Faster Convergence and Accelerating for Diffusion-Based Generative Models

Gen Li

Department of Statistics and Data Science, CUHK

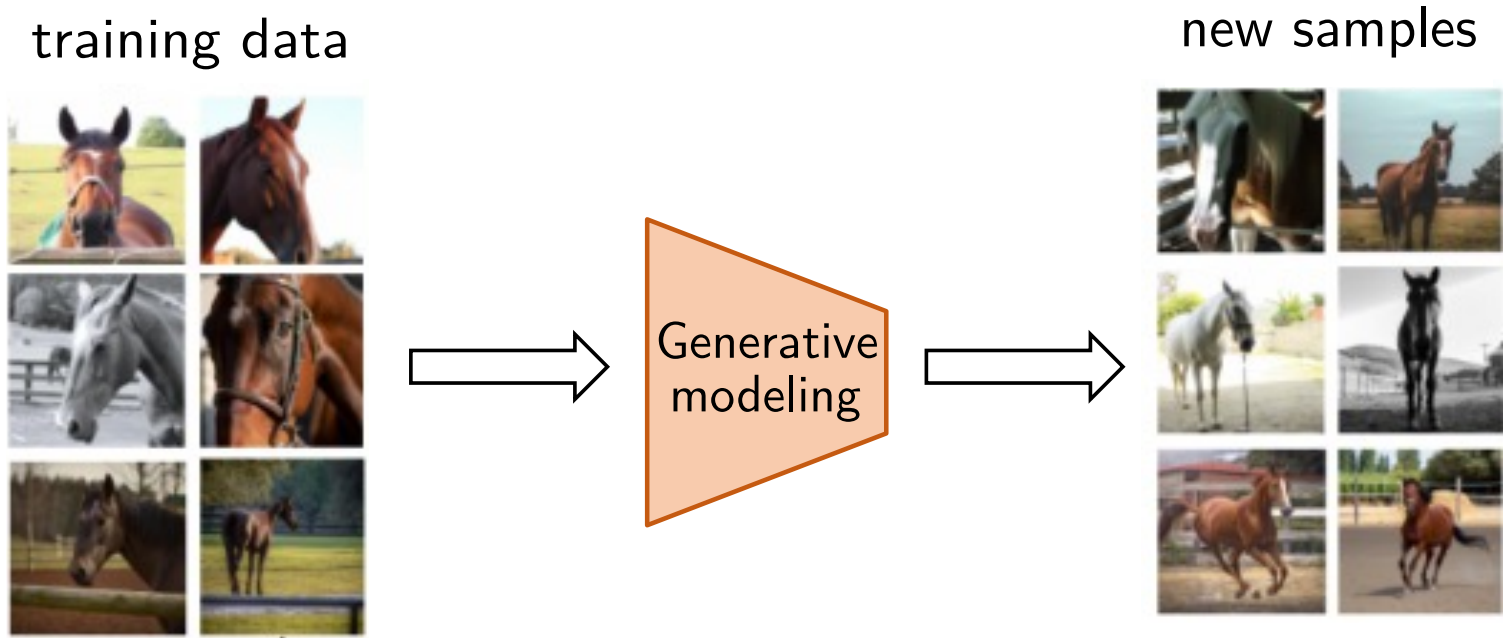
The era of generative AI

Generative models

training data

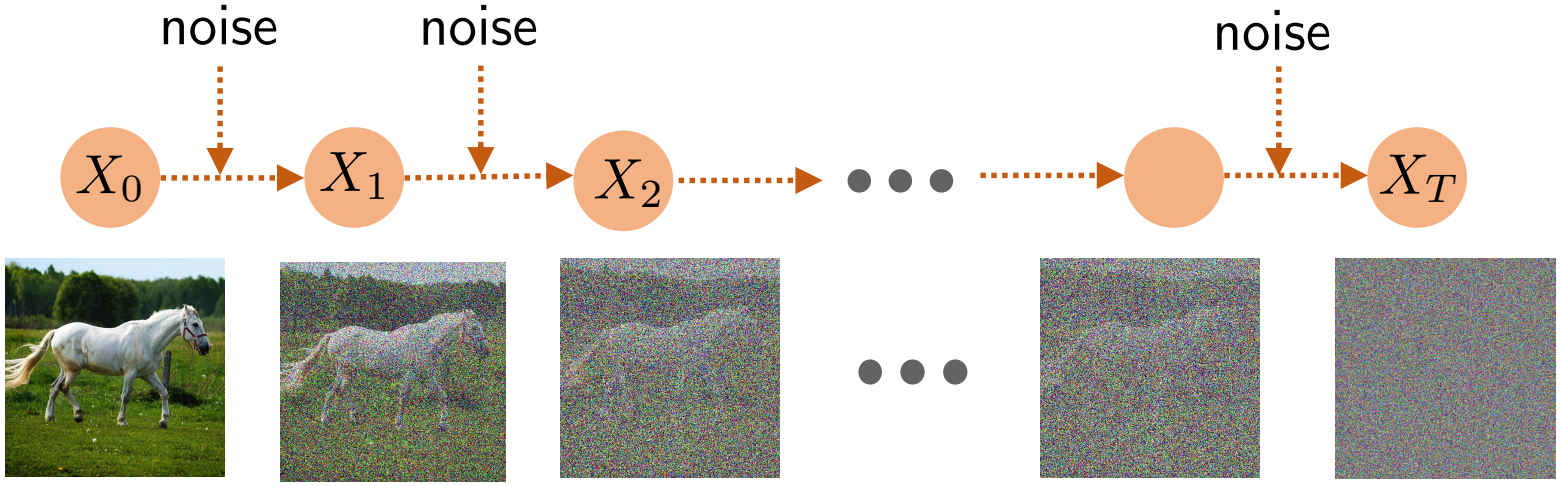
- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a general distribution}}$ $(1 \leq i \leq N)$ in $\mathbb{R}^d$

Generative models

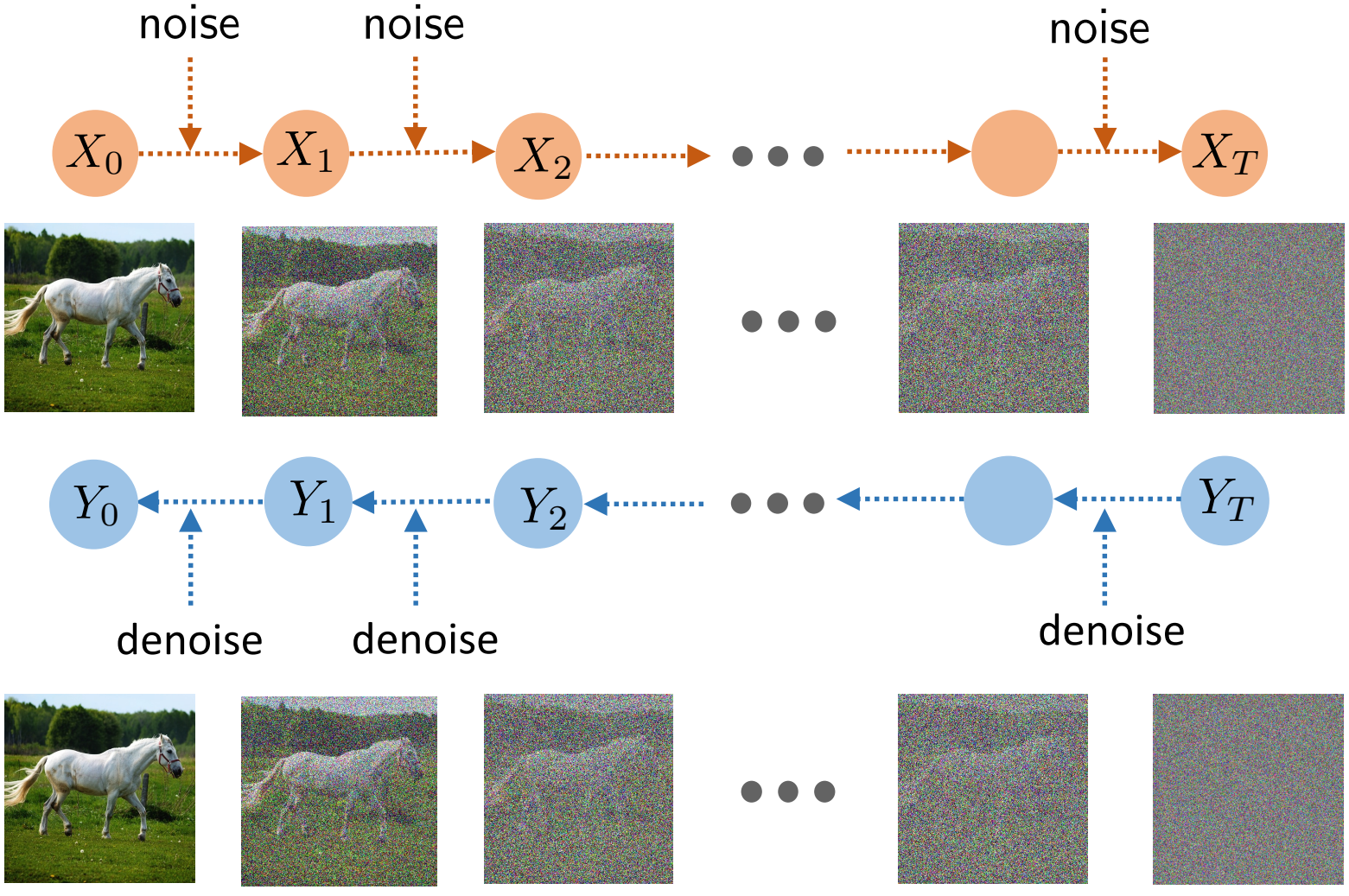


- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a general distribution}}$ $(1 \leq i \leq N)$ in $\mathbb{R}^d$
- Generate new samples $Y \sim p_{\text{data}}$

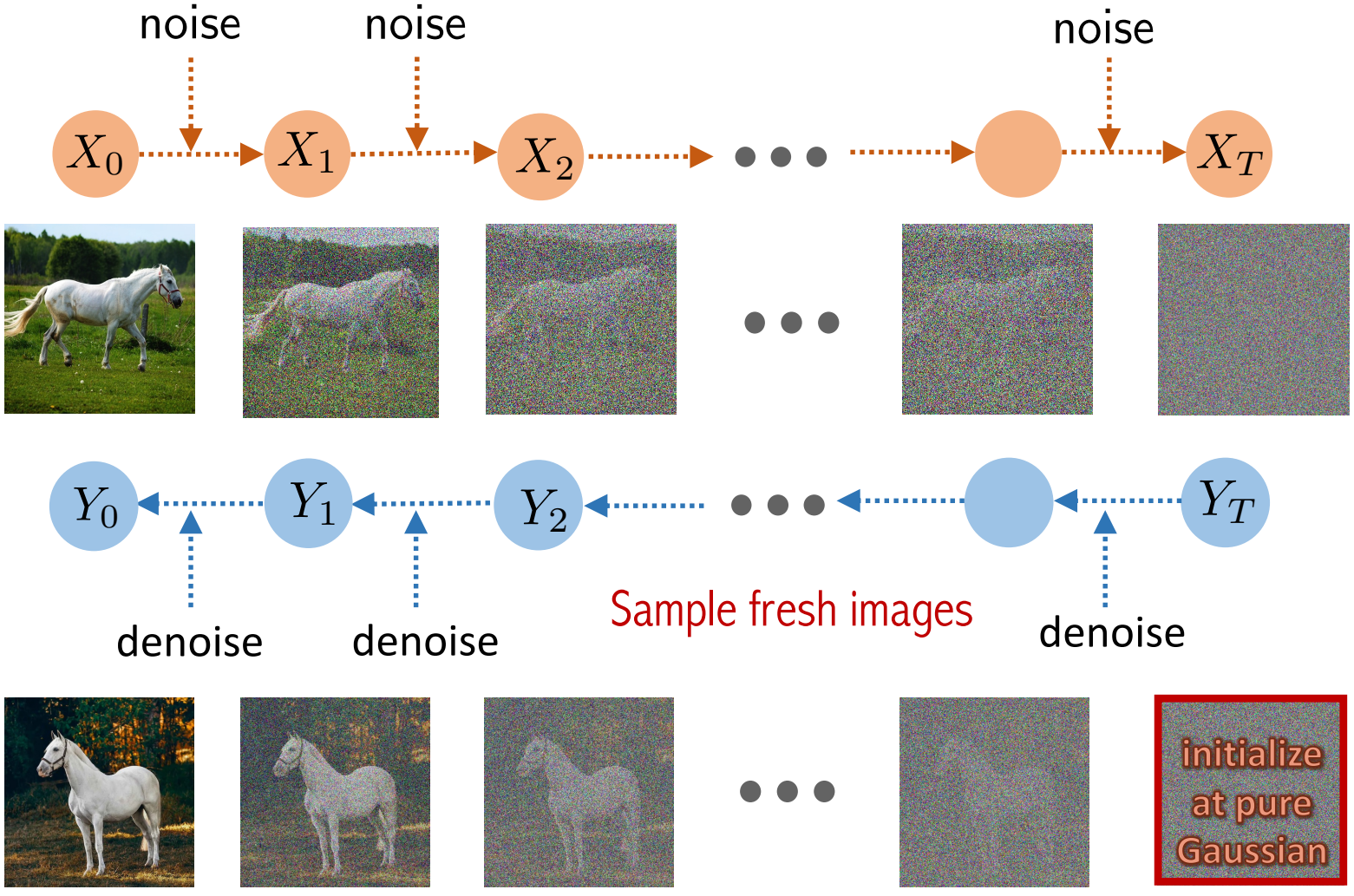
Preliminaries: score-based diffusion models



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



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

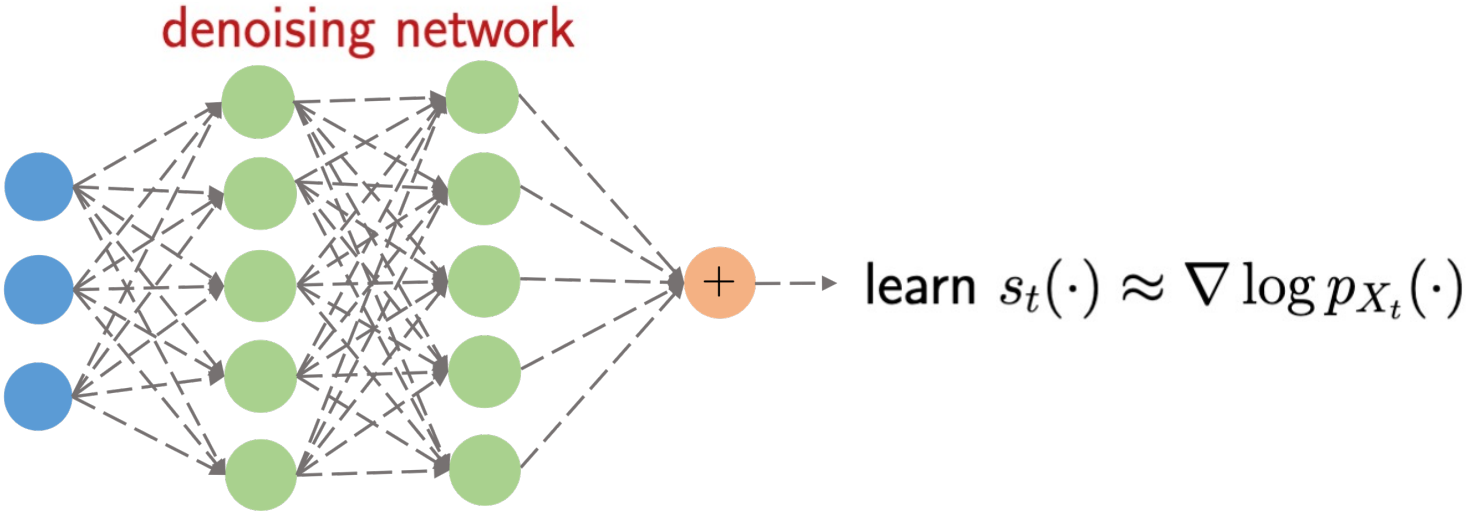


- **forward process:** (progressively) diffuse data into noise
- **reverse process:** convert pure noise into data-like distributions

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

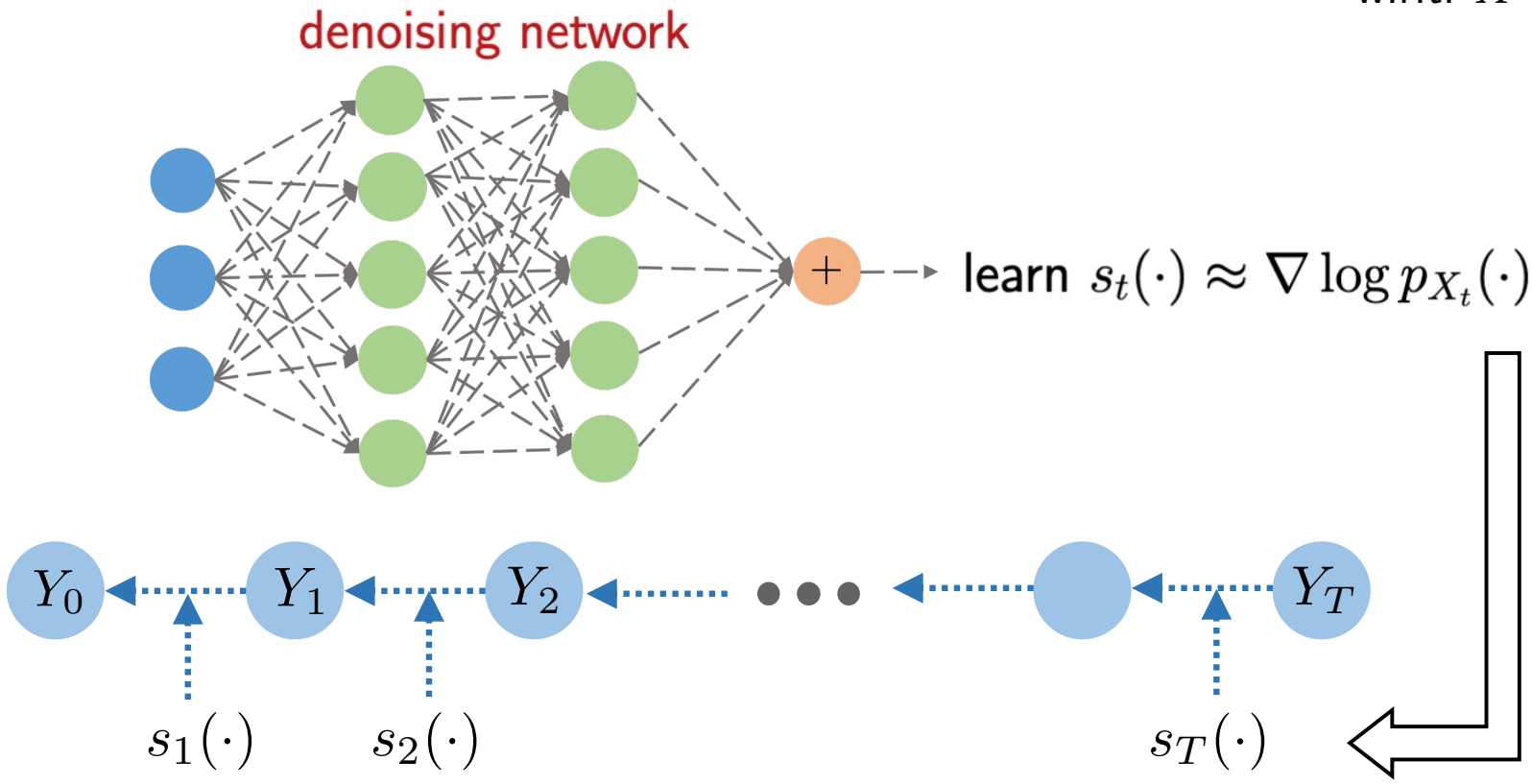
key component: score functions of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$

key component: score functions of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$



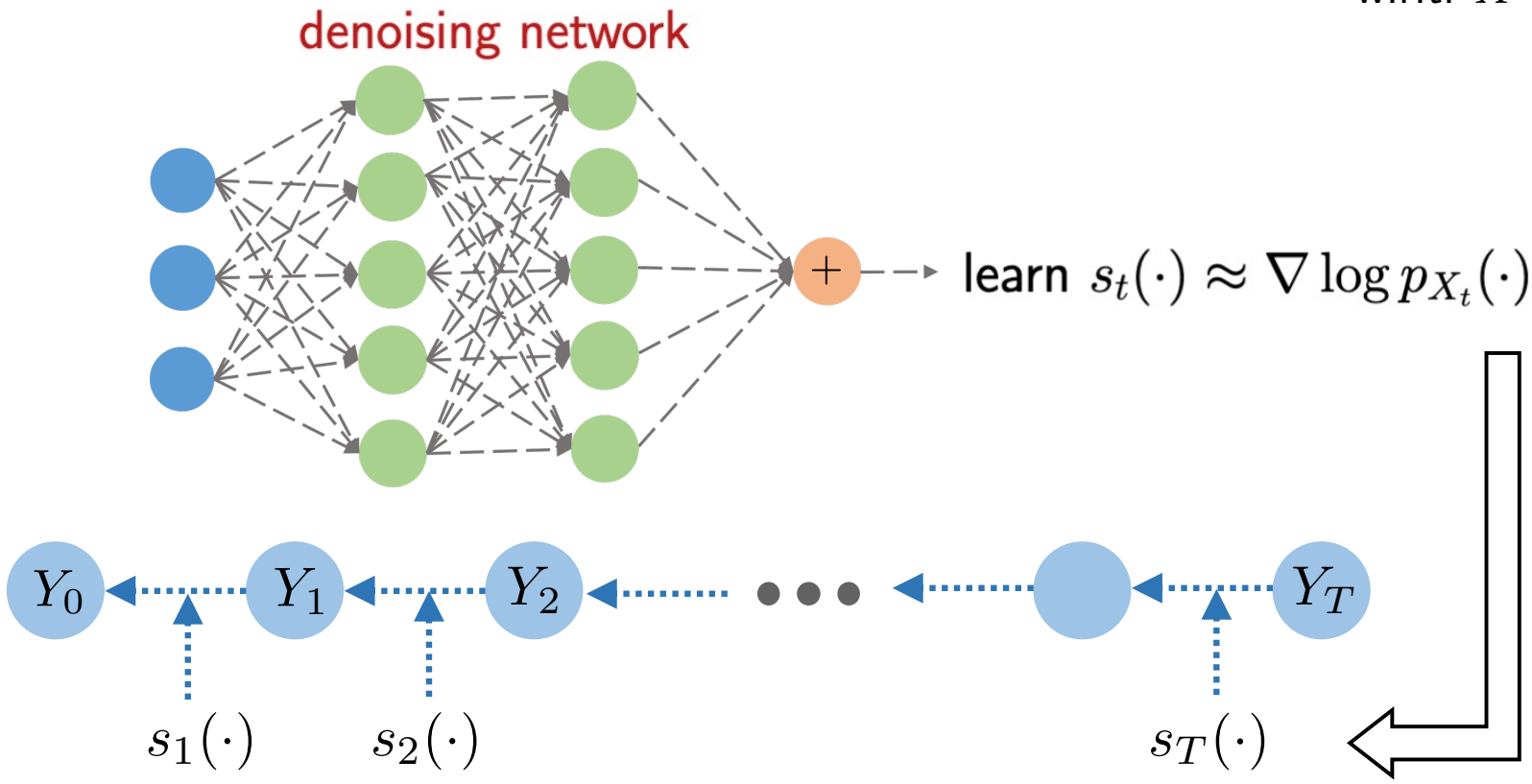
1. **score learning/matching:** learn estimates $s_t(\cdot)$ for $\nabla \log p_{X_t}(\cdot)$

key component: **score functions** of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$



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

key component: **score functions** of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$



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

Two mainstream approaches

— Ho, Jain, Abbeel '20

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

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

Two mainstream approaches

— Ho, Jain, Abbeel '20

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

1. A stochastic sampler: **denoising diffusion probabilistic models** (DDPM)

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

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

- Score estimates: $s_t(x) \approx \nabla \log p_{X_t}(x)$

Two mainstream approaches

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

— Song, Meng, Ermon '20

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

2. A deterministic sampler: based on **probability flow ODE**

or DDIM

Two mainstream approaches

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

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

2. A deterministic sampler: based on **probability flow ODE**

or DDIM

$$Y_T \sim \mathcal{N}(0, I_d)$$
$$Y_{t-1} = \underbrace{\frac{1}{\sqrt{\alpha_t}}\left(Y_t + \frac{1-\alpha_t}{2} s_t(Y_t)\right)}_{\text{purely deterministic}}, \qquad t = T, \cdots, 1$$

- Score estimates: $s_t(x) \approx \nabla \log p_{X_t}(x)$

Interpretations: continuous-time limits

forward process
(marginal: $q_t := p_{X_t}$)

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

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

Interpretations: continuous-time limits

forward process
(marginal: $q_t := p_{X_t}$)

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

$\Big\|$ marginals

DDPM-type
stochastic sampler
(time-reversed SDE, Anderson '82)

$$Y_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\Big(Y_t + \beta_t \nabla \log q_t(Y_t)\Big) + \sqrt{\frac{\beta_t}{1-\beta_t}}\mathcal{N}(0, I_d)$$
$$\Longrightarrow \quad \mathrm{d}Y_t = \Big(-\frac{1}{2}\beta(t) Y_t - \beta(t)\nabla \log q_t(Y_t)\Big)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\widetilde{W}_t \quad \text{(reversed)}$$

Interpretations: continuous-time limits

forward process
(marginal: $q_t := p_{X_t}$)

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

|| marginals

DDPM-type
stochastic sampler
(time-reversed SDE, Anderson '82)

$$Y_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\Big(Y_t + \beta_t \nabla \log q_t(Y_t)\Big) + \sqrt{\frac{\beta_t}{1-\beta_t}}\mathcal{N}(0, I_d)$$
$$\Longrightarrow \quad \mathrm{d}Y_t = \Big(-\frac{1}{2}\beta(t) Y_t - \beta(t)\nabla \log q_t(Y_t)\Big)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\widetilde{W}_t \quad \text{(reversed)}$$

|| marginals

deterministic sampler
(probability flow ODE)

$$Y_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(Y_t + \frac{\beta_t}{2}\nabla \log q_t(Y_t)\right)$$
$$\Longrightarrow \quad \mathrm{d}Y_t = \left(-\frac{1}{2}\beta(t) Y_t - \frac{1}{2}\beta(t)\nabla \log q_t(Y_t)\right)\mathrm{d}t \quad \text{(reversed)}$$

Diffusion-based sampling is often slow

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

— *Song, Meng, Ermon '20*

Diffusion-based sampling is often slow

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

— *Song, Meng, Ermon '20*

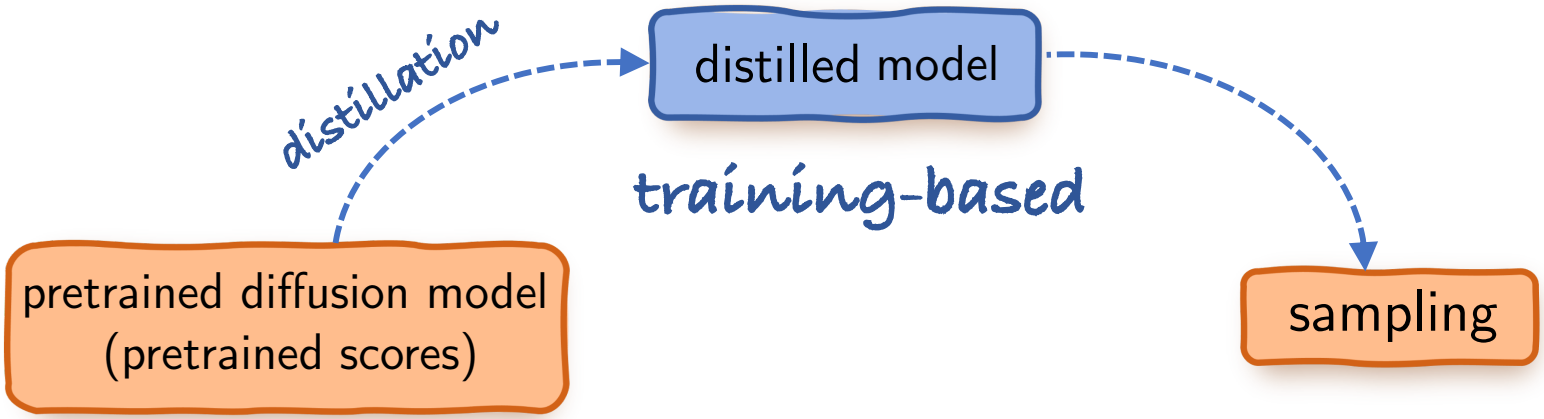
Understanding and improving convergence rate are important!

Part 1: acceleration

— with application to inverse problems

"Provable acceleration for diffusion models under minimal assumptions," G. Li*, C. Cai*, arXiv:2410.23285, 2024

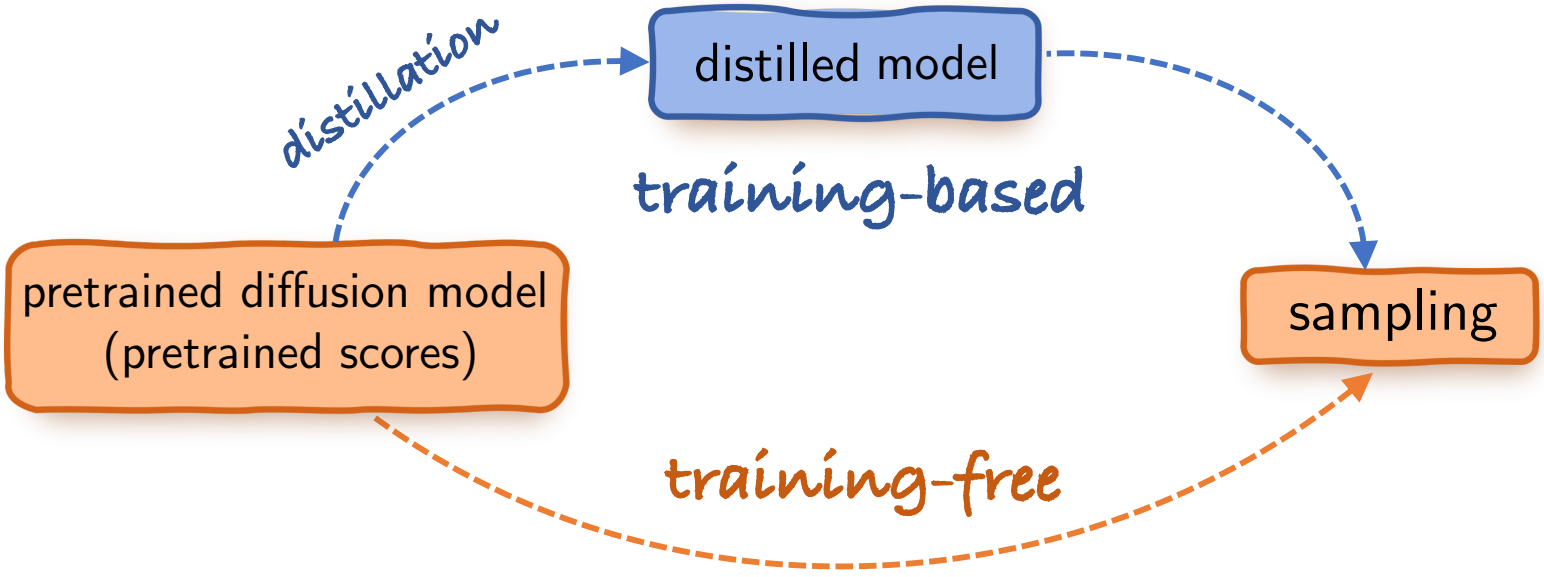
"Improving Diffusion-based Inverse Algorithms under Few-Step Constraint via Learnable Linear Extrapolation," J. Zhang, Z. Liu, L. Yan, G. Li, Y. Gu, NeurIPS, 2025



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

  requires additional training

  - e.g., progressive distillation (Salimans et al. '22), consistency model (Song et al. '23), ...



- **Training-free:** directly invoke pre-trained diffusion models (particularly 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 converges provably faster?

Motivation from high-order discretization

$$X_t \overset{\mathrm{d}}{=} \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\,\mathcal{N}(0, I_d) \qquad \text{with } \overline{\alpha}_t := \prod_{k=1}^{t} (1-\beta_k)$$

Motivation from high-order discretization

$$X_t \overset{\mathrm{d}}{=} \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\,\mathcal{N}(0, I_d) \qquad \text{with } \overline{\alpha}_t := \prod_{k=1}^{t} (1-\beta_k)$$

General form for $0 < \gamma < 1$:

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

Motivation from high-order discretization

$$X_t \overset{\mathrm{d}}{=} \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\,\mathcal{N}(0, I_d) \qquad \text{with } \overline{\alpha}_t := \prod_{k=1}^{t}(1-\beta_k)$$

General form for $0 < \gamma < 1$:

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

$$\Longrightarrow \quad X(\overline{\alpha}_t) \overset{\mathrm{d}}{=} X_t, \;\; s^\star_{\overline{\alpha}_t} = s^\star_t$$

A key reversed relation $X(\overline{\alpha}_t) \to X(\overline{\alpha}_{t-1})$:

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s^\star_\gamma\big(X(\gamma)\big)\mathrm{d}\gamma$$

Motivation from high-order discretization

$$X_t \overset{\mathrm{d}}{=} \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\,\mathcal{N}(0, I_d) \qquad \text{with } \overline{\alpha}_t := \prod_{k=1}^{t}(1-\beta_k)$$

General form for $0 < \gamma < 1$:

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

$$\Longrightarrow \quad X(\overline{\alpha}_t) \overset{\mathrm{d}}{=} X_t,\ s^\star_{\overline{\alpha}_t} = s^\star_t$$

A key reversed relation $X(\overline{\alpha}_t) \to X(\overline{\alpha}_{t-1})$:

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s^\star_\gamma\big(X(\gamma)\big)\mathrm{d}\gamma$$

"solution" to probability flow ODE

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \mathrm{d}\gamma$$

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \mathrm{d}\gamma$$

Scheme 1: $s^\star_\gamma\big(X(\gamma)\big) \approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$

$$\Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \qquad \text{original DDIM}$$

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \, \mathrm{d}\gamma$$

Scheme 1: $s^\star_\gamma\big(X(\gamma)\big) \approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$

$$\Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \qquad \text{original DDIM}$$

refined approximation?

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \mathrm{d}\gamma$$

Scheme 1: $s^\star_\gamma\big(X(\gamma)\big) \approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$

$$\Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \qquad \text{original DDIM}$$

refined approximation?

$$\begin{aligned} s^\star_\gamma\big(X(\gamma)\big) &\approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) + \frac{\mathrm{d}s^\star_\gamma\big(X(\gamma)\big)}{\mathrm{d}\gamma} (\gamma - \overline{\alpha}_t) \\ &\approx s_t(X_t) + \frac{\gamma - \overline{\alpha}_t}{\overline{\alpha}_t - \overline{\alpha}_{t+1}} \Big(s_t(X_t) - s_{t+1}\big(X_{t+1}\big) \Big) \end{aligned}$$

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \mathrm{d}\gamma$$

Scheme 1: $s^\star_\gamma\big(X(\gamma)\big) \approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$

$$\Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \qquad \text{original DDIM}$$

Scheme 2: $s^\star_\gamma\big(X(\gamma)\big) \approx s_t(X_t) + \frac{\gamma - \overline{\alpha}_t}{\overline{\alpha}_t - \overline{\alpha}_{t+1}} \big(s_t(X_t) - s_{t+1}(X_{t+1})\big)$

$$\begin{aligned} \Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx & \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \\ & + \frac{1}{\sqrt{\alpha_t}} \left(\frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})} \Big(s_t\big(X_t\big) - \sqrt{\alpha_{t+1}} s_{t+1}\big(X_{t+1}\big) \Big) \right) \end{aligned}$$ **Ours**

Motivation from high-order discretization

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}} X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2} \int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} \underbrace{s^\star_\gamma\big(X(\gamma)\big)}_{\text{approximated by?}} \mathrm{d}\gamma$$

Scheme 1: $s^\star_\gamma\big(X(\gamma)\big) \approx s^\star_{\overline{\alpha}_t}\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$

$$\Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \qquad \text{original DDIM}$$

Scheme 2: $s^\star_\gamma\big(X(\gamma)\big) \approx s_t(X_t) + \frac{\gamma - \overline{\alpha}_t}{\overline{\alpha}_t - \overline{\alpha}_{t+1}} \big(s_t(X_t) - s_{t+1}(X_{t+1})\big)$

$$\begin{aligned} \Longrightarrow \quad X(\overline{\alpha}_{t-1}) \approx & \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t) \right) \\ & + \frac{1}{\sqrt{\alpha_t}} \left(\frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})} \Big(s_t\big(X_t\big) - \sqrt{\alpha_{t+1}} s_{t+1}\big(X_{t+1}\big) \Big) \right) \end{aligned}$$ **Ours**

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

Proposed accelerated deterministic sampler

$$Y_t^- = \Phi_t(Y_t), \qquad Y_{t-1} = \Psi_t(Y_t, Y_t^-) \qquad \text{for } t = T, \cdots, 1 \qquad (1)$$

- compute $\underbrace{\text{a midpoint } Y_t^-}_{\text{estimate of } Y_{t+1} \text{ using } Y_t}$; update based on $\underbrace{\text{both } Y_t \text{ and } Y_t^-}_{\text{provide 2nd-order info}}$

Proposed accelerated deterministic sampler

$$Y_t^- = \Phi_t(Y_t), \qquad Y_{t-1} = \Psi_t(Y_t, Y_t^-) \qquad \text{for } t = T, \cdots, 1 \qquad (1)$$

$$\Psi_t(Y_t, Y_t^-) = \frac{1}{\sqrt{\alpha_t}} \Bigg(\underbrace{Y_t + \frac{1-\alpha_t}{2} s_t(Y_t)}_{\text{original DDIM}} \Bigg)$$

- compute $\underbrace{\text{a midpoint } Y_t^-}_{\text{estimate of } Y_{t+1} \text{ using } Y_t}$; update based on $\underbrace{\text{both } Y_t \text{ and } Y_t^-}_{\text{provide 2nd-order info}}$

Proposed accelerated deterministic sampler

$$Y_t^- = \Phi_t(Y_t), \qquad Y_{t-1} = \Psi_t(Y_t, Y_t^-) \qquad \text{for } t = T, \cdots, 1 \qquad (1)$$

$$\Phi_t(Y_t) = \sqrt{\alpha_{t+1}}\left(Y_t - \frac{1-\alpha_{t+1}}{2} s_t(Y_t)\right)$$

$$\Psi_t(Y_t, Y_t^-) = \frac{1}{\sqrt{\alpha_t}}\Bigg(\underbrace{Y_t + \frac{1-\alpha_t}{2} s_t(Y_t)}_{\text{original DDIM}} + \underbrace{\frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})}\left(s_t(Y_t) - \sqrt{\alpha_{t+1}} s_{t+1}(Y_t^-)\right)}_{\text{“momentum”}}\Bigg)$$

- compute $\underbrace{\text{a midpoint } Y_t^-}_{\text{estimate of } Y_{t+1} \text{ using } Y_t}$; update based on $\underbrace{\text{both } Y_t \text{ and } Y_t^-}_{\text{provide 2nd-order info}}$
- 2 score function evaluations per iteration

Proposed accelerated stochastic sampler

$$Y_T \to Y_{T-\frac{1}{2}} \to Y_{T-1} \to \cdots Y_1 \to Y_{\frac{1}{2}} \to Y_0$$

For $t = T, \ldots, 2$:

$$Y_{t-\frac{1}{2}} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + \frac{1-\alpha_t}{2\alpha_t} s_t(Y_t)\Big) + (1-\alpha_t) Z_{t-\frac{1}{2}} \tag{2}$$

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t) s_t(Y_t)\Big) + \frac{\sigma_t}{\sqrt{\alpha_t}} Z_t + \underbrace{\frac{1-\alpha_t}{\sqrt{\alpha_t}} \mathsf{clip}_t\Big[\alpha_t^{3/2} s_{t-1}\big(Y_{t-\frac{1}{2}}\big) - s_t\big(Y_t + (1-\alpha_t) Z_{t-\frac{1}{2}}\big)\Big]}_{\text{"second-order approximation"}} \tag{3}$$

- $\mathsf{clip}_t : \mathbb{R}^d \mapsto \mathbb{R}^d_{\geq 0}$: truncation function
- two score function evaluations per iteration

Minimal assumptions

- **Minimal data assumptions:**

$$\mathbb{E}[\|X_0\|_2] \leq T^{c_R}$$

for arbitrarily large constant $c_R > 0$

Minimal assumptions

- **Minimal data assumptions:**

$$\mathbb{E}[\|X_0\|_2] \leq T^{c_R}$$

for arbitrarily large constant $c_R > 0$

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

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

Convergence theory: DDPM stochastic sampler

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

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t)\Big) + \sqrt{\frac{1-\alpha_t}{\alpha_t}}\mathcal{N}(0, I_d), \qquad t = T, \cdots, 1 \quad (4)$$

Theorem 1 (Li & Yan '24)

The DDPM-type stochastic sampler (4) *obeys*

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim \frac{d\log^3 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}$$

Convergence theory: DDPM stochastic sampler

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

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(Y_t + (1-\alpha_t)s_t(Y_t)\Big) + \sqrt{\frac{1-\alpha_t}{\alpha_t}}\mathcal{N}(0, I_d), \qquad t = T, \cdots, 1 \quad (4)$$

Theorem 1 (Li & Yan '24)

The DDPM-type stochastic sampler (4) *obeys*

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d\log^3 T}{T} + \varepsilon_{\mathsf{score}}\sqrt{\log T}$$

- **iteration complexity** $\widetilde{O}(d/\varepsilon)$ to yield TV dist $\leq \varepsilon$
- **stability:** $\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \propto$ error measure $\varepsilon_{\mathsf{score}}$

Main result: accelerated stochastic sampler

Theorem 2 (Li, Cai '24)

The proposed sampler satisfies (up to log factors)

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim \frac{d^{5/2}}{T^2} + \varepsilon_{\mathsf{score}}$$

Main result: accelerated stochastic sampler

Theorem 2 (Li, Cai '24)

The proposed sampler satisfies (up to log factors)

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim \frac{d^{5/2}}{T^2} + \varepsilon_{\mathsf{score}}$$

- $\underbrace{\textbf{iteration complexity}}_{\text{to yield TV dist} \le \varepsilon}$: $\widetilde{O}\big(d^{5/4}/\sqrt{\varepsilon}\big)$ for small enough ε

Main result: accelerated stochastic sampler

Theorem 2 (Li, Cai '24)

The proposed sampler satisfies (up to log factors)

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim \frac{d^{5/2}}{T^2} + \varepsilon_{\mathsf{score}}$$

- $\underbrace{\textbf{iteration complexity}}_{\text{to yield TV dist} \le \varepsilon}$: $\widetilde{O}(d^{5/4}/\sqrt{\varepsilon})$ for small enough ε
- **mild data assumption**: finite second-order moment (no need of smoothness)

Main result: accelerated stochastic sampler

Theorem 2 (Li, Cai '24)

The proposed sampler satisfies (up to log factors)

$$\mathsf{TV}(p_{X_1}, p_{Y_1}) \lesssim \frac{d^{5/2}}{T^2} + \varepsilon_{\mathsf{score}}$$

- $\underbrace{\textbf{iteration complexity}}_{\text{to yield TV dist} \le \varepsilon}$: $\widetilde{O}(d^{5/4}/\sqrt{\varepsilon})$ for small enough ε
- **mild data assumption**: finite second-order moment (no need of smoothness)
- **minimal score estimation**: L^2 score error suffices (no need of Jacobians)

Comparison with prior works

Additional assumptions:

1. structured data distribution:

	iteration complexity	requirement
Li '24	$Ld^{1/3}/\varepsilon^{2/3}$	1-order Lipschitz score
Huang '24	$(Ld)^{1+1/p}/\varepsilon^{1/p}$	$(p+1)$-order Lipschitz score
Huang '24	$L^4d^2/\varepsilon^{2/p}$	$(p+1)$-order Lipschitz score

Comparison with prior works

Additional assumptions:

1. structured data distribution:

	iteration complexity	requirement
Li '24	$Ld^{1/3}/\varepsilon^{2/3}$	1-order Lipschitz score
Huang '24	$(Ld)^{1+1/p}/\varepsilon^{1/p}$	$(p+1)$-order Lipschitz score
Huang '24	$L^4d^2/\varepsilon^{2/p}$	$(p+1)$-order Lipschitz score

2. higher-order score estimation:

	iteration complexity	requirement
Li '24	$d^3/\sqrt{\varepsilon}$	accurate Jacobian estimate

Comparison with prior works

Additional assumptions:

1. structured data distribution:

	iteration complexity	requirement
Li '24	$Ld^{1/3}/\varepsilon^{2/3}$	1-order Lipschitz score
Huang '24	$(Ld)^{1+1/p}/\varepsilon^{1/p}$	$(p+1)$-order Lipschitz score
Huang '24	$L^4d^2/\varepsilon^{2/p}$	$(p+1)$-order Lipschitz score

2. higher-order score estimation:

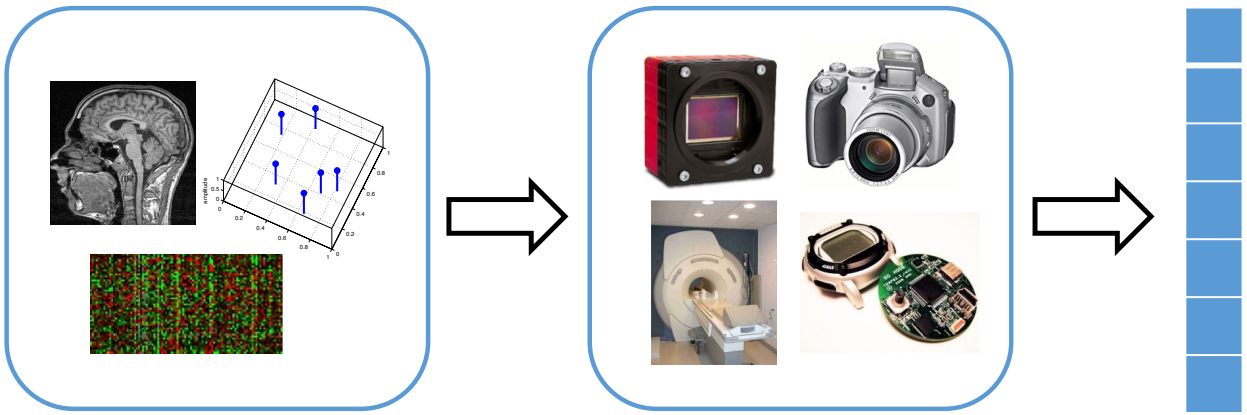
	iteration complexity	requirement
Li '24	$d^3/\sqrt{\varepsilon}$	accurate Jacobian estimate

We design a provable training-free accelerated sampler under minimal assumptions on data distribution and score estimation

Inverse problems

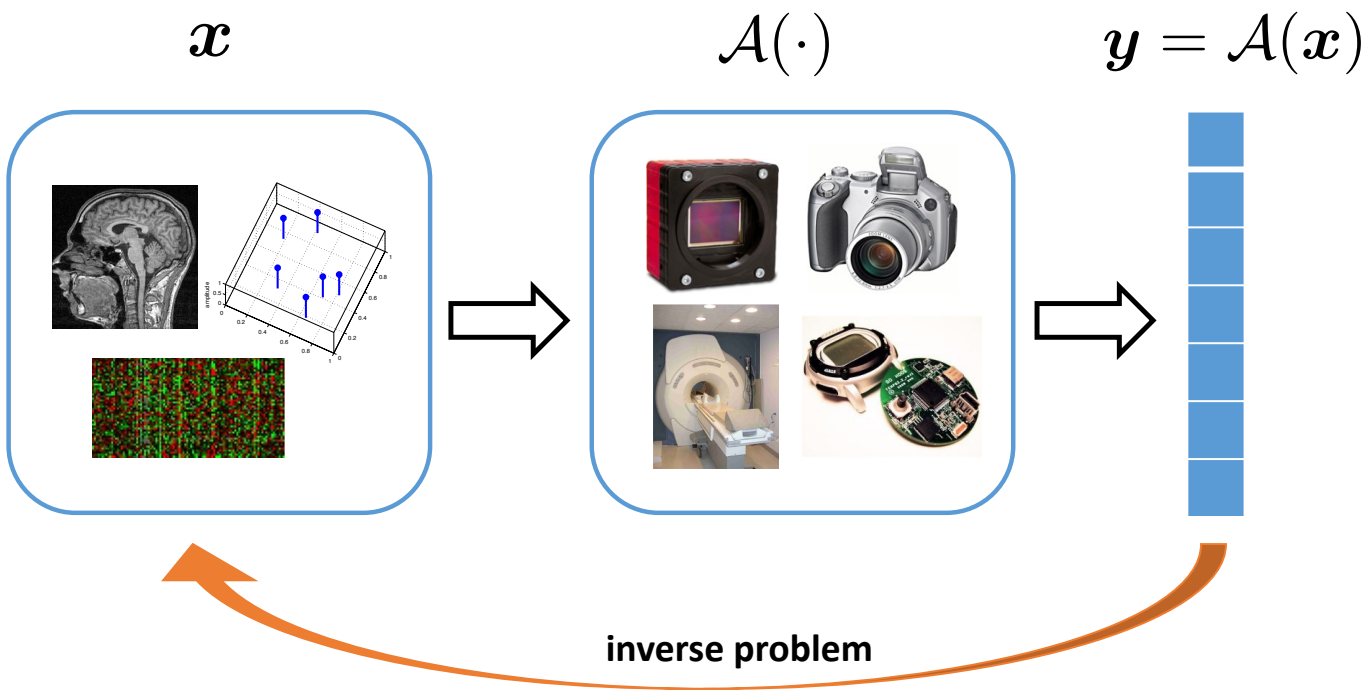
Forward model: we interrogate the signal of interest $\boldsymbol{x}$ through forward model $\mathcal{A}$ and make measurements $\boldsymbol{y}$.

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



Inverse problems

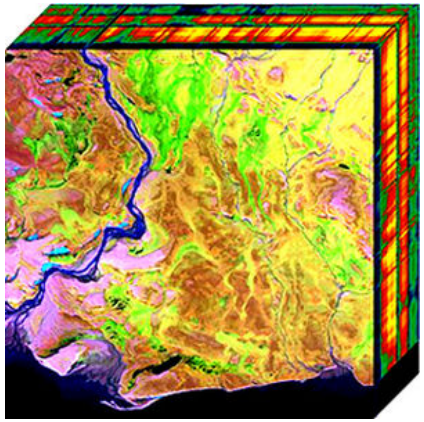
Forward model: we interrogate the signal of interest $\boldsymbol{x}$ through forward model $\mathcal{A}$ and make measurements $\boldsymbol{y}$.



Inverse problem: recover the signal of interest $\boldsymbol{x}$ from $\boldsymbol{y}$.

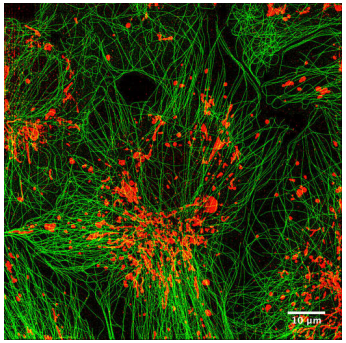
Ubiquitous, but often ill-posed

healthcare



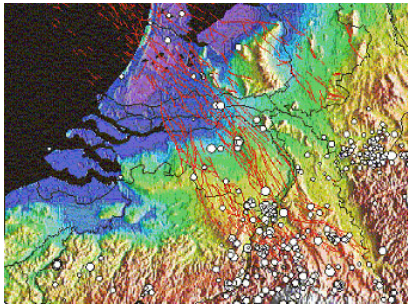
hyperspectral

Internet traffic



microscopy

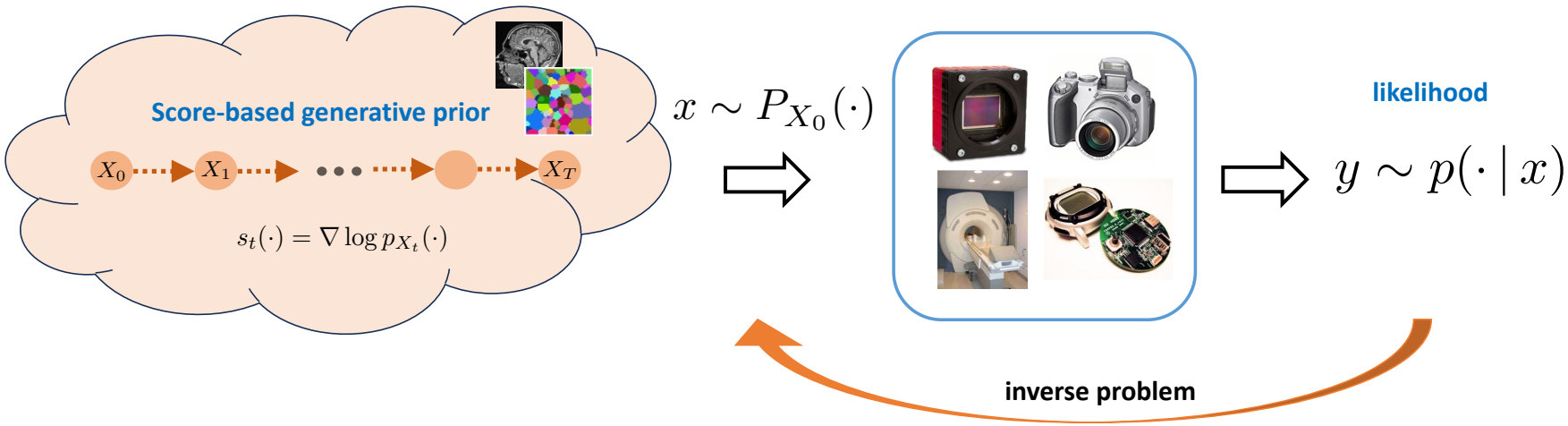
Radio astronomy



seismic imaging

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

Score-based diffusion model for inverse problems



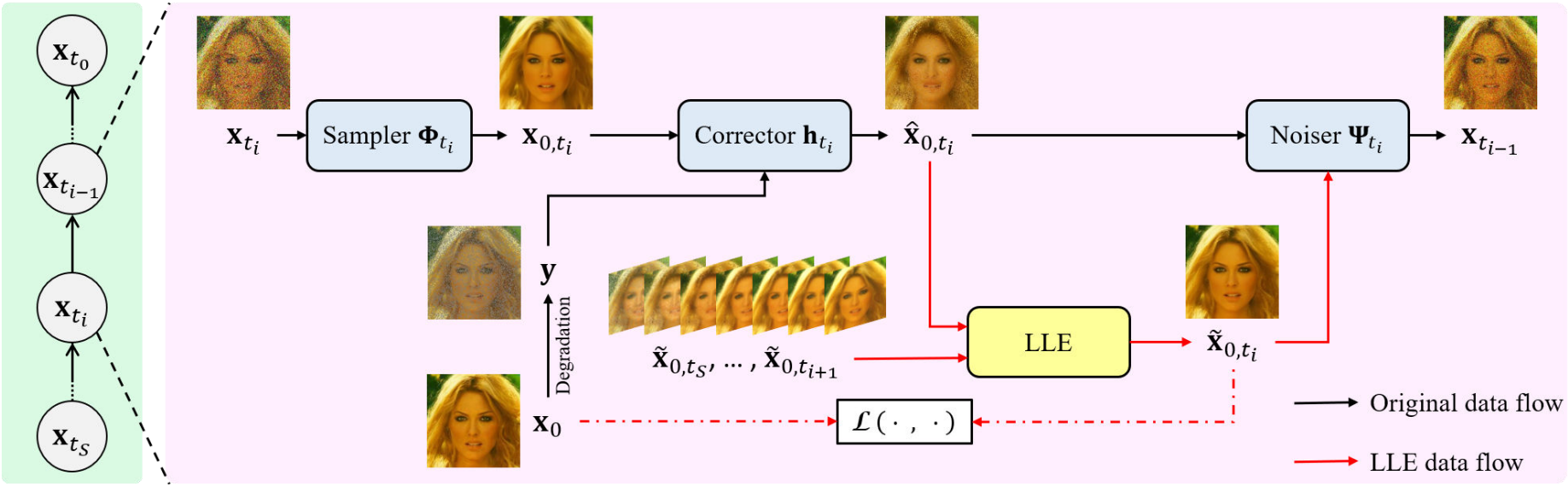
Posterior sampling: sample from

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

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

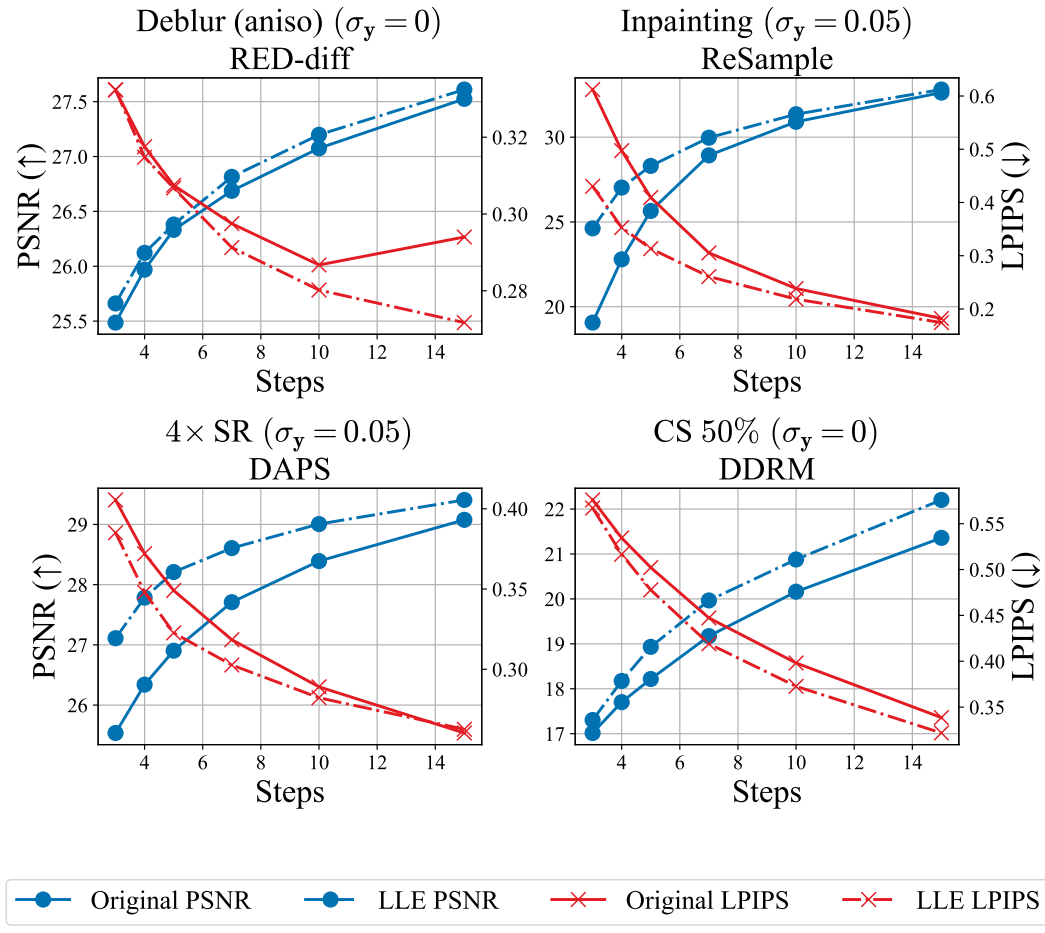
Accelerated diffusion-based inverse algorithms

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 2: diffusion language models

— an information-theoretic rate

"A Convergence Theory for Diffusion Language Models: An Information-Theoretic Perspective," G. Li*, C. Cai*, NeurIPS, 2025

Autoregressive (AR) modeling

Token sequence $x = (x^{(1)}, \ldots, x^{(L)})$:

$$p(x) = p(x^{(1)}) \prod_{i=2}^{L} p(x^{(i)} \mid x^{(1)}, \ldots, x^{(i-1)}),$$

- one-by-one generation: slow speed

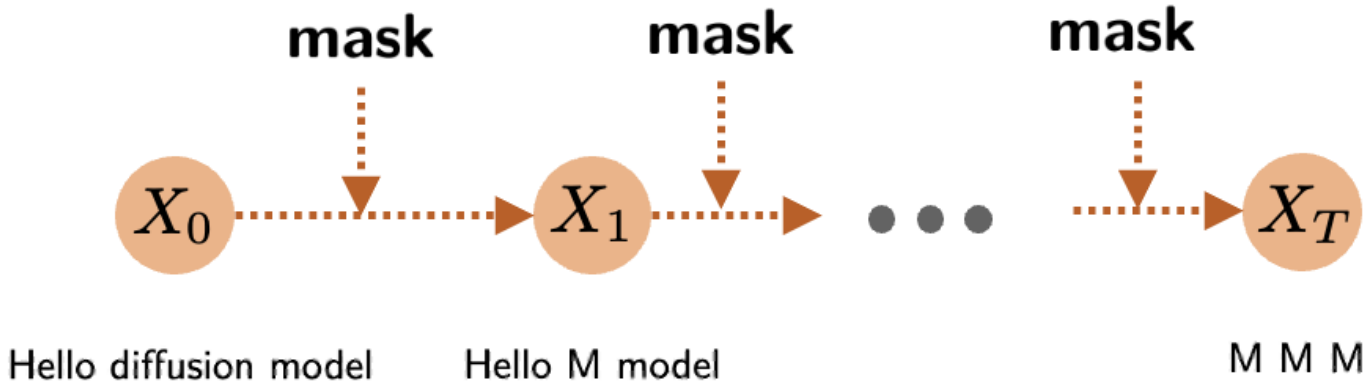
Autoregressive (AR) modeling

Token sequence $x = (x^{(1)}, \ldots, x^{(L)})$:

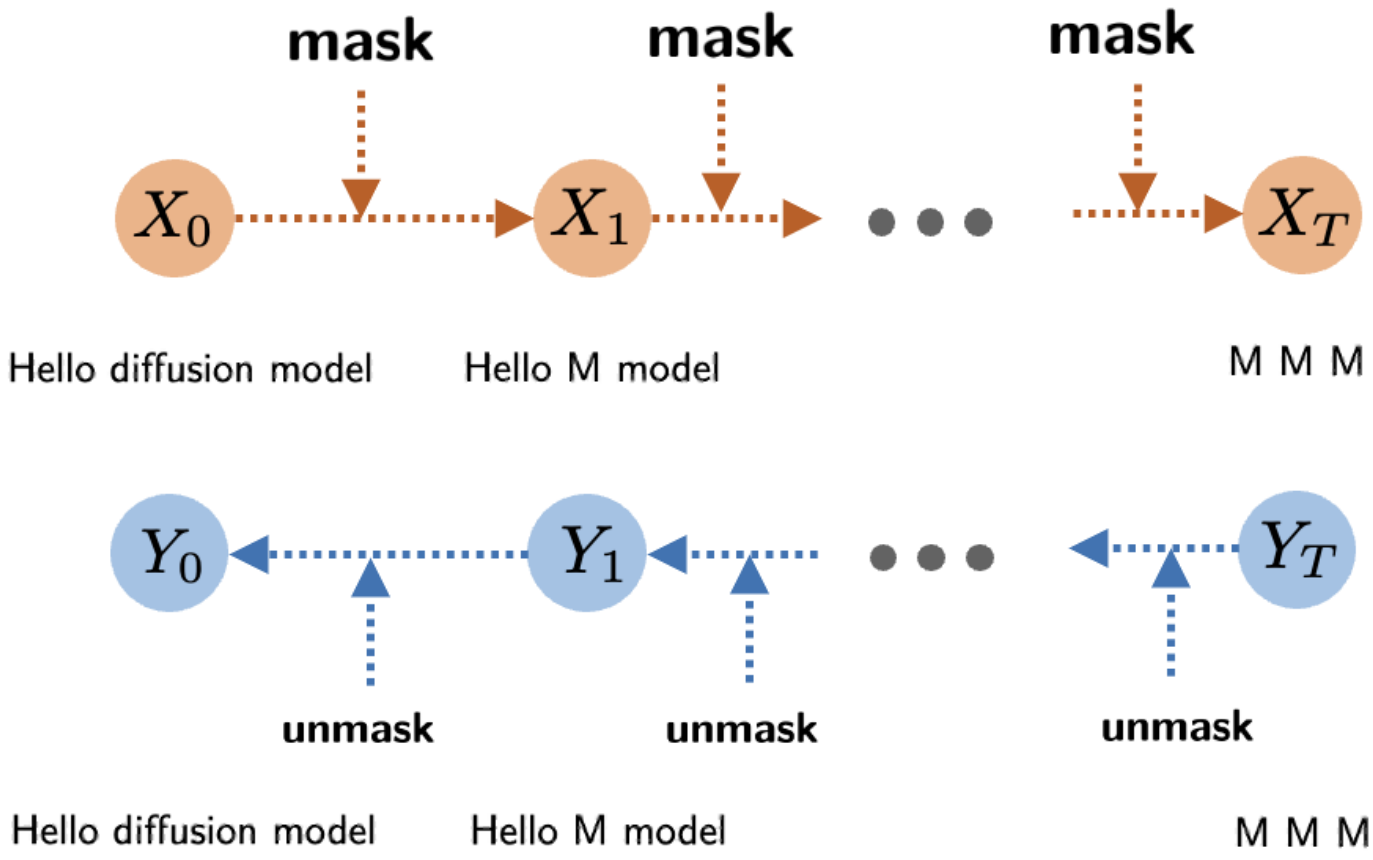
$$p(x) = p(x^{(1)}) \prod_{i=2}^{L} p(x^{(i)} \mid x^{(1)}, \ldots, x^{(i-1)}),$$

- one-by-one generation: slow speed
- left-to-right order: forward-only generation

Preliminaries: diffusion language models



- **forward process:** (progressively) mask data



- **forward process:** (progressively) mask data
- **reverse process:** convert masks into data-like distributions

Goal: $Y_0 \overset{\mathrm{d}}{\approx} X_0$

Mask predictor learning

Given mask size schedule $\{s_t\}_{t=1}^{L}$ with $\sum_t s_t = L$, generate random mask index sets $\{M_t\}_{t=0}^{T}$ s.t.

- $\varnothing = M_0 \subseteq \cdots \subseteq M_T = [L]$
- $M_t \setminus M_{t-1}$: random subset of M_{t-1}^{c} with size s_t

Mask predictor learning

Given mask size schedule $\{s_t\}_{t=1}^L$ with $\sum_t s_t = L$, generate random mask index sets $\{M_t\}_{t=0}^T$ s.t.

- $\varnothing = M_0 \subseteq \cdots \subseteq M_T = [L]$
- $M_t \setminus M_{t-1}$: random subset of M_{t-1}^{c} with size s_t

Find $\widehat{p}$ that solves:

$$\min_{p=\prod_{i=1}^L p_i} -\mathbb{E}_{\tau, X_0, M_\tau}\left[\frac{L}{|M_\tau|}\sum_{i\in M_\tau}\log p_i(X_0^{(i)} \mid X_\tau)\right], \tag{5}$$

- random step $\tau \in \{1, \ldots, T\}$ s.t. $\boldsymbol{P}\{\tau = t\} = s_t/L$
- training data $X_0 \sim p_{\mathsf{data}}$
- masked sequence $X_\tau = \mathcal{P}_{M_\tau^{\mathsf{c}}}(X_0)$

Sampling procedure

- Initialize $Y_T \leftarrow (\mathsf{M}, \ldots, \mathsf{M})$ and $M_T \leftarrow [L]$
- For $t = T, \ldots, 1$:

$$\widehat{X}_t \sim \widehat{p}\,(\cdot \mid Y_t)$$

$$Y_{t-1} \leftarrow \underbrace{\mathcal{P}_{M_t^{\mathsf{c}}}(Y_t)}_{\text{already unmasked}} + \underbrace{\mathcal{P}_{M_t \setminus M_{t-1}}(\widehat{X}_t)}_{\text{unmasked at step } t}$$

Output: sample Y_0

Prior theories on language models

Perplexity to measure the performance:

$$\exp\Big(-\frac{1}{L}\mathbb{E}_{x\sim p_{X_0}}\big[\log p_{Y_0}(x)\big]\Big) = \exp\Big(\frac{1}{L}\big[\mathsf{KL}(p_{X_0}\,\|\,p_{Y_0}) - H(X_0)\big]\Big)$$

Prior theories on language models

Perplexity to measure the performance:

$$\exp\Big(-\frac{1}{L}\mathbb{E}_{x\sim p_{X_0}}\big[\log p_{Y_0}(x)\big]\Big) = \exp\Big(\frac{1}{L}\big[\mathsf{KL}(p_{X_0}\,\|\,p_{Y_0}) - H(X_0)\big]\Big)$$

- *AR model*: $T = L$

Prior theories on language models

Perplexity to measure the performance:

$$\exp\Big(-\frac{1}{L}\mathbb{E}_{x\sim p_{X_0}}\big[\log p_{Y_0}(x)\big]\Big) = \exp\Big(\frac{1}{L}\big[\mathsf{KL}(p_{X_0}\,\|\,p_{Y_0}) - H(X_0)\big]\Big)$$

- *AR model*: $T = L$
- *Chen, Ying, '24 & Zhang, Chen, Gu, '24*: $T > L$

Prior theories on language models

Perplexity to measure the performance:

$$\exp\Big(-\frac{1}{L}\mathbb{E}_{x\sim p_{X_0}}\big[\log p_{Y_0}(x)\big]\Big) = \exp\Big(\frac{1}{L}\big[\mathsf{KL}(p_{X_0}\,\|\,p_{Y_0}) - H(X_0)\big]\Big)$$

- *AR model*: $T = L$
- *Chen, Ying, '24 & Zhang, Chen, Gu, '24*: $T > L$
- *Feng, et al. '25*: $((n-1)/T)^{1/n} L \log|\mathcal{X}|$ error bound for n-gram language model with $n \ll \log L$

Training error

Training error of mask predictor $\widehat{p} = \prod_{i=1}^{L} \widehat{p}_i$:

$$\varepsilon_{\text{train}} := \mathbb{E}_{\tau, X_0, M_\tau}\left[\frac{L}{|M_\tau|} \sum_{i \in M_\tau} \log p_i^\star(X_0^{(i)} \mid X_\tau)\right] - \mathbb{E}_{\tau, X_0, M_\tau}\left[\frac{L}{|M_\tau|} \sum_{i \in M_\tau} \log \widehat{p}_i(X_0^{(i)} \mid X_\tau)\right]$$

where $p^\star$ is the minimizer of the training loss

$\varepsilon_{\text{train}}$: likelihood gap caused by imperfect training

Main result: diffusion language model

Theorem 3 (Li, Cai '25)

The output Y_0 of the diffusion language model satisfies

$$\mathbb{E}_M\left[\mathsf{KL}(p_{X_0} \,\|\, p_{Y_0|M})\right] \le \frac{2^{\lceil \log_2 s_{\max} \rceil} - 1}{L} \sum_{i=1}^{L} I(X_0^{(i)}; X_0^{(-i)}) + \varepsilon_{\text{train}}.$$

- **masking strategy:** $s_{\max}$ is the maximum mask size
- **token dependency:** mutual information between each token and the rest of the sequence
- **training error:** imperfect mask predictions

Main result: diffusion language model

For balanced mask size schedule:

Theorem 4 (Li, Cai '25)

Suppose $\frac{1}{T}\sum_{t=1}^{T} s_t \asymp s_{\max}$. *Then*

$$\mathbb{E}_M\left[\mathsf{KL}(p_{X_0} \,\|\, p_{Y_0|M})\right] - \varepsilon_{\mathsf{train}} \asymp \frac{1}{T} \times \textit{info-term}$$

Main result: diffusion language model

For balanced mask size schedule:

Theorem 4 (Li, Cai '25)

Suppose $\frac{1}{T}\sum_{t=1}^{T} s_t \asymp s_{\max}$. *Then*

$$\mathbb{E}_M\left[\mathsf{KL}(p_{X_0} \,\|\, p_{Y_0|M})\right] - \varepsilon_{\mathsf{train}} \asymp \frac{1}{T} \times \textit{info-term}$$

- information term $\leq \sum_{i=1}^{L} I(X_0^{(i)}; X_0^{(-i)}) \leq L \log |\mathcal{X}|$

Main result: diffusion language model

For balanced mask size schedule:

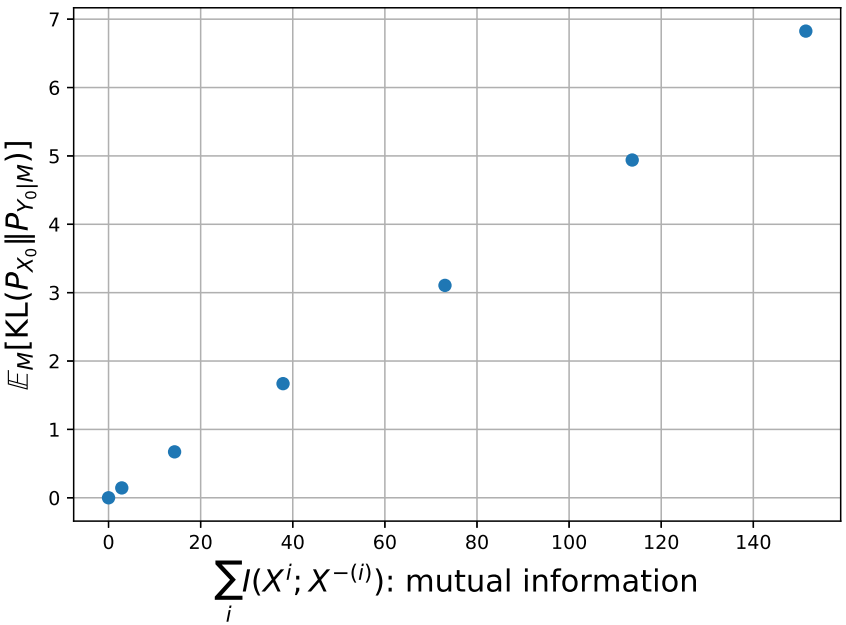
Theorem 4 (Li, Cai '25)

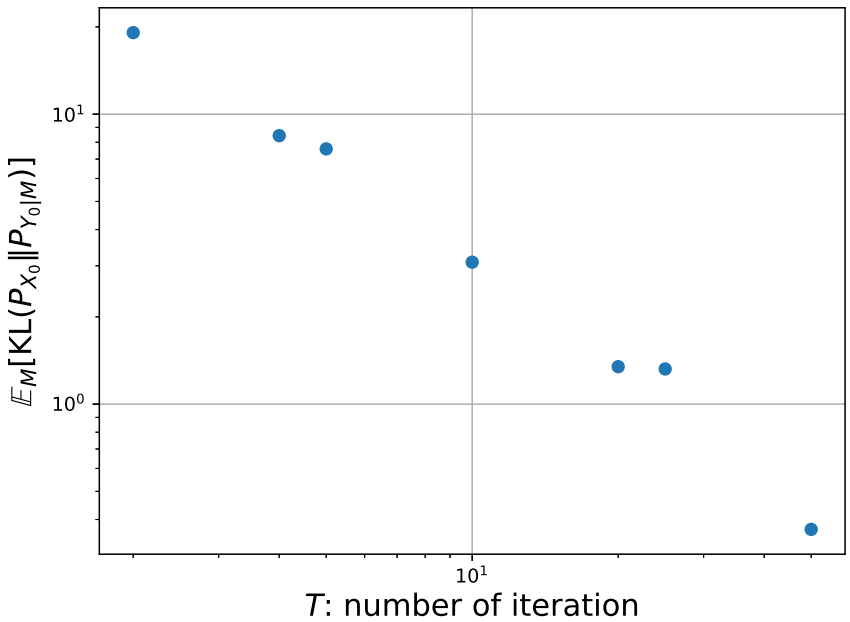
Suppose $\frac{1}{T}\sum_{t=1}^{T} s_t \asymp s_{\max}$. *Then*

$$\mathbb{E}_M\left[\mathsf{KL}(p_{X_0} \,\|\, p_{Y_0|M})\right] - \varepsilon_{\mathsf{train}} \asymp \frac{1}{T} \times \textit{info-term}$$

- information term $\leq \sum_{i=1}^{L} I(X_0^{(i)}; X_0^{(-i)}) \leq L \log |\mathcal{X}|$
- *Chen, Ying, '24 & Zhang, Chen, Gu, '24*: update < 1 token (on average) per step.
- *Feng, et al. '25*: $((n-1)/T)^{1/n} L \log |\mathcal{X}|$ error bound for n-gram language model with $n \ll \log L$

Numerical experiments





- Sampling error: convergence rate $1/T$ with linear dependence on mutual information among tokens.

Concluding remarks

- accelerate sampling processes provably (using only score functions)
- establish tight convergence guarantees for diffusion language models from an information-theoretic perspective

Paper:

"$O(d/T)$ Convergence Theory for Diffusion Probabilistic Models under Minimal Assumptions," G. Li*, Y. Yan*, ICLR, 2025

"Provable acceleration for diffusion models under minimal assumptions," G. Li*, C. Cai*, arXiv:2410.23285, 2024

"Improving Diffusion-based Inverse Algorithms under Few-Step Constraint via Learnable Linear Extrapolation," J. Zhang, Z. Liu, L. Yan, G. Li, Y. Gu, NeurIPS, 2025

"A Convergence Theory for Diffusion Language Models: An Information-Theoretic Perspective," G. Li*, C. Cai*, NeurIPS, 2025