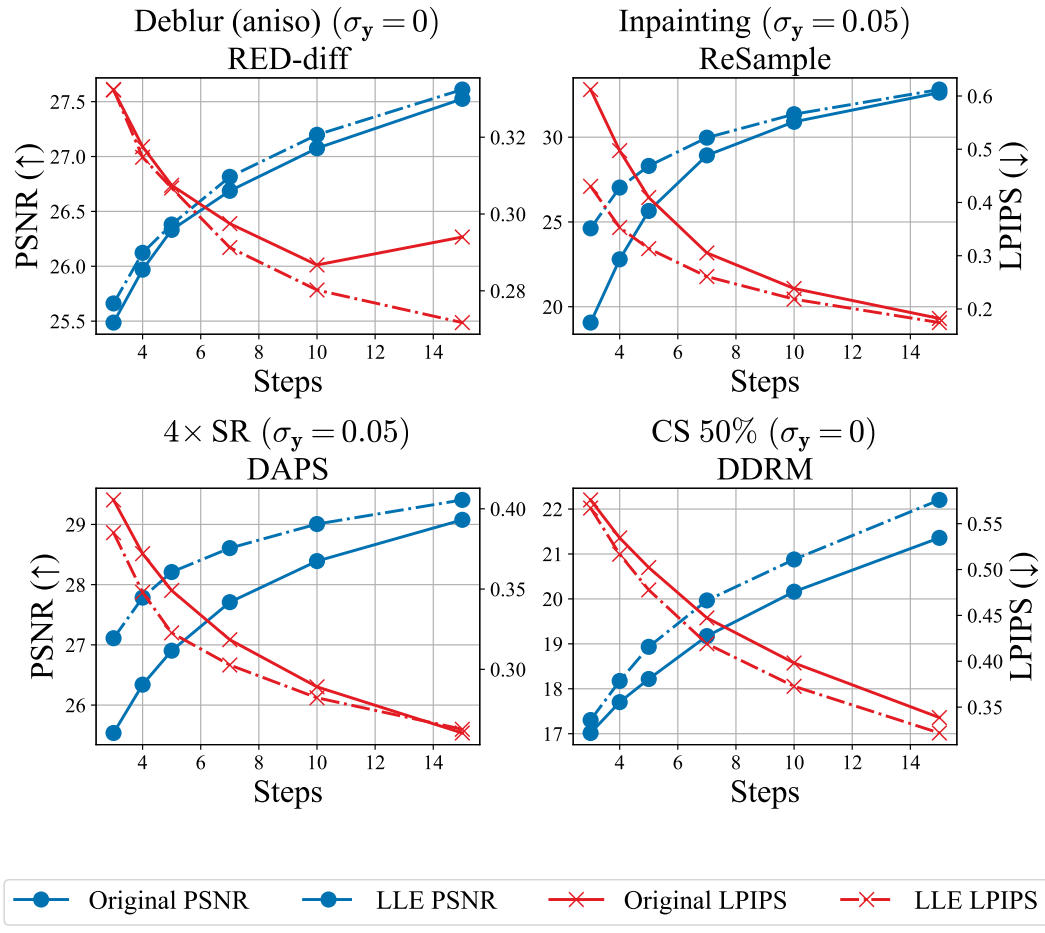
Accelerated diffusion-based inverse algorithms

Inspired by accelerated diffusion sampler



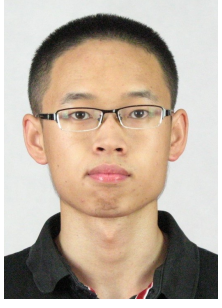
The proposed canonical form of diffusion-based inverse algorithms and the workflow of our Learnable Linear Extrapolation (LLE) method

Numerical experiments

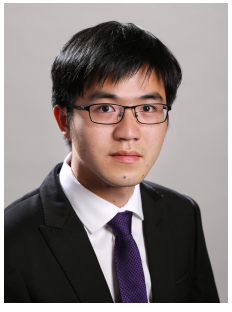


LLE achieves improvements across multiple tasks consistently

Part 6: discrete diffusion (diffusion language models)



Gen Li
CUHK



Changxiao Cai
UMich

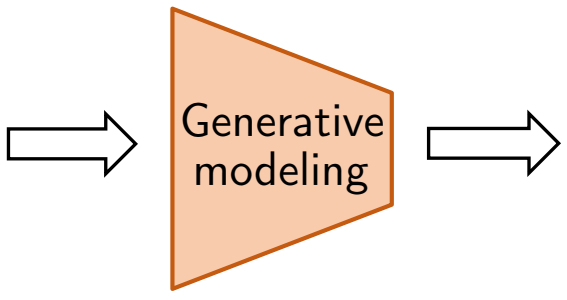
Daniil Dmitriev
UPenn

Zhihan Huang
UPenn

Yuting Wei
UPenn

Language generative models

training data



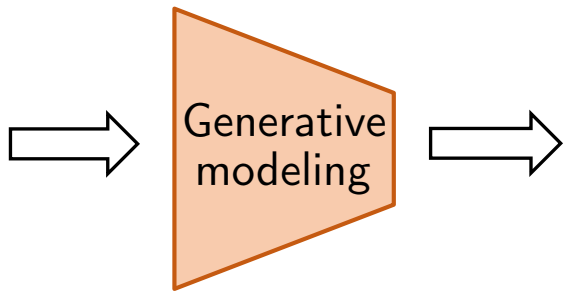
new samples

How to deal with terrible twos?

- Drink a coffee before engaging
- Lower your expectations
- Always carry snacks
- Pick your battles
- Master distraction
- Accept that logic is useless

Language generative models

training data



new samples

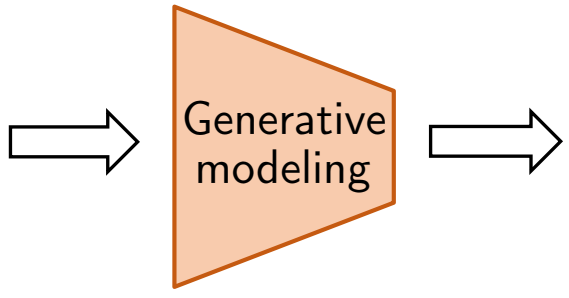
How to deal with terrible twos?

- Drink a coffee before engaging
- Lower your expectations
- Always carry snacks
- Pick your battles
- Master distraction
- Accept that logic is useless

- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a discrete distribution}}$ $(1 \leq i \leq N)$ in $[S]^d$

Language generative models

training data



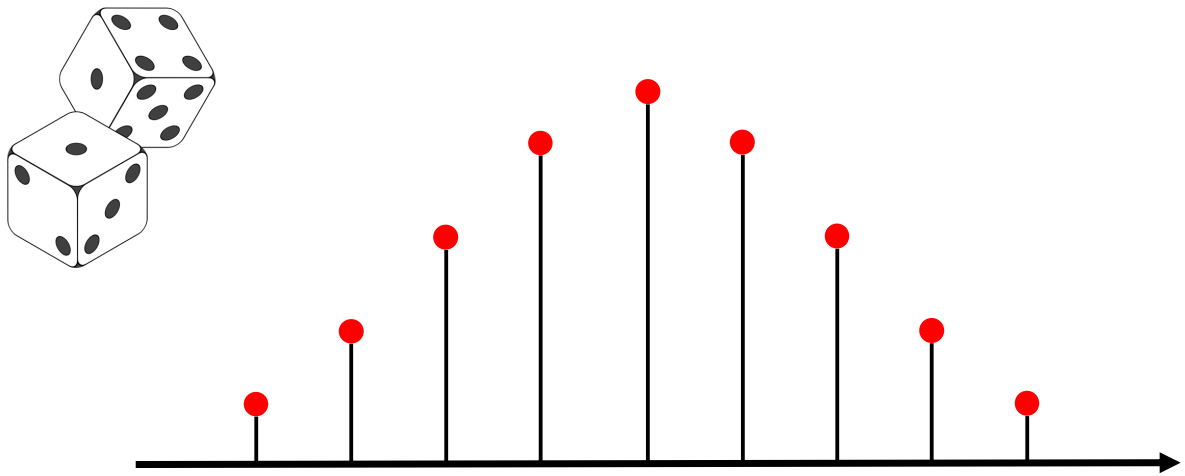
new samples

How to deal with terrible twos?

- Drink a coffee before engaging
- Lower your expectations
- Always carry snacks
- Pick your battles
- Master distraction
- Accept that logic is useless

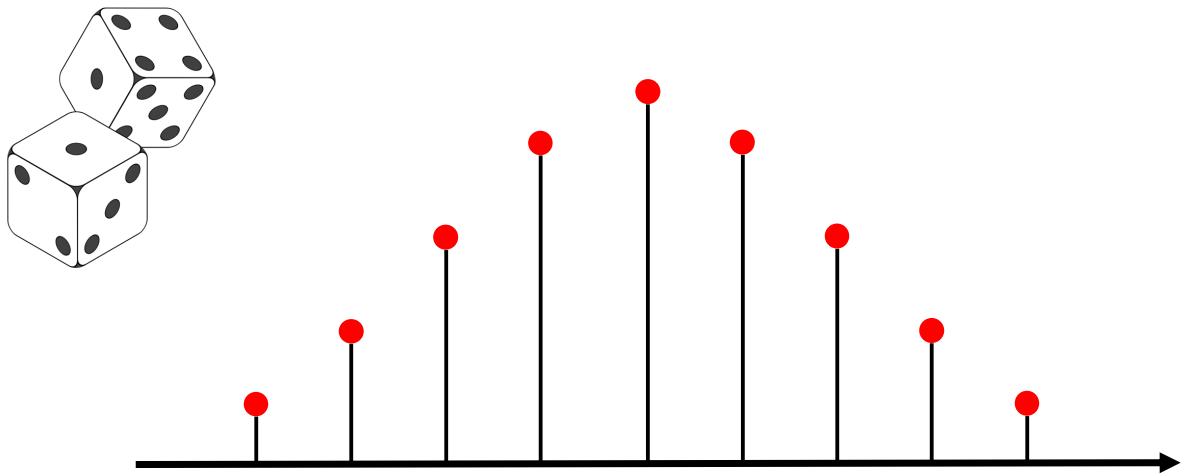
- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a discrete distribution}}$ $(1 \leq i \leq N)$ in $[S]^d$
- Generate new samples $Y \sim p_{\text{data}}$

Challenges in discrete probabilistic modeling



Discrete distribution: $p(x) \geq 0$, and $\sum_{x \in \mathcal{X}} p(x) = 1$.

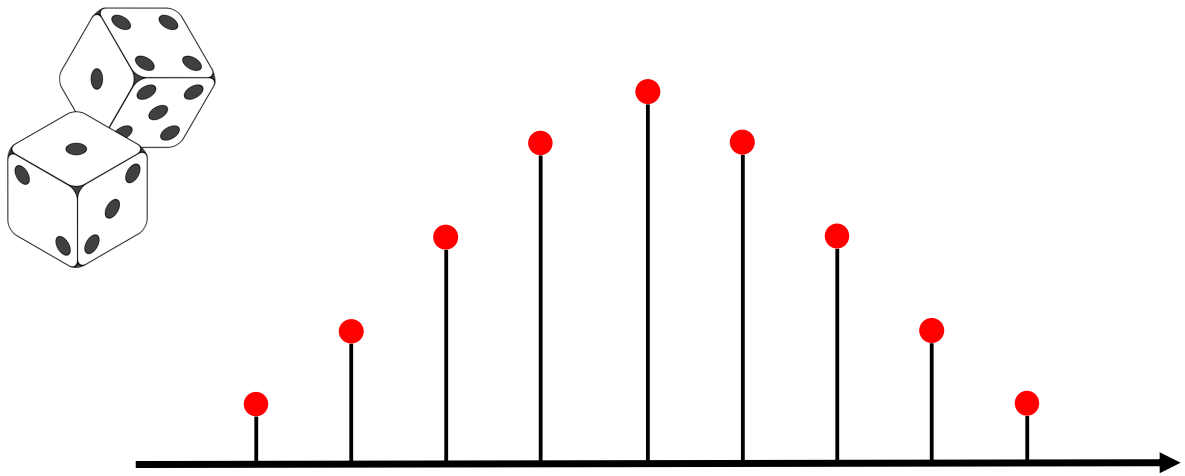
Challenges in discrete probabilistic modeling



Discrete distribution: $p(x) \geq 0$, and $\sum_{x \in \mathcal{X}} p(x) = 1$.

- large sample space S^d, exponentially grow with d

Challenges in discrete probabilistic modeling

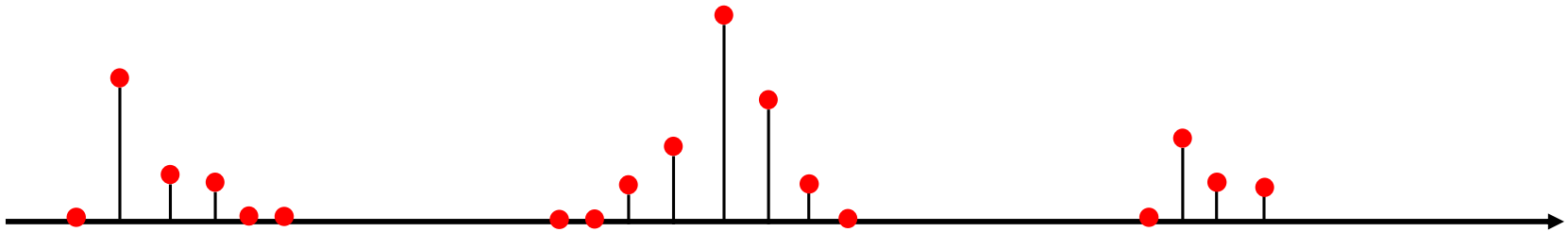


Discrete distribution: $p(x) \geq 0$, and $\sum_{x \in \mathcal{X}} p(x) = 1$.

- large sample space S^d, exponentially grow with d
- no gradient $\nabla_x p(x)$

Challenges in discrete probabilistic modeling

Proper text: We thank the reviewer for the insightful comments



Discrete distribution: $p(x) \geq 0$, and $\sum_{x \in \mathcal{X}} p(x) = 1$.

- large sample space S^d, exponentially grow with d
- no gradient $\nabla_x p(x)$
- real texts are highly structured

A successful paradigm: Autoregressive modeling

$\mathcal{X} = \{1, \ldots, S\}^d$, $x =$ All animals are equal, but some are more equal

$$p_{\text{data}}(x) = p_{\text{data}}(x_1, \ldots, x_d)$$

$$= p_{\text{data}}(x_1) p_{\text{data}}(x_2 \mid x_1) \ldots p_{\text{data}}(x_d \mid x_1, x_2, \ldots, x_{d-1})$$

$$\downarrow$$

$$p_\theta(x_i \mid \text{previous words})$$

A successful paradigm: Autoregressive modeling

$\mathcal{X} = \{1, \ldots, S\}^d$, $x =$ All animals are equal, but some are more equal

$$p_{\text{data}}(x) = p_{\text{data}}(x_1, \ldots, x_d)$$

$$= p_{\text{data}}(x_1) p_{\text{data}}(x_2 \mid x_1) \ldots p_{\text{data}}(x_d \mid x_1, x_2, \ldots, x_{d-1})$$

$$\downarrow$$

$$p_\theta(x_i \mid \text{previous words})$$

advantages:

- only requires modeling $p_\theta(x_i \mid \text{previous words})$ on $[S]$
- approximate any probability distribution

A successful paradigm: Autoregressive modeling

$\mathcal{X} = \{1, \ldots, S\}^d$, $x =$ All animals are equal, but some are more equal

$$p_{\text{data}}(x) = p_{\text{data}}(x_1, \ldots, x_d)$$

$$= p_{\text{data}}(x_1) p_{\text{data}}(x_2 \mid x_1) \ldots p_{\text{data}}(x_d \mid x_1, x_2, \ldots, x_{d-1})$$

$$\downarrow$$

$$p_\theta(x_i \mid \text{previous words})$$

disavantages:

- a rigid left-to-right order $\rightarrow$ hard to control

A successful paradigm: Autoregressive modeling

$\mathcal{X} = \{1, \ldots, S\}^d$, $x =$ All animals are equal, but some are more equal

$$p_{\text{data}}(x) = p_{\text{data}}(x_1, \ldots, x_d)$$

$$= p_{\text{data}}(x_1) p_{\text{data}}(x_2 \mid x_1) \ldots p_{\text{data}}(x_d \mid x_1, x_2, \ldots, x_{d-1})$$

$$\downarrow$$

$$p_\theta(x_i \mid \text{previous words})$$

disavantages:

- a rigid left-to-right order $\rightarrow$ hard to control
- slow sampling speed

A successful paradigm: Autoregressive modeling

$\mathcal{X} = \{1, \ldots, S\}^d$, $x =$ All animals are equal, but some are more equal

$$p_{\text{data}}(x) = p_{\text{data}}(x_1, \ldots, x_d)$$

$$= p_{\text{data}}(x_1) p_{\text{data}}(x_2 \mid x_1) \ldots p_{\text{data}}(x_d \mid x_1, x_2, \ldots, x_{d-1})$$

$$\downarrow$$

$$p_\theta(x_i \mid \text{previous words})$$

disavantages:

- a rigid left-to-right order $\rightarrow$ hard to control
- slow sampling speed
- errors accumulate

"autoregressive transformers are "doomed", as generation "drifts" from the data distribution and diverges during sampling." — Yann LeCun

An alternative: modeling score function

- write discrete distribution (parameterized by θ):

$$p_\theta(x) = \frac{e^{f_\theta(x)}}{Z_\theta}$$

where $Z_\theta = \sum_{x \in \mathcal{X}} p_\theta(x)$ is a normalizing const. depending on θ

An alternative: modeling score function

- write discrete distribution (parameterized by θ):

$$p_\theta(x) = \frac{e^{f_\theta(x)}}{Z_\theta}$$

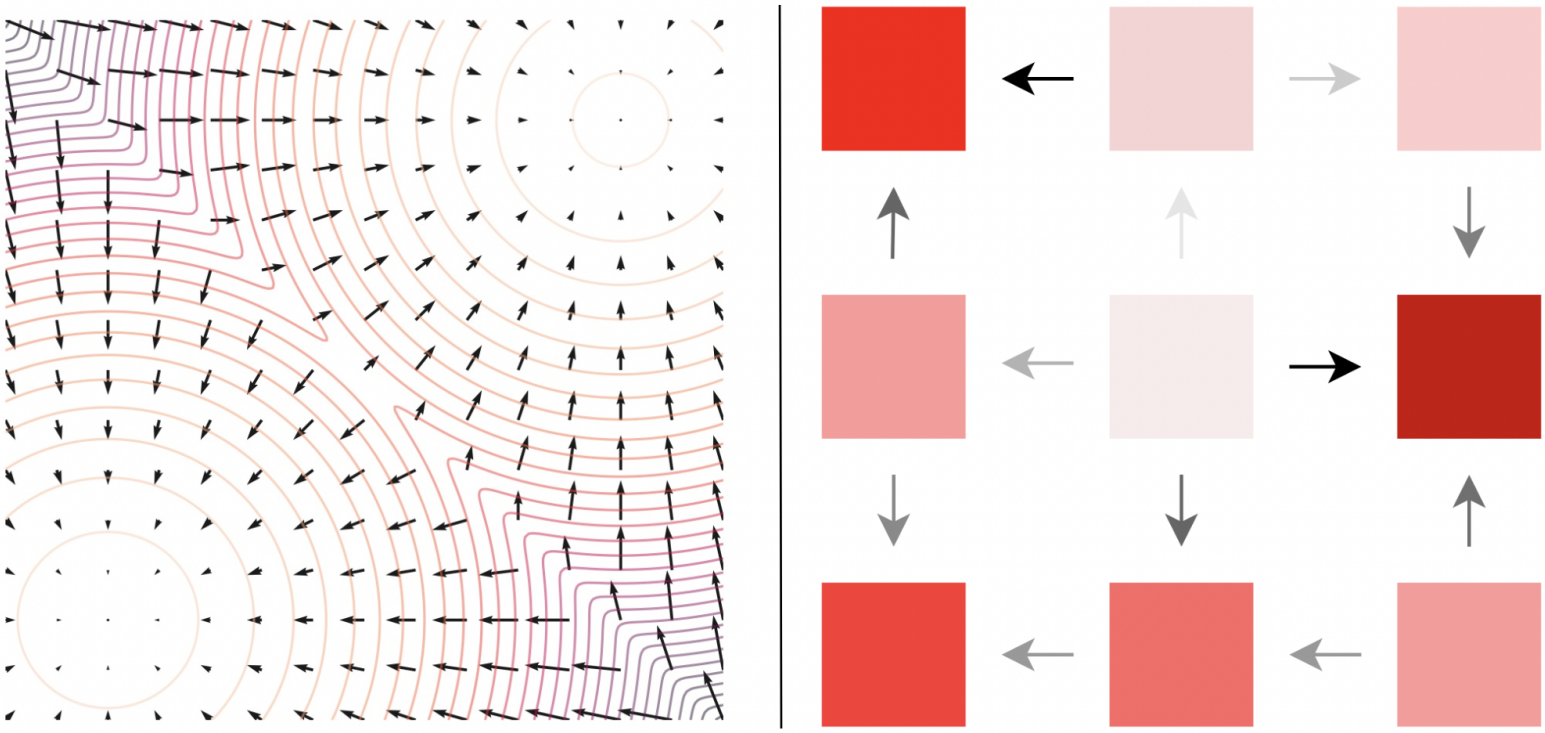
where $Z_\theta = \sum_{x \in \mathcal{X}} p_\theta(x)$ is a normalizing const. depending on θ

- concrete score function:

$$s_\theta(y, x) = \frac{p_\theta(y)}{p_\theta(x)} = \frac{e^{f_\theta(y)}}{e^{f_\theta(x)}}$$

— *analogous to* $\nabla \log p_\theta(x) = \nabla_x f_\theta(x)$ *in cont. case*

An alternative: modeling score function



Left (continous space): score function $\nabla_x \log p(x)$ points to higher density regime

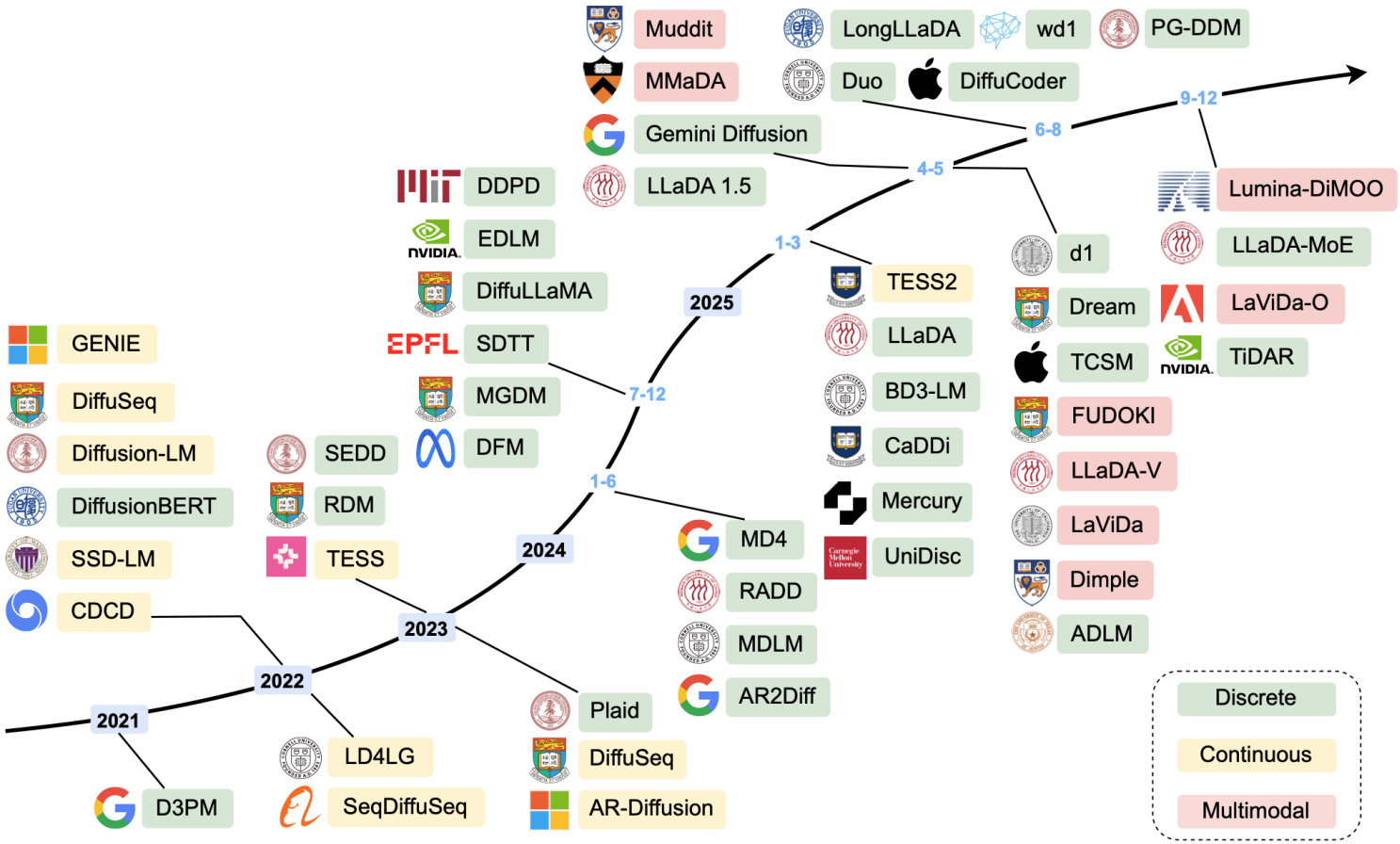
Right (discrete space): concrete score $\frac{p(y)}{p(x)}$ generalizes for discrete spaces

— fig credit: Aaron Lou

Towards a solid foundation for discrete diffusion models

Austin, Johnson, Ho, Tarlow, van den Berg '21
Lou, Meng, Ermon '24
Sahoo, Arriola, Schiff, Gokaslan, Marroquin, Chiu, Rush, Kuleshov '24
Ou, Nie, Xue, Zhu, Sun, Li, Li '24
Campbell, Benton, De Bortoli, Rainforth, Deligiannidis, Doucet '22
Chen, Ying '25
Liang, Liang, Lai, Shroff '25
Li, Cai '25
Bach, Saremi '25
. . .

Progress of discrete diffusions

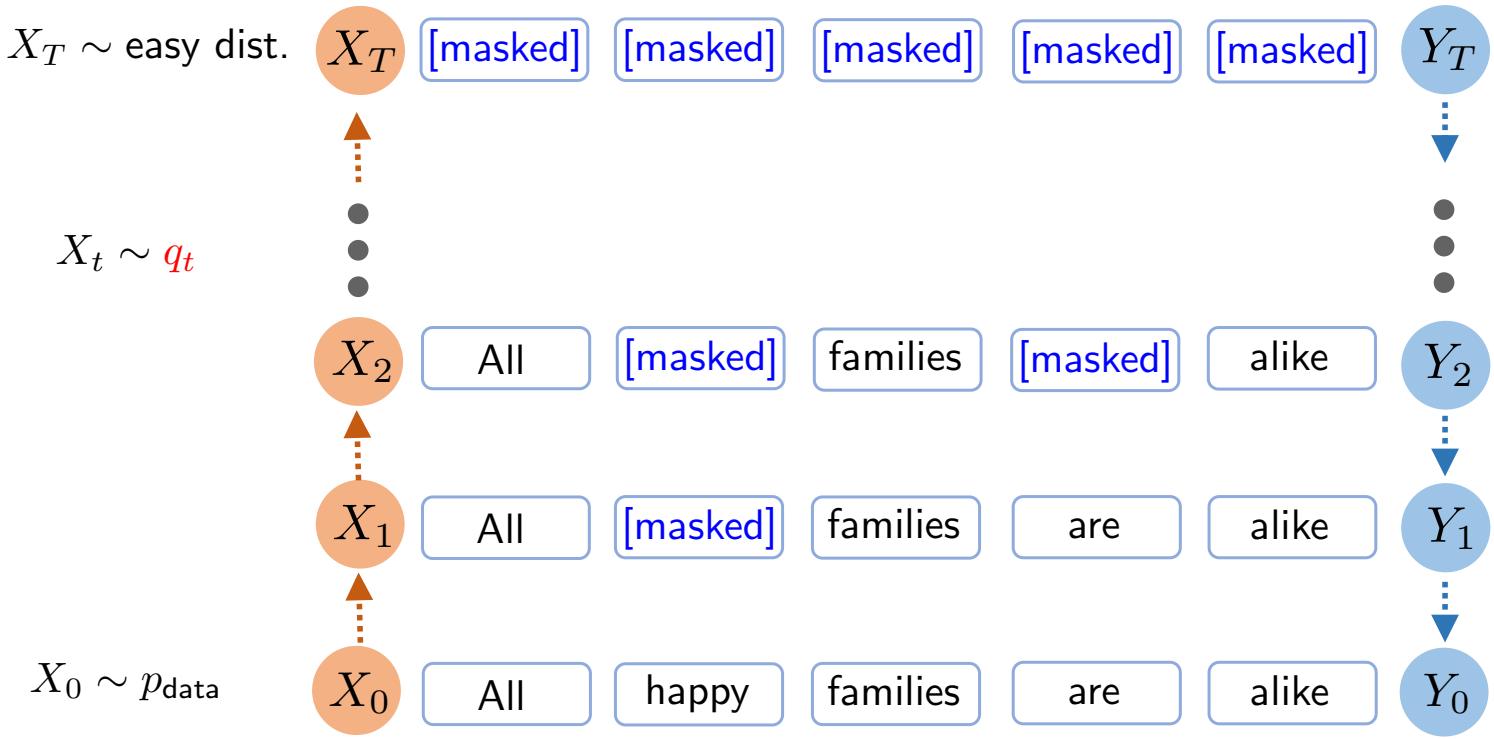


— the theoretical understanding remains highly inadequate...

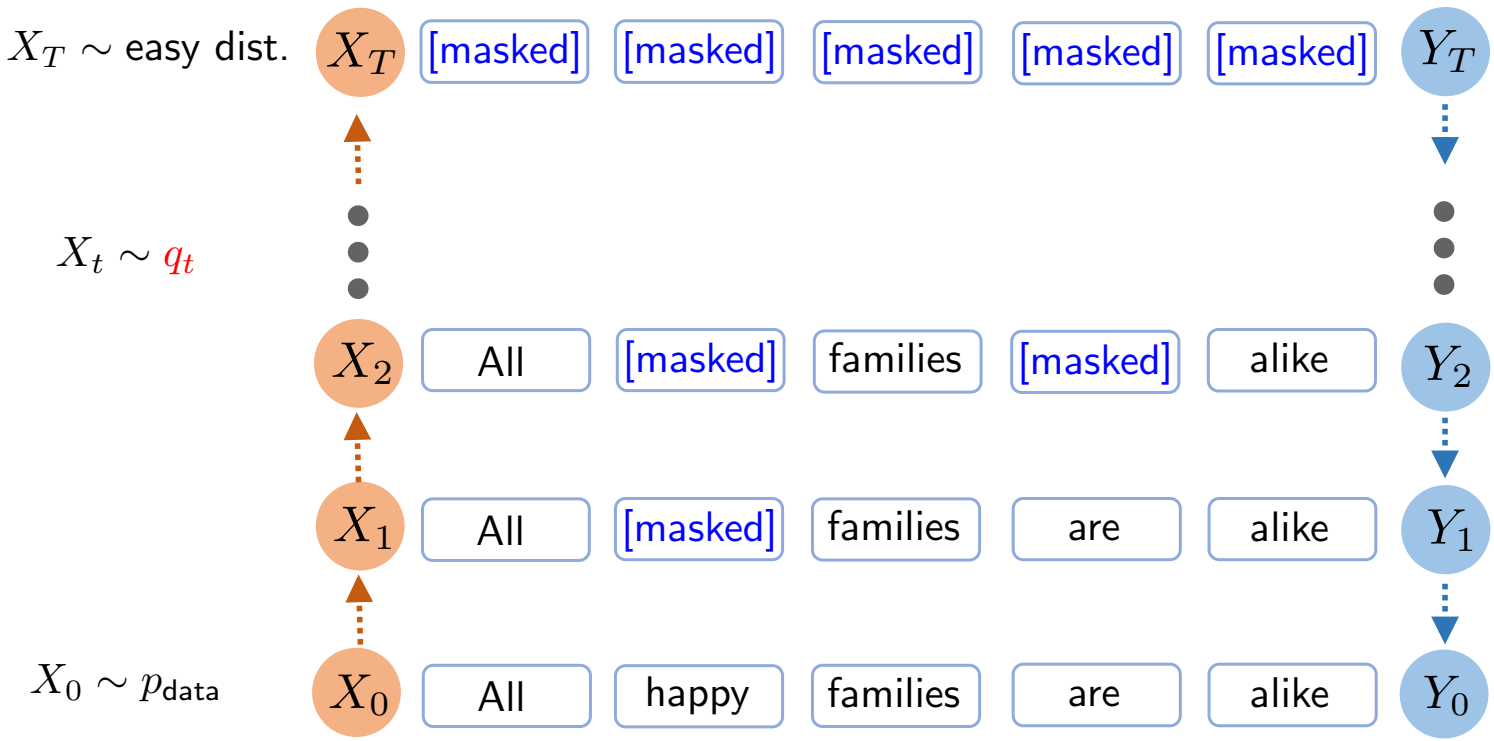
A brief introduction to discrete diffusion under continuous-time Markov chain formulation

CTMC

A CTMC formulation for discrete case



A CTMC formulation for discrete case

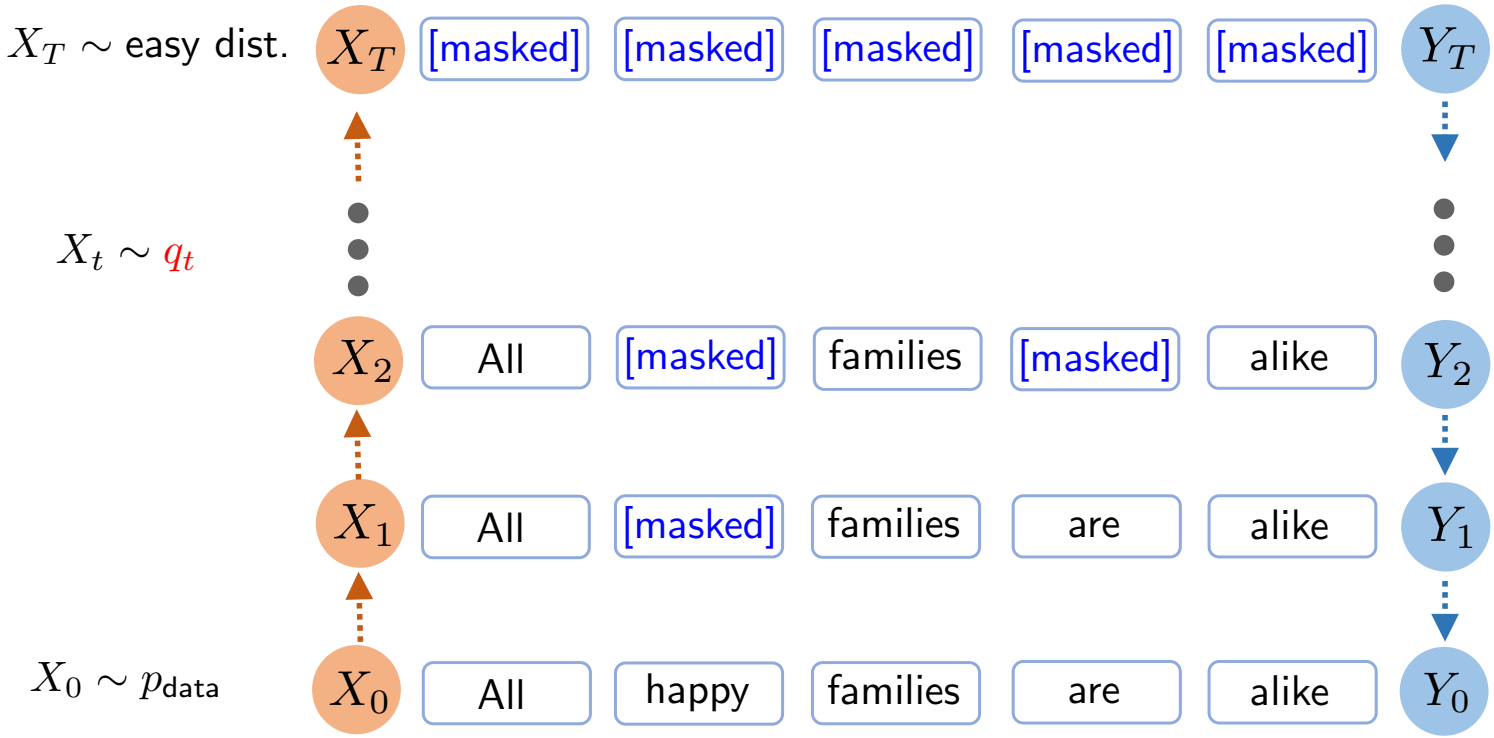


The transition probabilities satisfy, as $\Delta t \to 0^+$:

$$\Pr(x_{t+\Delta t} = y \mid x_t = x) = \boldsymbol{I}\{x = y\} + Q_t(x, y)\Delta t + o(\Delta t)$$

rate matrix: $Q_t(x, y) \geq 0, y \neq x,\ Q_t(x, x) = -\sum_{y \neq x} Q_t(x, y)$

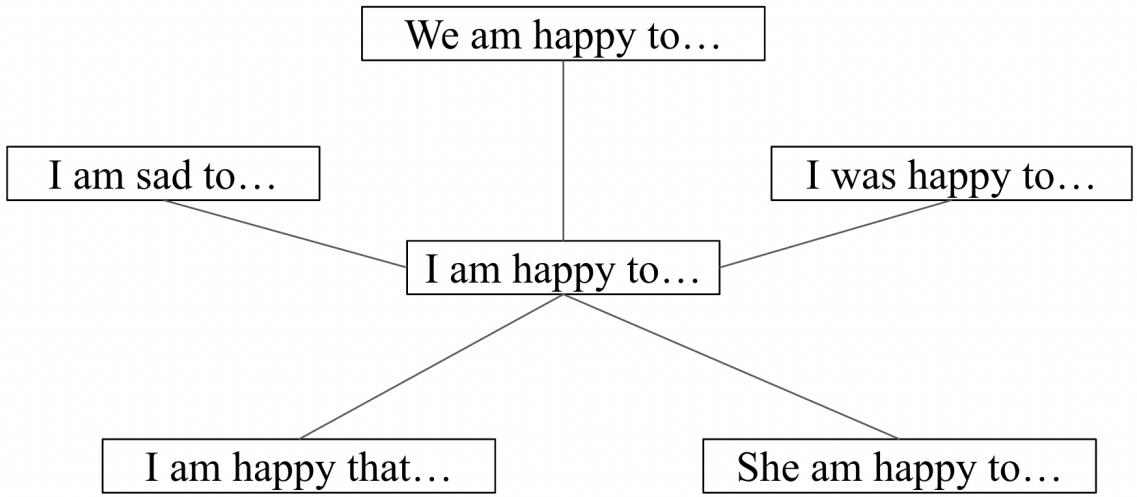
A CTMC formulation for discrete case



The transition probabilities satisfy, as $\Delta t \to 0^+$:

Kolmogorov equation: $\dfrac{\mathrm{d}q_t}{\mathrm{d}t} = {Q_t}^\top q_t, \qquad \text{for } 0 \le t \le T$

Two prevalent examples



- Uniform noising process $\longrightarrow$ Unif($\mathcal{X}$)

$$Q_t(x, y) = 1/S, \quad x \sim y\text{: differ at one coordinate}$$

- Masking noising process $\longrightarrow$ δ_{MASK}

$$Q_t(x, y) = 1, \quad \text{for some } i,\ x_i \neq \text{MASK},\ y_i = \text{MASK},\ x^{-i} = y^{-i}$$

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

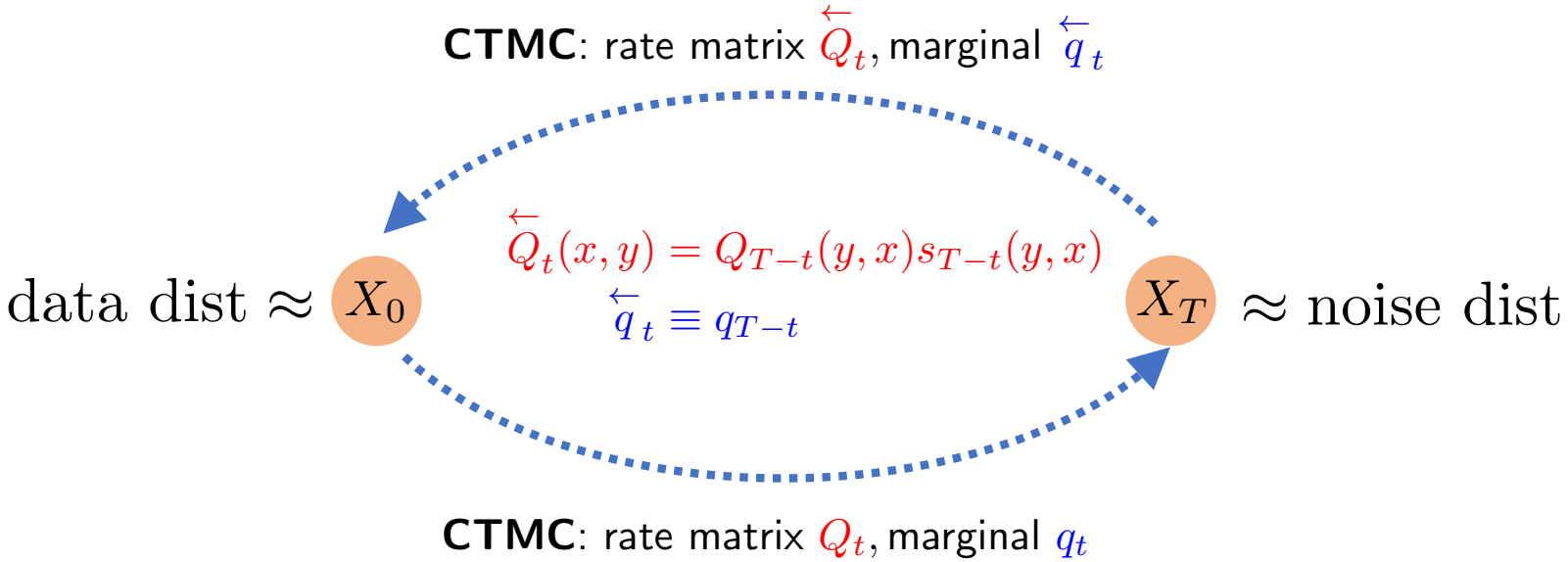
How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$, for $t = T, \cdots, 1$?

It is feasible as long as one knows score $s_t(y, x) := p_t(y)/p_t(x)$

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 score $s_t(y, x) := p_t(y)/p_t(x)$



Score estimation

Score entropy loss Lou et al.'24:

$$\mathcal{L}_{\mathrm{SE}}(t, \widehat{s}, s) := \mathbb{E}_{x \sim q_t} \left[\sum_{y \neq x} Q_t(y, x) s(y, x) \underbrace{\left(\frac{\widehat{s}(y, x)}{s(y, x)} - 1 - \log \left(\frac{\widehat{s}(y, x)}{s(y, x)} \right) \right)}_{\text{Bregman divergence } D(\widehat{s}, s) \text{ with } \phi(x) = -\log x} \right]$$

Score estimation

Score entropy loss Lou et al.'24:

$$\mathcal{L}_{\mathrm{SE}}(t, \widehat{s}, s) := \mathbb{E}_{x \sim q_t} \left[\sum_{y \neq x} Q_t(y, x) s(y, x) \underbrace{\left(\frac{\widehat{s}(y, x)}{s(y, x)} - 1 - \log \left(\frac{\widehat{s}(y, x)}{s(y, x)} \right) \right)}_{\text{Bregman divergence } D(\widehat{s}, s) \text{ with } \phi(x) = -\log x} \right]$$

Key observations:

- if $p(x) = \sum_{x_0} p(x \mid x_0) p_0(x_0)$, $s(y, x) := \frac{p(y)}{p(x)}$ can be replaced by $\frac{p(y|x_0)}{p(x|x_0)}$

Score estimation

Score entropy loss Lou et al.'24:

$$\mathcal{L}_{\mathrm{SE}}(t, \widehat{s}, s) := \mathbb{E}_{x \sim q_t} \left[\sum_{y \neq x} Q_t(y, x) s(y, x) \underbrace{\left(\frac{\widehat{s}(y,x)}{s(y,x)} - 1 - \log \left(\frac{\widehat{s}(y,x)}{s(y,x)} \right) \right)}_{\text{Bregman divergence } D_{\widehat{s},s} \text{ with } \phi(x) = -\log x} \right]$$

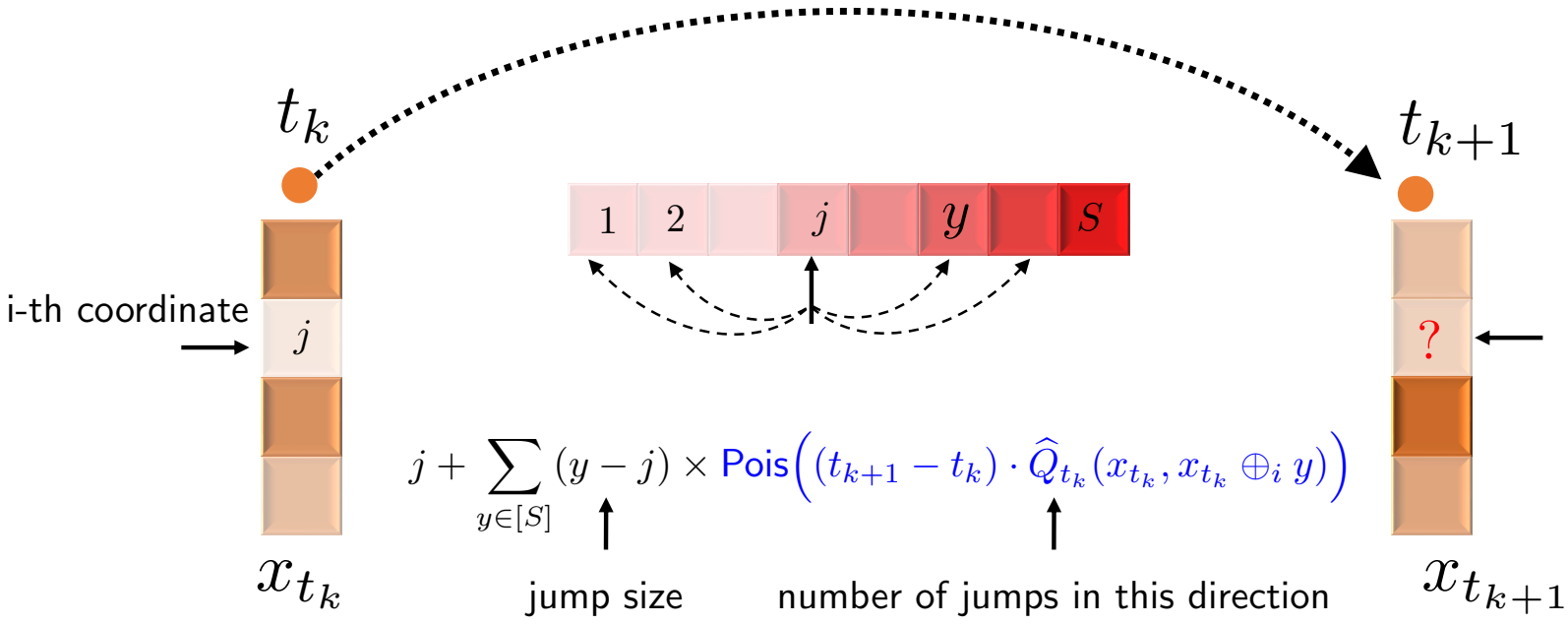
Key observations:

- if $p(x) = \sum_{x_0} p(x \mid x_0) p_0(x_0)$, $s(y,x) := \frac{p(y)}{p(x)}$ can be replaced by $\frac{p(y|x_0)}{p(x|x_0)}$
- instead of evaluating $s(y,x)$ for every $y \neq x$, only requires $x \sim y$

— *see, Lou et al.'24, Benton et al.'24, Ou et al.'25*

Score-based sampling: τ-leaping

— *Campbell, et al.'22*



Given $\widehat{s}_{t_i}(y, x)$ at points $0 \le t_0 < t_1 < \ldots < t_N \le T$, τ-leaping is equivalent to a CTMC with

$$\widehat{Q}^i_t(a, b) = \widehat{Q}_{t_k}(x_{t_k}, x_{t_k} \oplus_i (b - a)), \qquad \text{for } i \in [d],$$

with $\widehat{Q}_t(x, y) = Q_{T-t}(y, x)\widehat{s}_{T-t}(y, x)$.

Prior theory

To obtain samples that are ε-close in KL to p_{data}, it takes

$$\widetilde{O}\Big(\frac{d^2 S}{\varepsilon}\Big) \qquad \text{and} \qquad \widetilde{O}\Big(\frac{dS}{\varepsilon}\Big)$$

for uniform and mask noising processes, respectively (*Liang et al.'25*)

Estimating the number of unseen species: How many words did Shakespeare know?

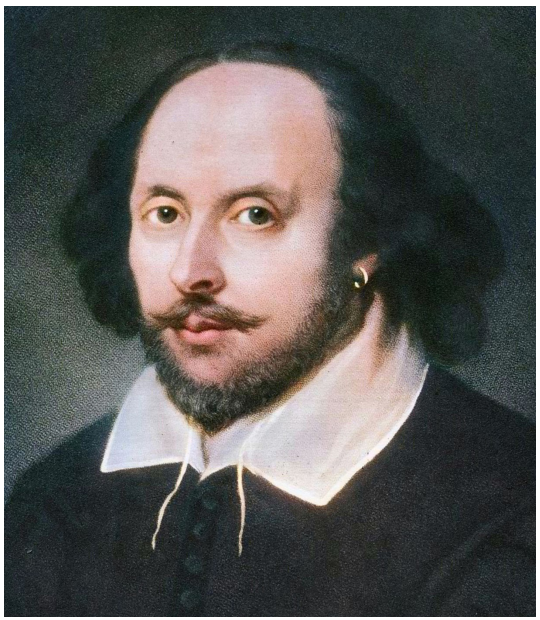
BRADLEY EFRON, RONALD THISTED

Biometrika, Volume 63, Issue 3, December 1976, Pages 435–447,

Abstract

SUMMARY

Shakespeare wrote 31534 different words, of which 14376 appear only once, 4343 twice, etc. The question considered is how many words he knew but did not use. A parametric empirical Bayes model due to Fisher and a nonparametric model due to Good & Toulmin are examined. The latter theory is augmented using linear programming methods. We conclude that the models are equivalent to supposing that Shakespeare knew at least 35000 more words.



Can we develop sharp dependences on d and S?

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

- For discretization points: $0 \le t_0 < t_1 < \ldots < t_N \le T$, assume

$$\sum_{k=0}^{N-1} (t_{k+1} - t_k) \mathcal{L}_{\mathrm{SE}}(T - t_k, \widehat{s}_{T-t_k}, s_{T-t_k}) \le \varepsilon_{\text{score}}$$

— *cont. case:* $\frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{X \sim q_t} \left[\| s_t(X) - s_t^\star(X) \|_2^2 \right] \le \varepsilon_{\text{score}}^2$

✗ boundedness assump, e.g. $\widehat{s}_{t_k}(x, y) \in [1/M, M]$

✗ regularity assump, e.g. *continuity of the score function*

Uniform discrete diffusion

Theorem 11 (Upper bound for uniform noising process)

For $0 = t_0 < t_1 < \ldots < t_N = T$, *let* $\Delta := \max_k \{t_{k+1} - t_k\} = O(1)$. *The* τ*-leaping algorithm achieves*

$$\mathsf{KL}(p_{\text{data}} \| p_{\text{output}}) \lesssim \varepsilon_{\text{score}} + e^{-T} d \log S + \Delta d \log \left(\frac{S}{\Delta} \right).$$

Uniform discrete diffusion

Theorem 11 (Upper bound for uniform noising process)

For $0 = t_0 < t_1 < \ldots < t_N = T$, *let* $\Delta := \max_k\{t_{k+1} - t_k\} = O(1)$. *The* τ*-leaping algorithm achieves*

$$\mathsf{KL}(p_{\text{data}} \| p_{\text{output}}) \lesssim \varepsilon_{\text{score}} + e^{-T} d \log S + \Delta d \log\left(\frac{S}{\Delta}\right).$$

Under constant step size schedule, $t_{k+1} - t_k = T/N$,

$$N = \widetilde{O}\left(\frac{d \log S}{\varepsilon}\right) \quad \text{discretization steps}$$

are enough to guarantee $\mathsf{KL}(p_{\text{data}} \| p_{\text{output}}) \leq \varepsilon_{\text{score}} + \varepsilon$.

Uniform discrete diffusion

Theorem 11 (Upper bound for uniform noising process)

For $0 = t_0 < t_1 < \ldots < t_N = T$, *let* $\Delta := \max_k \{t_{k+1} - t_k\} = O(1)$. *The* τ*-leaping algorithm achieves*

$$\mathsf{KL}(p_{\text{data}} \| p_{\text{output}}) \lesssim \varepsilon_{\text{score}} + e^{-T} d \log S + \Delta d \log \left(\frac{S}{\Delta} \right).$$

Under constant step size schedule, $t_{k+1} - t_k = T/N$,

$$N = \widetilde{O}\left(\frac{d \log S}{\varepsilon} \right) \quad \text{discretization steps}$$

are enough to guarantee $\mathsf{KL}(p_{\text{data}} \| p_{\text{output}}) \leq \varepsilon_{\text{score}} + \varepsilon$.

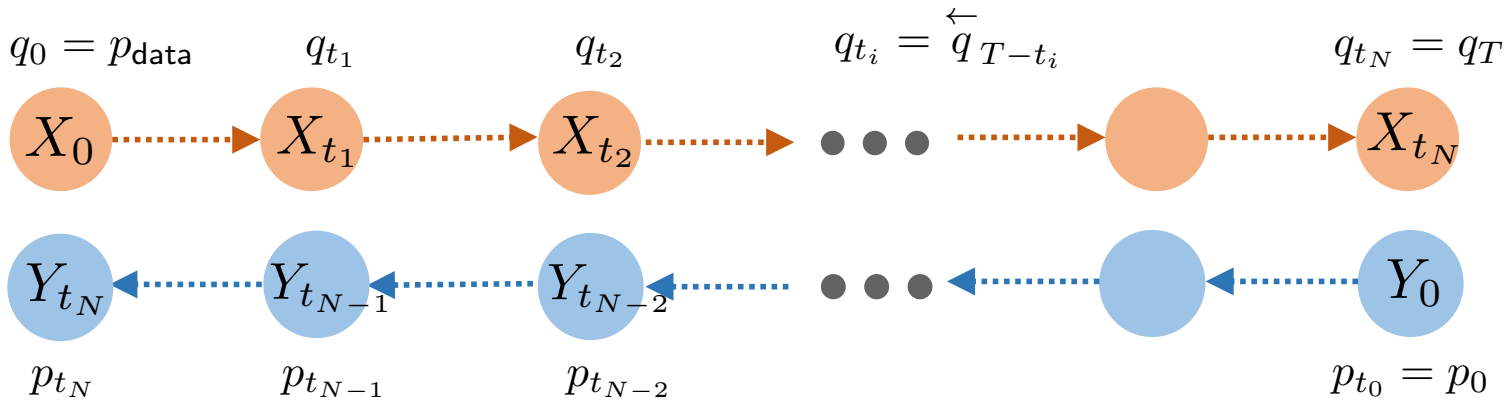
— matching lower bound for dist. far from uniform.

Comparisons with prior work

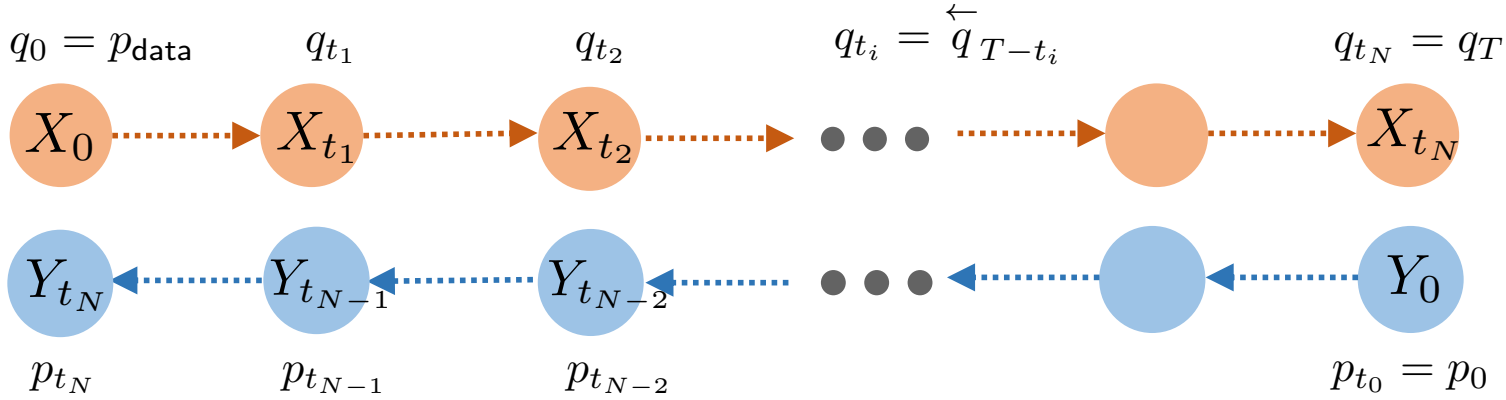
Paper	Score Est. Assump.	No Early Stopping	Iteration Complexity
Ren et al.,'24	Bounded	✗	d^2S^2/ε
Liang et al.,'25a	Bounded	✗	d^2S/ε
This work	None	✓	$d\log S/\varepsilon$

No early stopping and extra constraints are needed on the score estimator!

Proof sketch: Step 1



Proof sketch: Step 1



$$\mathsf{KL}(q_0 \parallel p_T) \le \mathsf{KL}(q_{T-t_0,\dots,T-t_N} \parallel p_{t_0,\dots,t_N})$$

$$= \mathsf{KL}(q_T \parallel p_0) + \sum_{k=0}^{N-1} \mathbb{E}_{x_{t_k} \sim \overleftarrow{q}_{t_k}} \mathsf{KL}(\overleftarrow{q}_{t_{k+1}|t_k}(\cdot \mid x_{t_k}) \parallel p_{t_{k+1}|t_k}(\cdot \mid x_{t_k}))$$

$$\le \mathsf{KL}(q_T \parallel p_0) + \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}_{x_t \sim \overleftarrow{q}_t} \sum_{y \neq x_t} \overleftarrow{Q}_t(x_t, y) \underbrace{D\left(\widehat{Q}_t(x_t, y), \overleftarrow{Q}_t(x_t, y)\right)}_{D(a,b) := \frac{a}{b} - 1 - \log(\frac{a}{b})} \mathrm{d}t$$

Girsanov's change-of-measure Theorem (e.g. *Liang et al., 2025*):

Proof sketch: Step 2

For τ-leaping sampler, consider $t \in [t_k, t_{k+1})$ and write $\ell = t_k$.

$$\sum_{y \neq x_t} \overleftarrow{Q}_t(x_t, y) D\left(\widehat{Q}_t(x_t, y), \overleftarrow{Q}_t(x_t, y)\right)$$

$$= \frac{1}{S} \sum_{i \in [d]} \sum_{c \in [S]} s_{T-t}(x_t \oplus_i c, x_t) D\left(\widehat{s}_{T-\ell}(x_\ell \oplus_i c, x_\ell), s_{T-t}(x_t \oplus_i c, x_t)\right)$$

$$= \underbrace{\frac{1}{S} \sum_{y_\ell \sim x_\ell} s_{T-\ell}(y_\ell, x_\ell) D\left(\widehat{s}_{T-\ell}(y_\ell, x_\ell), s_{T-\ell}(y_\ell, x_\ell)\right)}_{\text{controlled by small score entropy loss}}$$

$$+ \underbrace{\frac{1}{S} \sum_{i \in [d]} \sum_{c \in [S]} \left(s_{T-\ell}(x_\ell \oplus_i c, x_\ell) - s_{T-t}(x_t \oplus_i c, x_t)\right) \log \widehat{s}_{T-\ell}(x_\ell \oplus_i c, x_\ell)}_{\text{expectation controlled by martingale property} = 0}$$

$$+ \frac{1}{S} \sum_{y_t \sim x_t} \left(-\log s_{T-t}(y_t, x_t)\right) - \frac{1}{S} \sum_{y_\ell \sim x_\ell} \left(-\log s_{T-\ell}(y_\ell, x_\ell)\right).$$

Proof sketch: Step 3

It remains to control $\varphi(T-t) - \varphi(T-t_k)$ with

$$\varphi(t) \coloneqq \frac{1}{S}\mathbb{E}_{x_t \sim q_t}\left[\sum_{i\in[d]}\sum_{c\in[S]} -\log s_t(x_t \oplus_i c, x_t)\right]$$

Proof sketch: Step 3

It remains to control $\varphi(T-t) - \varphi(T-t_k)$ with

$$\varphi(t) \coloneqq \frac{1}{S}\mathbb{E}_{x_t \sim q_t}\left[\sum_{i\in[d]}\sum_{c\in[S]} -\log s_t(x_t \oplus_i c, x_t)\right]$$

Key observation: $\varphi(t)$ is (i) non-increasing and (ii) bounded $0 \le \varphi(t) \le d(\log S + \max\{\log t^{-1}, 0\})$.

Proof sketch: Step 3

It remains to control $\varphi(T-t) - \varphi(T-t_k)$ with

$$\varphi(t) := \frac{1}{S}\mathbb{E}_{x_t \sim q_t}\left[\sum_{i\in[d]}\sum_{c\in[S]} -\log s_t(x_t \oplus_i c, x_t)\right]$$

Key observation: $\varphi(t)$ is (i) non-increasing and (ii) bounded $0 \le \varphi(t) \le d(\log S + \max\{\log t^{-1}, 0\})$.

As a result, using (i), (ii), and the telescoping summation, one arrives at

$$\underbrace{\sum_{k=0}^{N-1}\int_{t_k}^{t_{k+1}} (\varphi(T-t) - \varphi(T-t_k))\,\mathrm{d}t}_{\text{key quantity for discretization error}} \lesssim \Delta d \log\frac{S}{\Delta}.$$

Sharp dependence but not yet adaptive...
What about masking diffusion?

— *e.g., for* $p_{\text{data}} = \frac{1}{2}\delta_{\mathbf{0}} + \frac{1}{2}\delta_{\mathbf{1}}$ *or* $p_{\text{data}} = \text{Unif}(\mathcal{X})$

Fundamental information-theoretic quantities

Let P be a distribution over $[S]^d$ and $X := (X_1, \ldots, X_d) \sim P$. Define

total correlation
$$\mathcal{C}(P) := \sum_{i=1}^{d} \underbrace{\mathcal{H}(X_i)}_{\text{entropy}} - \mathcal{H}(X_1, \ldots, X_d)$$

dual total correlation
$$\mathcal{B}(P) := \mathcal{H}(X_1, \ldots, X_d) - \sum_{i=1}^{d} \mathcal{H}(X_i \mid X_{-i})$$

Fundamental information-theoretic quantities

Let P be a distribution over $[S]^d$ and $X := (X_1, \ldots, X_d) \sim P$. Define

total correlation $$\mathcal{C}(P) := \sum_{i=1}^{d} \underbrace{\mathcal{H}(X_i)}_{\text{entropy}} - \mathcal{H}(X_1, \ldots, X_d)$$

dual total correlation $$\mathcal{B}(P) := \mathcal{H}(X_1, \ldots, X_d) - \sum_{i=1}^{d} \mathcal{H}(X_i \mid X_{-i})$$

- $\mathcal{C}(P) = \int_0^\infty (e^t - 1)\mathcal{I}(t)\mathrm{d}t$ and $\mathcal{B}(P) := \int_0^\infty \mathcal{I}(t)\mathrm{d}t$ for

$$\mathcal{I}(t) := \sum_{i \neq j \in [d]} \underbrace{\mathrm{I}(x_t^i; x_t^j \mid x_t^{-(i,j)})}_{\text{conditional mutual information}}, \qquad \text{for } P = q_0 = p_{\text{data}} \text{ and } x_t \sim q_t.$$

Fundamental information-theoretic quantities

Let P be a distribution over $[S]^d$ and $X := (X_1, \ldots, X_d) \sim P$. Define

total correlation $$\mathcal{C}(P) := \sum_{i=1}^{d} \underbrace{\mathcal{H}(X_i)}_{\text{entropy}} - \mathcal{H}(X_1, \ldots, X_d)$$

dual total correlation $$\mathcal{B}(P) := \mathcal{H}(X_1, \ldots, X_d) - \sum_{i=1}^{d} \mathcal{H}(X_i \mid X_{-i})$$

- $\mathcal{C}(P) = \int_0^\infty (e^t - 1)\mathcal{I}(t)\mathrm{d}t$ and $\mathcal{B}(P) := \int_0^\infty \mathcal{I}(t)\mathrm{d}t$ for

$$\mathcal{I}(t) := \sum_{i \neq j \in [d]} \underbrace{\mathrm{I}(x_t^i; x_t^j \mid x_t^{-(i,j)})}_{\text{conditional mutual information}}, \qquad \text{for } P = q_0 = p_{\text{data}} \text{ and } x_t \sim q_t.$$

- define effective total correlation

$$\mathcal{D}(P) := \int_0^\infty \min(1, t) \cdot \mathcal{I}(t)\mathrm{d}t \leq \min\{\mathcal{C}(P), \mathcal{B}(P)\}.$$

Modified τ-leaping

Consider the following modification of τ-leaping (for $t \in [t_k, t_{k+1})$):

$$\widehat{Q}_t^i(a,b) = \frac{e^{T-t_k}-1}{e^{T-t}-1}\widehat{s}_{T-t_k}(x_{t_k} \odot_i b, x_{t_k}) \cdot \underbrace{\boldsymbol{I}\{a = x_{t_k}^i = \mathrm{MASK}\}}_{\substack{\text{at most one transition at } x_t^i \\ \text{at the interval } [t_k, t_{k+1})}}$$

Key observation: for any $t > 0$, $b \in [S]$, $x \in ([S] \cup \{\mathrm{MASK}\})^d$, and i such that $x^i = \mathrm{MASK}$,

$$s_t(x \odot_i b, x) = \frac{1}{e^t - 1} \cdot s_0(x \odot_i b, x)$$

Masking discrete diffusion

Theorem 12 (Upper bound for uniform noising process)

For $0 = t_0 < t_1 < \ldots < t_N = T$, *modified* τ*-leaping satisfies*

$$\mathrm{KL}(p_{\mathrm{data}} \| p_{\mathrm{output}}) \lesssim \varepsilon_{\mathrm{score}} + e^{-T} d \log S + \sum_{k=0}^{N-1} (t_{k+1} - t_k) \int_{T-t_{k+1}}^{T-t_k} \mathcal{I}(t) \mathrm{d}t.$$

Under exponential-then-constant schedule, $t_{k+1} - t_k \le \kappa \min(1, T - t_{k+1})$,

$$N = \widetilde{O}\left(\frac{\mathcal{D}(p_{\mathrm{data}})}{\varepsilon}\right) \quad \text{discretization steps}$$

are enough to guarantee $\mathrm{KL}(p_{\mathrm{data}} \parallel p_{\mathrm{output}}) \le \varepsilon_{\mathrm{score}} + \varepsilon$.

— *it satisfies* $\mathcal{D}(P) \le \min\{\mathcal{C}(P), \mathcal{B}(P)\} \le d \log S$

Comparisons with prior work

Paper	Score Est. Assump.	No Early Stopping	Sampler	Iteration Complexity	Adaptation
Liang et al., 25b	Bounded	✗	τ-leaping	dS/ε	✗
Conforti et al., 25	$\widehat{s}_t \approx s_t$	✗	DMPM	dS/ε	✗
This work	None	✓	Modified τ-leaping	$\mathcal{D}/\varepsilon$	✓

No early stopping and **no** extra constraints on the score estimator!

Distributions of small $\mathcal{D}(p_{\text{data}})$

Completely independent of d

- product distributions on $[S]^d$: $\mathcal{D}(p_{\text{data}}) = 0$ ($N = 2$ suffices)
- mixture of two Dirac measures, $\frac{1}{2}\delta_{k_1} + \frac{1}{2}\delta_{k_2}$: $\mathcal{D}(p_{\text{data}}) = \log 2$

Sublinear dependence in d

- hidden Markov models
- cont. dist. with intrinsic dimension k, plus quantization
- sparse random regular graphs & stochastic block models

— sublinear convergence rates for masking discrete diffusion!

Example: Latent parity model

Let $d = 2k$ and consider the following distribution:

Example: Latent parity model

Let $d = 2k$ and consider the following distribution:

- sample $b \sim \mathrm{Bern}(1/2)$ & set $x_1, \ldots, x_k = b$

Example: Latent parity model

Let $d = 2k$ and consider the following distribution:

- sample $b \sim \mathrm{Bern}(1/2)$ & set $x_1, \ldots, x_k = b$
- sample $x_{k+1}, \ldots, x_{2k-1} \overset{\text{i.i.d.}}{\sim} \mathrm{Bern}(1/2)$

Example: Latent parity model

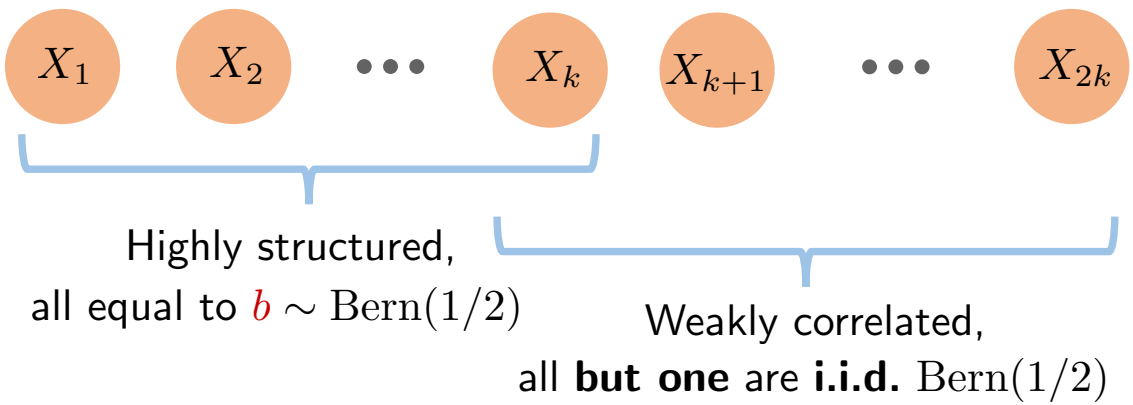
Let $d = 2k$ and consider the following distribution:

- sample $b \sim \mathrm{Bern}(1/2)$ & set $x_1, \ldots, x_k = b$
- sample $x_{k+1}, \ldots, x_{2k-1} \overset{\text{i.i.d.}}{\sim} \mathrm{Bern}(1/2)$
- set $x_{2k} = (b + \sum_{i=k+1}^{2k-1} x_i) \bmod 2$

Example: Latent parity model

Let $d = 2k$ and consider the following distribution:

- sample $b \sim \mathrm{Bern}(1/2)$ & set $x_1, \dots, x_k = b$
- sample $x_{k+1}, \dots, x_{2k-1} \overset{\text{i.i.d.}}{\sim} \mathrm{Bern}(1/2)$
- set $x_{2k} = (b + \sum_{i=k+1}^{2k-1} x_i) \bmod 2$



In this example, $\mathcal{D}(q_0) = \Theta(1) \ll \Theta(d) = \min(\mathcal{B}(q_0), \mathcal{C}(q_0))$

Proof sketch: Step 1

Similarly to the uniform case, writing $x_t := (x_t \odot_i c, x_t)$, and $x_{t_k} := (x_{t_k} \odot_i c, x_{t_k})$, we decompose

$$\sum_{y \neq x_t} \overleftarrow{Q}_t(x_t, y) D\left(\widehat{Q}_t(x_t, y), \overleftarrow{Q}_t(x_t, y)\right)$$

$$= \sum_{i \in m(x_t), c \in [S]} s_{T-t}(x_t) \cdot D\left(\frac{e^{T-t_k} - 1}{e^{T-t} - 1} \widehat{s}_{T-t_k}(x_{t_k}), s_{T-t}(x_t)\right)$$

$$= \sum_{i \in m(x_t), c \in [S]} \underbrace{s_{T-t}(x_{t_k}) D\left(\widehat{s}_{T-t_k}(x_{t_k}), s_{T-t_k}(x_{t_k})\right)}_{\text{controlled by small score entropy loss}}$$

$$+ \underbrace{\left(s_{T-t}(x_{t_k}) - s_{T-t}(x_t)\right) \log \frac{\widehat{s}_{T-t_k}(x_{t_k})}{s_{T-t_k}(x_{t_k})}}_{\text{expectation controlled by martingale property} = 0} + s_{T-t}(x_t) D(s_{T-t}(x_{t_k}), s_{T-t}(x_t)).$$

Proof sketch: Step 2

Using martingale property and changing measure under expectation:

$$\begin{aligned}
&\mathbb{E}_{x_t, x_{t_k} \sim \overleftarrow{q}_{t,t_k}} s_{T-t}(x_t \odot_i c, x_t) D(s_{T-t}(x_{t_k} \odot_i c, x_{t_k}), s_{T-t}(x_t \odot_i c, x_t)) \\
&\quad = \mathbb{E}_{y_t, y_{t_k} \sim \overleftarrow{q}_{t,t_k}} \log \frac{q_0(y_t) q_0(y_{t_k} \odot_i \mathrm{MASK})}{q_0(y_t \odot_i \mathrm{MASK}) q_0(y_{t_k} \odot_i y_t^i)} \\
&\quad = \mathbb{E}_{y_t, y_{t_k} \sim \overleftarrow{q}_{t,t_k}} \left(f_i(y_{t_k}) - f_i(y_t)\right),
\end{aligned}$$

where we define

$$f_i(y) \coloneqq \log \frac{q_0(y \odot_i y_t^i)}{q_0(y \odot_i \mathrm{MASK})}.$$

Connection to the conditional mutual information (via *Dynkin's formula*)

$$\mathbb{E}_{y_t, y_{t_k} \sim \overleftarrow{q}_{t,t_k}} \left(f_i(y_{t_k}) - f_i(y_t)\right) \lesssim \sum_{j \neq i} \int_{t_k}^{t} \mathrm{I}(y_v^i; y_v^j \mid y_v^{-(i,j)}) \mathrm{d}v.$$

Summary

- nonasymptotic convergence theory for diffusion models
- adaptation to unknown low dimensionality
- demystifying diffusion guidance
- provable training-free acceleration
- diffusion language models

Summary

- nonasymptotic convergence theory for diffusion models
- adaptation to unknown low dimensionality
- demystifying diffusion guidance
- provable training-free acceleration
- diffusion language models

Future directions:

- end-to-end theory that accounts for score learning + sampling?
- adaptive improvement under stylized statistical models
- design of high-order stochastic samplers

Papers I

"A sharp convergence theory for the probability flow ODEs of diffusion models," G. Li, Y. Wei, Y. Chi, Y. Chen, arXiv:2408.02320, 2024

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

"$O(d/T)$ convergence theory for diffusion probabilistic models under minimal assumptions," G. Li*, Y. Yan*, ICLR 2025

"Optimal convergence analysis of DDPM for general distributions," Y. Jiao, Y. Zhou, G. Li, arXiv:2510.27562, 2025

"Minimax optimality of the probability flow ode for diffusion models," C. Cai, G. Li, arXiv:2503.09583, 2025

"Adapting to unknown low-dimensional structures in score-based diffusion models," G. Li*, Y. Yan*, NeurIPS 2024

"Low-dimensional adaptation of diffusion models: convergence in total variation," J. Liang, Z. Huang, Y. Chen, COLT 2025

"Denoising diffusion probabilistic models are optimally adaptive to unknown low dimensionality," Z. Huang, Y. Wei, Y. Chen, *Mathematics of Operations Research*, 2026+

"Dimension-free convergence of diffusion models for approximate Gaussian mixtures," G. Li*, C. Cai*, Y. Wei, ICML 2026

Papers II

"Accelerating convergence of score-based diffusion models, provably," G. Li*, Y. Huang*, T. Efimov, Y. Wei, Y. Chi, Y. Chen, ICML 2024

"Stochastic Runge-Kutta methods: Provable acceleration of diffusion models," Y. Wu, Y. Chen, Y. Wei, arXiv:2410.04760, 2024

"Faster diffusion models via higher-order approximation," G. Li*, Y. Zhou*, Y. Wei, Y. Chen, arXiv:2506.24042, 2025

"Improved convergence rate for diffusion probabilistic models," G. Li, Y. Jiao, ICLR 2025

"Provable acceleration for diffusion models under minimal assumptions," G. Li*, C. Cai*, arXiv:2410.23285

"Theoretical insights for diffusion guidance: A case study for Gaussian mixture models," Y. Wu, M. Chen, Z. Li, M. Wang, Y. Wei, ICML 2024

"Provable efficiency of guidance in diffusion models for general data distribution," Y. Jiao, G. Li, ICML 2025

"Towards a unified framework for guided diffusion models," Y. Jiao, Y. Chen, G. Li, arXiv:2512.04985, 2025

"Provably solving inverse problems with diffusion prior through DDIM-type sampler," Y. Jiao, N. Li, C. Cai, Y. Chen, G. Li, 2026

Papers III

"Breaking AR's sampling bottleneck: provable acceleration via diffusion language models," G. Li*, C. Cai*, NeurIPS 2025

"Efficient sampling with discrete diffusion models: sharp and adaptive guarantees," Daniil Dmitriev*, Zhihan Huang*, Yuting Wei, COLT 2026

(*=equal contributions)