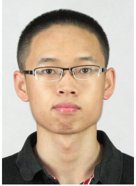
Fantastic Diffusion Models and Where to Apply Them

Yuejie Chi

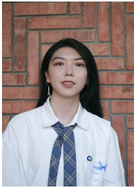
Carnegie Mellon University

IEEE Information Theory Workshop
November 2024



Gen Li
CUHK

Xingyu Xu
CMU



Yu Huang
UPenn

Timofey Efimov
CMU

Yuting Wei
UPenn

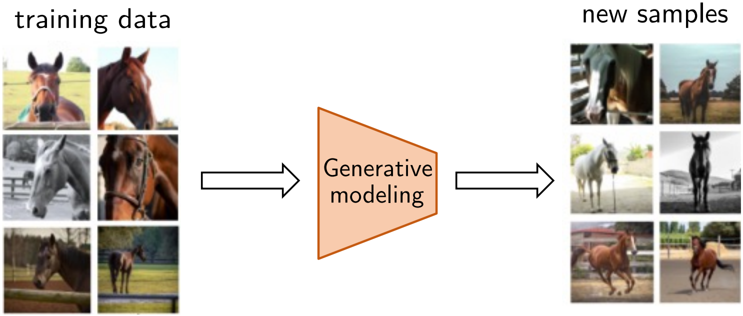
Yuxin Chen
UPenn

Generative models

training data

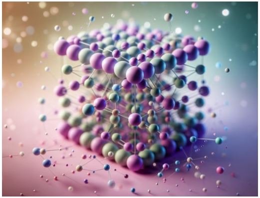
- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a general distribution}}$ $(1 \le i \le N)$ in $\mathbb{R}^d$

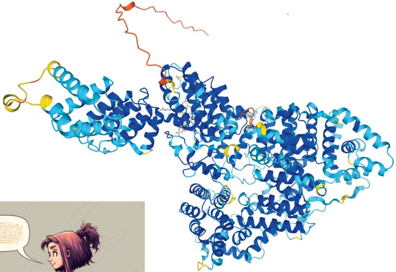
Generative models

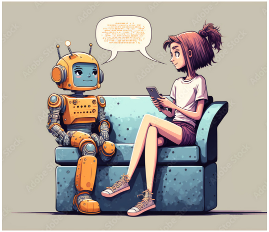


- Given training data $\underbrace{X^{\text{train},i} \sim p_{\text{data}}}_{\text{from a general distribution}}$ $(1 \leq i \leq N)$ in $\mathbb{R}^d$
- Generate new samples $Y \sim p_{\text{data}}$

From generative models to generative AI







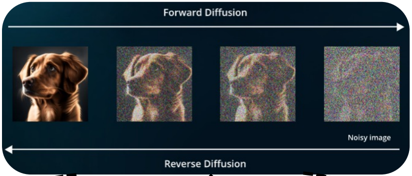
Generative AI is transforming nearly every field of our society.

State-of-the-art diffusion models

Inspired by nonequilibrium thermodynamics

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

Diffusion models

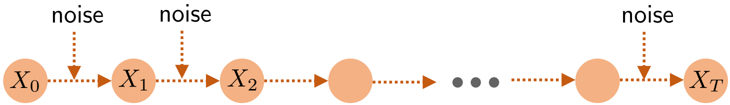


Stable Diffusion

DALLE

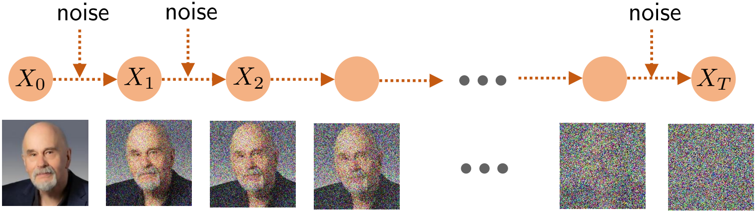
Sora

A high-level description of diffusion models



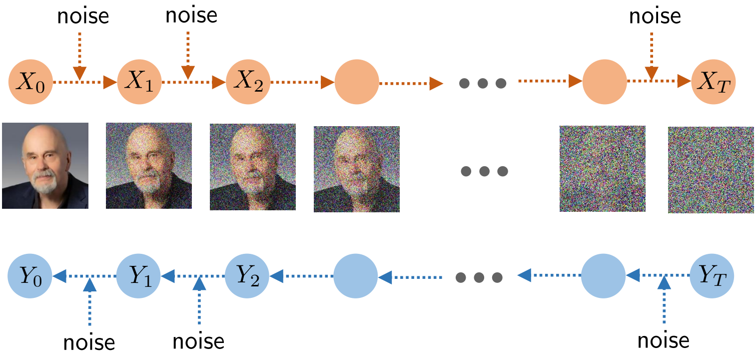
- **forward process:** (progressively) diffuse data into noise

A high-level description of diffusion models



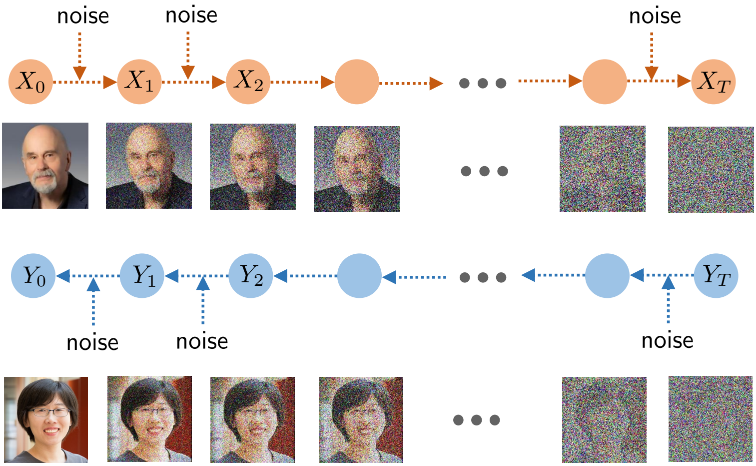
- **forward process:** (progressively) diffuse data into noise

A high-level description of diffusion models



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

A high-level description of diffusion models



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

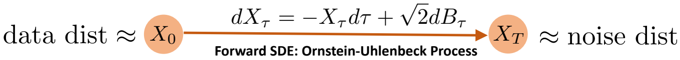
How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t \ (1 \leq t \leq T)$?

How to learn a reverse process s.t. $Y_t \stackrel{\mathrm{d}}{\approx} X_t$ $(1 \le t \le T)$?

It is feasible as long as one knows the score function (Anderson'82; Haussmann and Pardoux'86; Song et al.'20)...

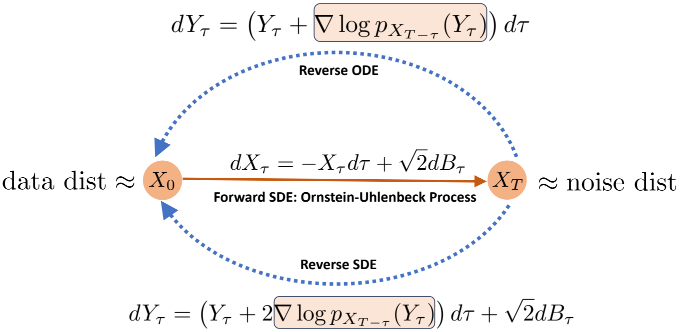
How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$ $(1 \leq t \leq T)$?

It is feasible as long as one knows the score function (Anderson'82; Haussmann and Pardoux'86; Song et al.'20)...



How to learn a reverse process s.t. $Y_t \overset{\mathrm{d}}{\approx} X_t$ $(1 \le t \le T)$?

It is feasible as long as one knows the score function (Anderson'82; Haussmann and Pardoux'86; Song et al.'20)...

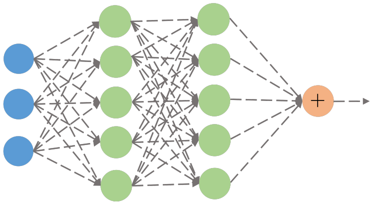


Score is all you need

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

Score is all you need

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

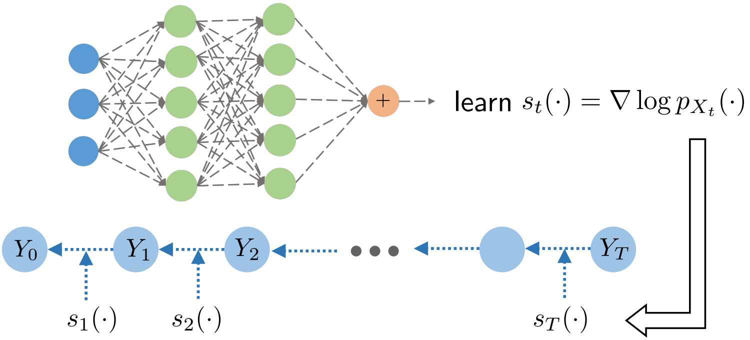


learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

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

Score is all you need

- **score functions** of marginals 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 $\{s_t(\cdot)\}$

Score matching via denoising

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

Score matching via denoising

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

Tweedie's formula (Hyvarinen, 2005; Vincent, 2011):

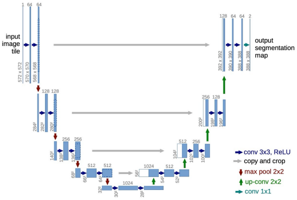
$$s_t^\star(x) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \underbrace{\mathbb{E}_{x_0 \sim p_{\mathsf{data}},\, \epsilon_t \sim \mathcal{N}(0, I_d)} \left[\epsilon_t \,\middle|\, \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t = x\right]}_{\text{MMSE denoising}}.$$

Score matching via denoising

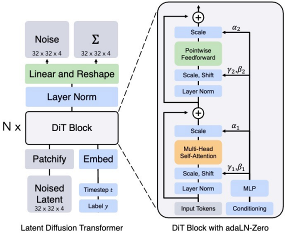
$$X_0 \sim p_{\mathsf{data}}, \quad X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\, \mathcal{N}(0, I_d)$$

Tweedie's formula (Hyvarinen, 2005; Vincent, 2011):

$$s_t^\star(x) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \underbrace{\mathbb{E}_{x_0 \sim p_{\mathsf{data}},\, \epsilon_t \sim \mathcal{N}(0, I_d)} \left[\epsilon_t \,\middle|\, \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon_t = x \right]}_{\text{MMSE denoising}} .$$



U-Net
[Ronneberger, Fischer, Brox, 2015]



Diffusion Transformers
[Peebles and Xie, 2022]

This talk

✄ decouple
score learning $\longleftarrow$ ✗ $\longrightarrow$ $\underbrace{\text{data generation}}_{\textbf{this talk}}$

This talk

✄ decouple

score learning $\longleftarrow$ ✗ $\longrightarrow$ $\underbrace{\text{data generation}}_{\textbf{this talk}}$

Sampling:

When and how fast do diffusion samplers converge?

This talk

✄ decouple

score learning $\longleftarrow$ ✗ $\longrightarrow$ data generation

$\underbrace{\text{data generation}}_{\textbf{this talk}}$

Sampling:

When and how fast do diffusion samplers converge?

Acceleration:

Can we accelerate the convergence of diffusion samplers provably?

This talk

✄ decouple

score learning $\leftarrow$ ✗ $\rightarrow$ $\underbrace{\text{data generation}}_{\textbf{this talk}}$

Sampling:

When and how fast do diffusion samplers converge?

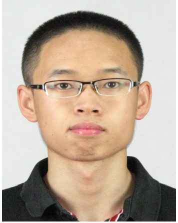
Acceleration:

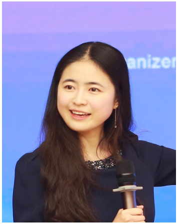
Can we accelerate the convergence of diffusion samplers provably?

Inverse problems:

Can we design provably robust posterior samplers using diffusion priors?

Non-asymptotic convergence for diffusion-based generative models





Gen Li
CUHK

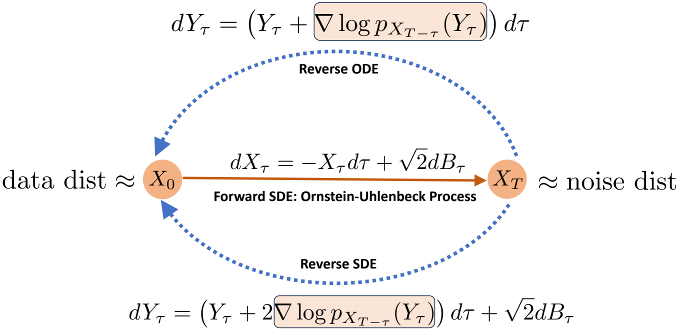
Yuxin Chen
UPenn

Yuting Wei
UPenn

"A Sharp Convergence Theory for The Probability Flow ODEs of Diffusion Models", arXiv:2408.02320.

Two mainstream approaches

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



Two mainstream approaches

— *Ho, Jain, Abbeel '20*

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

1. A stochastic sampler: $\underbrace{\textbf{denoising diffusion probabilistic models}}_{\text{DDPM}}$

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

$$Y_{t-1} = \Psi_t(Y_t, \mathsf{noise}), \quad t = T, \cdots, 1$$

Two mainstream approaches

— *Ho, Jain, Abbeel '20*

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

1. A stochastic sampler: $\underbrace{\textbf{denoising diffusion probabilistic models}}_{\text{DDPM}}$

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

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

Probability flow ODE

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

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

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

Probability flow ODE

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

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

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

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

$$Y_{t-1} = \Phi_t(Y_t), \quad t = T, \cdots, 1$$

Probability flow ODE

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

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

2. A <u>deterministic</u> sampler based on **probability flow ODE**

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

Stochastic versus deterministic samplers

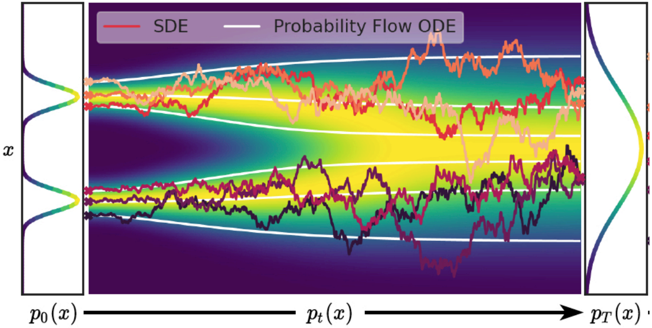
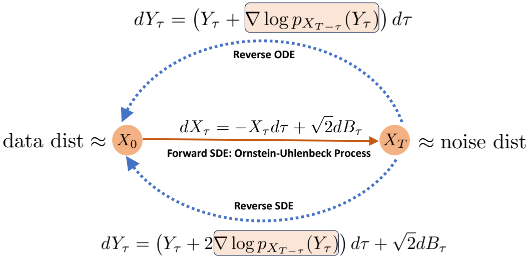


Figure credit: (Song et al '20)

- The stochastic sampler generates more diverse samples, while the deterministic sampler is much faster.

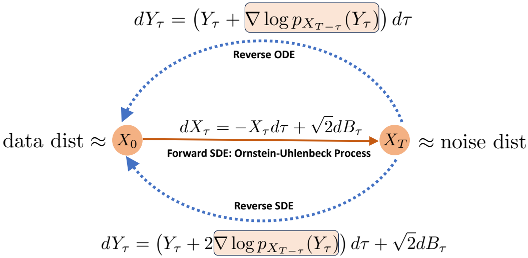
Towards understanding the non-asymptotic convergence

Question: can we understand non-asymptotic convergence of diffusion models in discrete time?



Towards understanding the non-asymptotic convergence

Question: can we understand non-asymptotic convergence of diffusion models in discrete time?



Sources of errors:

- initialization error (dealing with the gap between X_T and Y_T)
- discretization error
- score estimation error

Prior approaches

— Li, Lu, Tan '22
— Chen, Lee, Lu '22
— Chen, Chewi, Li, Li, Salim, Zhang '22
— Chen, Daras, Dimakis '23
— Chen, Chewi, Lee, Li, Lu, Salim '23

discrete-time
diffusion process

continuous-time limits via
SDE/ODE toolbox (e.g., Girsanov thm)

Prior approaches

— *Li, Lu, Tan '22*
— *Chen, Lee, Lu '22*
— *Chen, Chewi, Li, Li, Salim, Zhang '22*
— *Chen, Daras, Dimakis '23*
— *Chen, Chewi, Lee, Li, Lu, Salim '23*

discrete-time
diffusion process

continuous-time limits via
SDE/ODE toolbox (e.g., Girsanov thm)

control discretization error

Prior approaches

— Li, Lu, Tan '22
— Chen, Lee, Lu '22
— Chen, Chewi, Li, Li, Salim, Zhang '22
— Chen, Daras, Dimakis '23
— Chen, Chewi, Lee, Li, Lu, Salim '23

discrete-time diffusion process

continuous-time limits via SDE/ODE toolbox (e.g., Girsanov thm)

control discretization error

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

Prior approaches

— *Li, Lu, Tan '22*
— *Chen, Lee, Lu '22*
— *Chen, Chewi, Li, Li, Salim, Zhang '22*
— *Chen, Daras, Dimakis '23*
— *Chen, Chewi, Lee, Li, Lu, Salim '23*

discrete-time diffusion process

continuous-time limits via SDE/ODE toolbox (e.g., Girsanov thm)

control discretization error

- Built upon toolboxes from SDE/ODE
- Existing analyses were **inadequate for deterministic samplers**

Prior approaches

— *Li, Lu, Tan '22*
— *Chen, Lee, Lu '22*
— *Chen, Chewi, Li, Li, Salim, Zhang '22*
— *Chen, Daras, Dimakis '23*
— *Chen, Chewi, Lee, Li, Lu, Salim '23*

discrete-time
diffusion process

continuous-time limits via
SDE/ODE toolbox (e.g., Girsanov thm)

control discretization error

- Built upon toolboxes from SDE/ODE
- Existing analyses were **inadequate for deterministic samplers**

This talk: non-asymptotic convergence guarantees
for deterministic samplers

Assumptions

- **Minimal data distributional assumptions:**

$$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1$$

for arbitrarily large constant $c_R > 0$

Assumptions

- **Minimal data distributional assumptions:**

$$\mathbb{P}(\|X_0\|_2 \le T^{c_R}) = 1$$

for arbitrarily large constant $c_R > 0$

- ℓ_2 **error of score functions:**

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

Assumptions

- **Minimal data distributional assumptions:**

$$\mathbb{P}(\|X_0\|_2 \leq T^{c_R}) = 1$$

for arbitrarily large constant $c_R > 0$

- ℓ_2 **error of score functions:**

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

- Jacobian **error of score functions:** denote by $J_{s_t^{\star}} = \frac{\partial s_t^{\star}}{\partial x}$ and $J_{s_t} = \frac{\partial s_t}{\partial x}$ the Jacobian matrices of $s_t^{\star}(\cdot)$ and $s_t(\cdot)$, which obey

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

Non-asymptotic complexity of generation

Learning rates: for some large constants $c_0, c_1 > 0$,

$$\beta_1 = \frac{1}{T^{c_0}}$$
$$\beta_t = \frac{c_1 \log T}{T} \min\left\{\beta_1\left(1 + \frac{c_1 \log T}{T}\right)^t, 1\right\}$$

Theorem (Li et al, 2024)

For the deterministic sampler (DDIM-type/prob. flow ODE),

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}} \quad \textit{up to log factor.}$$

Non-asymptotic complexity of generation

Learning rates: for some large constants $c_0, c_1 > 0$,

$$\beta_1 = \frac{1}{T^{c_0}}$$
$$\beta_t = \frac{c_1 \log T}{T} \min\left\{\beta_1\left(1 + \frac{c_1 \log T}{T}\right)^t, 1\right\}$$

Theorem (Li et al, 2024)

For the <u>deterministic</u> sampler (DDIM-type/prob. flow ODE),

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\text{score}} + d\varepsilon_{\text{Jacobi}} \quad \textit{up to log factor.}$$

Our results of **deterministic samplers** provide *sharp* bounds with near optimal dependency with d up to log factors.

Non-asymptotic complexity of generation

Learning rates: for some large constants $c_0, c_1 > 0$,

$$\beta_1 = \frac{1}{T^{c_0}}$$
$$\beta_t = \frac{c_1 \log T}{T} \min\left\{\beta_1 \left(1 + \frac{c_1 \log T}{T}\right)^t, 1\right\}$$

Theorem (Li et al, 2024)

For the <u>deterministic</u> sampler (DDIM-type/prob. flow ODE),

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d}{T} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}} \quad \textit{up to log factor.}$$

Fast convergence for general data distribution,
given good score estimates.

Acceleration?

Acceleration?

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

Acceleration?

- **Training-based methods:** progressive distillation (Salimans et al., 2022), consistency model (Song et al., 2023)...

Acceleration?

- **Training-based methods:** progressive distillation (Salimans et al., 2022), consistency model (Song et al., 2023)...

additional training steps are required 

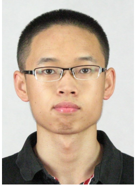
Acceleration?

- **Training-based methods:** progressive distillation (Salimans et al., 2022), consistency model (Song et al., 2023)...

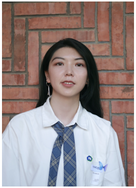
  additional training steps are required 

- **Training-free methods:** DPM-Solver/++ (Lu et al., 2022ab), UniPC (Zhao et al., 2023)...

Can we develop training-free samplers that converge provably faster?



Gen Li
CUHK



Yu Huang
UPenn

Timofey Efimov
CMU

Yuting Wei
UPenn

Yuxin Chen
UPenn

"Accelerating Convergence of Score-Based Diffusion Models, Provably", ICML 2024.

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^{t} \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^{t} \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}}\Big(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\Big) \quad \text{probability flow ODE}$$

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^{t} \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}}\Big(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\Big) \quad \text{probability flow ODE}$$

Refined approximation?

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^{t} \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \Big(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\Big) \quad \text{probability flow ODE}$$

Refined approximation?

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

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^{t} \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \Big(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\Big)$$ probability flow ODE

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\right) + \frac{1}{\sqrt{\alpha_t}} \left(\frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})} \Big(s_t\big(X_t\big) - \sqrt{\alpha_{t+1}}\, \underbrace{s_{t+1}\big(X_{t+1}\big)}_{\text{reuse}}\Big)\right)$$ **Ours**

Acceleration via high-order ODE discretization

Solving the probability flow ODE ($\overline{\alpha}_t := \prod_{k=1}^t \alpha_k$ with $\alpha_t = 1 - \beta_t$):

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

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \Big(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\Big)$$ probability flow ODE

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

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}} \left(X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2} s_t(X_t)\right) + \frac{1}{\sqrt{\alpha_t}} \left(\frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})} \Big(s_t\big(X_t\big) - \sqrt{\alpha_{t+1}}\, \underbrace{s_{t+1}\big(X_{t+1}\big)}_{\text{reuse}}\Big)\right)$$ **Ours**

DPM-Solver-2 (Lu et al, 2022a): to construct second-order ODE solver

Accelerated deterministic sampler

Theorem (Li et al. 2024, informal)

The accelerated deterministic sampler obeys

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^6}{T^2} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}}$$

- Improved rate $\widetilde{O}(1/T^2)$ compared to probability flow ODE $\widetilde{O}(1/T)$

Accelerated deterministic sampler

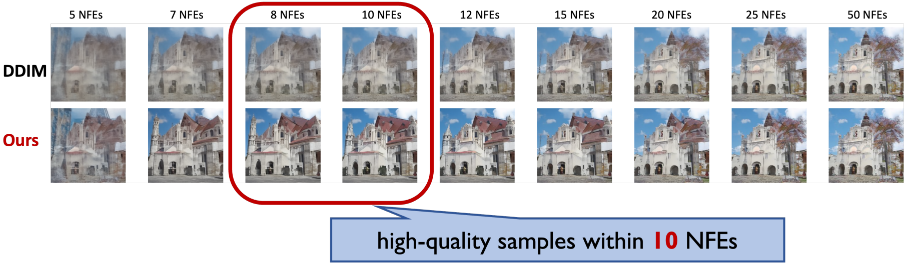
Theorem (Li et al. 2024, informal)

The accelerated deterministic sampler obeys

$$\mathsf{TV}\left(p_{X_1}, p_{Y_1}\right) \lesssim \frac{d^6}{T^2} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}}$$

- Improved rate $\widetilde{O}(1/T^2)$ compared to probability flow ODE $\widetilde{O}(1/T)$

Numbers of function evaluation (NFE) 4 ⟶ 50



Accelerated deterministic sampler

Theorem (Li et al. 2024, informal)

The accelerated deterministic sampler obeys

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^6}{T^2} + \sqrt{d}\varepsilon_{\mathsf{score}} + d\varepsilon_{\mathsf{Jacobi}}$$

- Improved rate $\widetilde{O}(1/T^2)$ compared to probability flow ODE $\widetilde{O}(1/T)$

Sampled images with 5 NFEs: crisper and less noisy

Provably robust diffusion posterior sampling for inverse problems

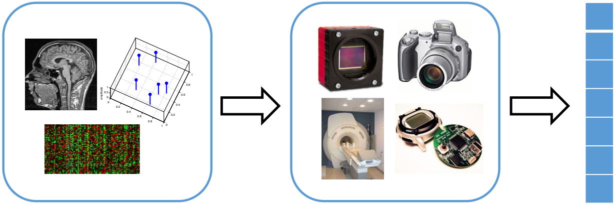
Xingyu Xu
CMU

"Provably Robust Score-Based Diffusion Posterior Sampling for Plug-and-Play Image Reconstruction", NeurIPS 2024, arXiv:2403.17042.

Inverse problems

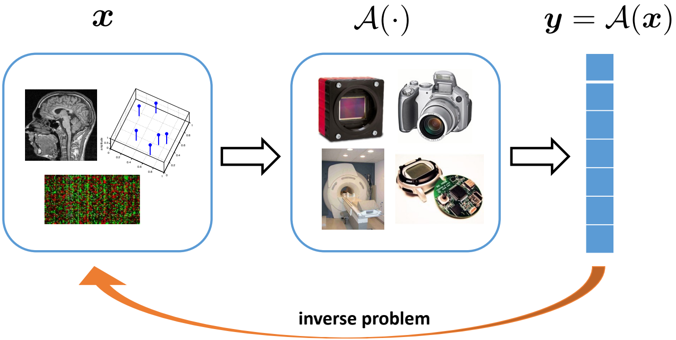
Forward model: we interrogate the signal of interest $\boldsymbol{x}$ through forward model $\mathcal{A}$ and make measurements $\boldsymbol{y}$.

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



Inverse problems

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

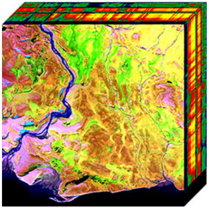


Inverse problem: recover the signal of interest x from y.

Ubiquitous, but often ill-posed

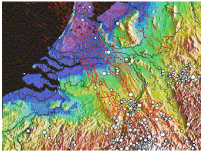
healthcare

Radio astronomy

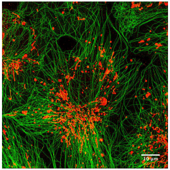


hyperspectral

Internet traffic



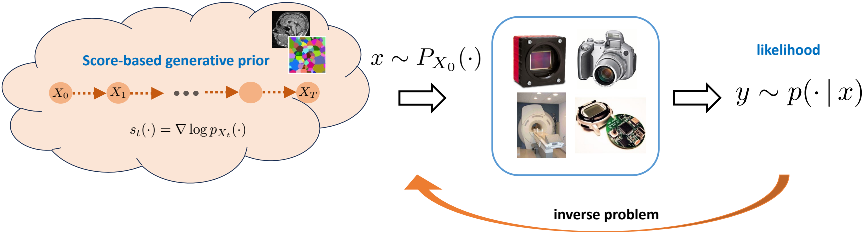
seismic imaging



microscopy

Can we exploit flexible / expressive data priors prescribed by diffusion models for ill-posed inverse problems?

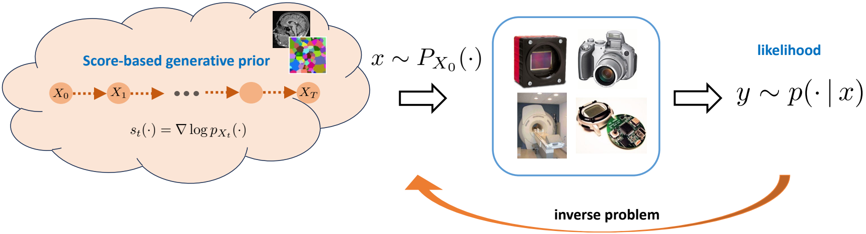
Score-based diffusion model for inverse problems



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 diffusion model for inverse problems



Posterior sampling: sample from

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

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

A highly incomplete list of prior work

- (Song et al., 2021)
- (Laumont et al., 2022)
- (Kawar et al., 2022)
- (Trippe et al., 2022)
- (Graikos et al., 2022)
- (Chung et al., 2023)
- (Cardoso et al., 2023)
- (Song et al., 2023)
- (Mardani et al., 2023)
- (Feng et al., 2023)
- (Chen et al., 2023)
- (Coeurdoux et al., 2023)
- (Wu et al., 2022)
- (Dou and Song, 2024)
- ...

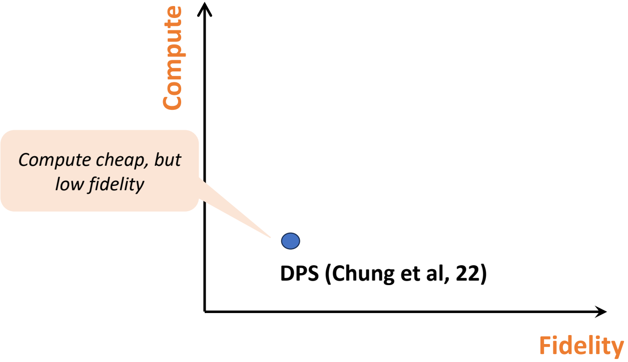
A highly incomplete list of prior work

- (Song et al., 2021)
- (Laumont et al., 2022)
- (Kawar et al., 2022)
- (Trippe et al., 2022)
- (Graikos et al., 2022)
- (Chung et al., 2023)
- (Cardoso et al., 2023)
- (Song et al., 2023)
- (Mardani et al., 2023)
- (Feng et al., 2023)
- (Chen et al., 2023)
- (Coeurdoux et al., 2023)
- (Wu et al., 2022)
- (Dou and Song, 2024)
- ...

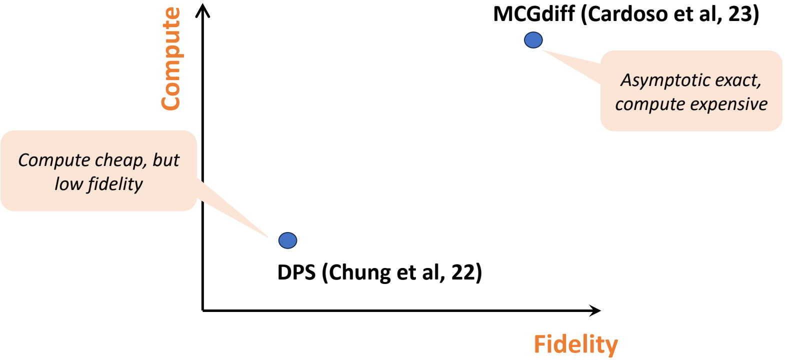
Majority of the existing algorithms are heuristic and/or tailored to linear inverse problems.

Towards provably efficient and accurate inversion

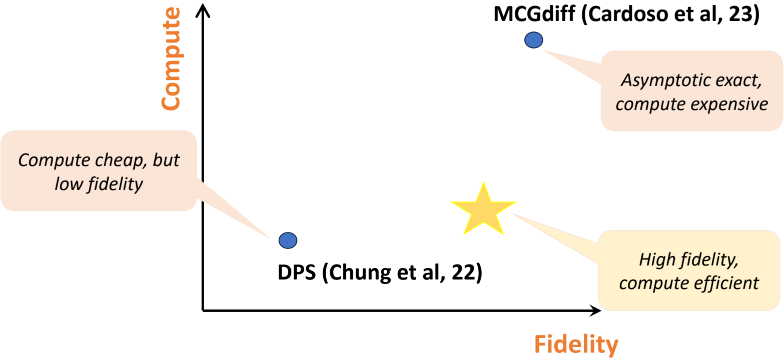
Towards provably efficient and accurate inversion



Towards provably efficient and accurate inversion



Towards provably efficient and accurate inversion



Goal: develop provably compute-efficient and high-fidelity diffusion-based inversion methods for arbitrary forward model.

Our approach: diffusion plug-and-play (DPnP)

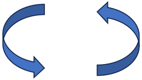
Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

$$p(\cdot|y) \propto \exp\Big(\log p(\cdot) + \mathcal{L}(\cdot\,;\,y)\Big)$$

Given an annealing schedule $\{\eta_k\}$,

Proximal consistency sampler:

$$\widehat{x}_{k+\frac{1}{2}} \propto \exp\left(\mathcal{L}(\cdot\,;\,y) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_k\|^2\right)$$



Diffusion denoising sampler:

$$\widehat{x}_{k+1} \propto \exp\left(\log p(\cdot) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_{k+\frac{1}{2}}\|^2\right)$$

Our approach: diffusion plug-and-play (DPnP)

Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

$$p(\cdot|y) \propto \exp\Big(\log p(\cdot) + \mathcal{L}(\cdot\,;\,y)\Big)$$

Given an annealing schedule $\{\eta_k\}$,

Proximal consistency sampler:

$$\widehat{x}_{k+\frac{1}{2}} \propto \exp\left(\mathcal{L}(\cdot\,;\,y) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_k\|^2\right)$$

✓ Readily implementable by, e.g., MALA

Diffusion denoising sampler:

$$\widehat{x}_{k+1} \propto \exp\left(\log p(\cdot) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_{k+\frac{1}{2}}\|^2\right)$$

Our approach: diffusion plug-and-play (DPnP)

Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

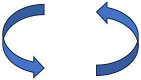
$$p(\cdot|y) \propto \exp\Big(\log p(\cdot) + \mathcal{L}(\cdot\,;\,y)\Big)$$

Given an annealing schedule $\{\eta_k\}$,

Proximal consistency sampler:

$$\widehat{x}_{k+\frac{1}{2}} \propto \exp\left(\mathcal{L}(\cdot\,;\,y) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_k\|^2\right)$$

✓ Readily implementable by, e.g., MALA



Diffusion denoising sampler:

$$\widehat{x}_{k+1} \propto \exp\left(\log p(\cdot) - \frac{1}{2\eta_k^2}\|\cdot - \widehat{x}_{k+\frac{1}{2}}\|^2\right)$$

How do we implement this step using diffusion score functions?

Diffusion denoising sampler

Posterior sampling for AWGN denoising:

$$\exp\left(\log p(x) - \frac{1}{2\eta_k^2}\|x - \widehat{x}_{k+\frac{1}{2}}\|^2\right) \propto p(x^\star \mid x^\star + \eta_k w = \widehat{x}_{k+\frac{1}{2}})$$

where $w \sim \mathcal{N}(0, I_d)$.

- Key insight: this can be solved by diffusion!

Diffusion denoising sampler

Posterior sampling for AWGN denoising:

$$\exp\left(\log p(x) - \frac{1}{2\eta_k^2}\|x - \widehat{x}_{k+\frac{1}{2}}\|^2\right) \propto p(x^\star \mid x^\star + \eta_k w = \widehat{x}_{k+\frac{1}{2}})$$

where $w \sim \mathcal{N}(0, I_d)$.

- Key insight: this can be solved by diffusion!
  - stochastic/deterministic samplers via reversing properly defined forward processes (e.g., Ornstein-Uhlenbeck process), whose score functions can be mapped from $s_t(\cdot)$.

Diffusion denoising sampler

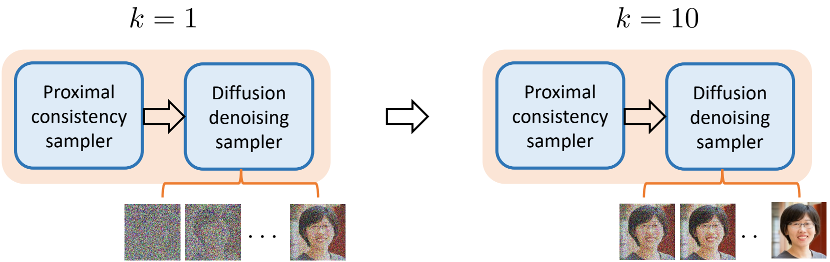
Posterior sampling for AWGN denoising:

$$\exp\left(\log p(x) - \frac{1}{2\eta_k^2}\|x - \widehat{x}_{k+\frac{1}{2}}\|^2\right) \propto p(x^\star \,|\, x^\star + \eta_k w = \widehat{x}_{k+\frac{1}{2}})$$

where $w \sim \mathcal{N}(0, I_d)$.

- Key insight: this can be solved by diffusion!
  - stochastic/deterministic samplers via reversing properly defined forward processes (e.g., Ornstein-Uhlenbeck process), whose score functions can be mapped from $s_t(\cdot)$.
- The resulting update rules are similar to, but not the same as, the ones used for generation.

Schematic view of DPnP



- Each iteration of DPnP contains a "full" reverse denoising process with multiple denoising steps.
- But, it can be easily combined with acceleration schemes, such as distillation, to speed up.

Our theory

Theorem (Xu and Chi, 2024)

Set constant $\eta_k = \eta > 0$. Define a stationary distribution π_η by

$$\pi_\eta(x) \propto p(x) q_\eta(x), \qquad q_\eta(x) = \mathrm{e}^{\mathcal{L}(\cdot\,;\,y)} * p_{\eta\zeta}(x),$$

where $\zeta \sim \mathcal{N}(0, I_d)$ and $*$ denotes convolution. There exists $\lambda := \lambda(p, \mathcal{L}, \eta) \in (0,1)$, such that for any accuracy level $\epsilon > 0$, with $K \asymp \frac{1}{1-\lambda}\log(1/\epsilon)$, we have

$$\mathsf{TV}(p_{\widehat{x}_K}, \pi_\eta) \lesssim \underbrace{\epsilon\sqrt{\chi^2(p_{\widehat{x}_1} \,\|\, \pi_\eta)}}_{\text{init error}} + \underbrace{\frac{1}{1-\lambda}(\epsilon_{\mathsf{DDS}} + \epsilon_{\mathsf{PCS}}) \log\left(\frac{1}{\epsilon}\right)}_{\text{sampler error}},$$

where ϵ_{PCS} and ϵ_{DDS} are the total variation error of PCS *and* DDS*.*

- A diminishing schedule $\{\eta_k\}$ ensures asymptotic consistency.

Our theory

Theorem (Xu and Chi, 2024)

Set constant $\eta_k = \eta > 0$. Define a stationary distribution π_η by

$$\pi_\eta(x) \propto p(x) q_\eta(x), \qquad q_\eta(x) = \mathrm{e}^{\mathcal{L}(\cdot\,;\,y)} * p_{\eta\zeta}(x),$$

where $\zeta \sim \mathcal{N}(0, I_d)$ and $*$ denotes convolution. There exists $\lambda := \lambda(p, \mathcal{L}, \eta) \in (0,1)$, such that for any accuracy level $\epsilon > 0$, with $K \asymp \frac{1}{1-\lambda} \log(1/\epsilon)$, we have

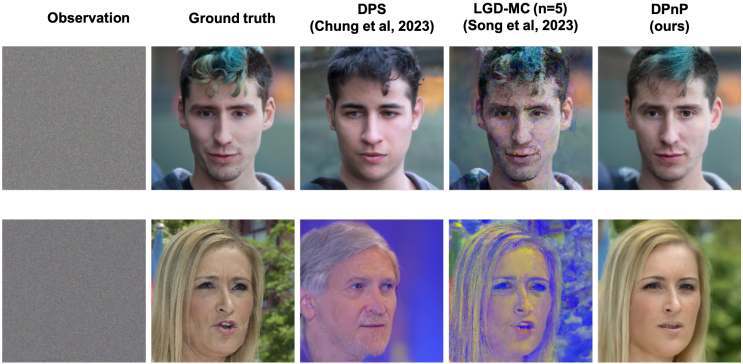
$$\mathsf{TV}(p_{\widehat{x}_K}, \pi_\eta) \lesssim \underbrace{\epsilon \sqrt{\chi^2(p_{\widehat{x}_1} \,\|\, \pi_\eta)}}_{\text{init error}} + \underbrace{\frac{1}{1-\lambda} (\epsilon_{\mathsf{DDS}} + \epsilon_{\mathsf{PCS}}) \log\left(\frac{1}{\epsilon}\right)}_{\text{sampler error}},$$

where ϵ_{PCS} and ϵ_{DDS} are the total variation error of PCS *and* DDS.

- A diminishing schedule $\{\eta_k\}$ ensures asymptotic consistency.

DPnP is the first provably-robust posterior sampling method for nonlinear inverse problems using unconditional diffusion priors.

Phase retrieval: recover an unknown image from the magnitude of its masked Fourier transform.



DPnP recovers the fine-grained details more faithfully.

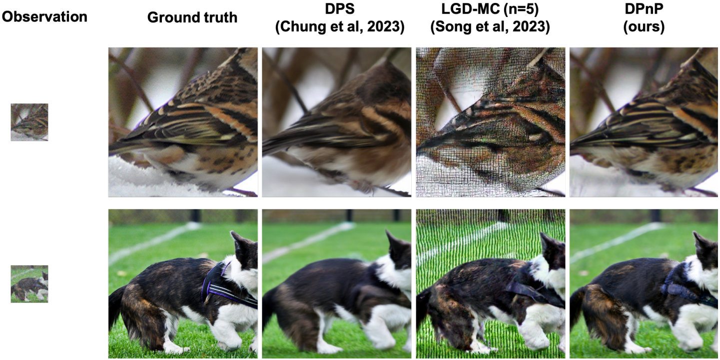
Numerical experiments

Quantized sensing: recover an unknown image from its one-bit dithered measurements.

DPnP recovers the fine-grained details more faithfully.

Numerical experiments

Super resolution: recover an unknown image from its 4x downsampled version.



DPnP recovers the fine-grained details more faithfully.

More metrics

Table: Performance on the ImageNet 256×256 validation dataset.

Algorithm	Super-resolution (4x, linear)		Phase retrieval (nonlinear)		Quantized sensing (nonlinear)		Time per sample
	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	
DPnP-DDIM (ours)	**0.416**	**21.6**	**0.562**	**13.4**	**0.363**	**23.0**	$\sim$ 240s
DPS	0.473	20.2	0.677	**13.4**	0.542	18.7	$\sim$ 150s
LGD-MC ($n = 5$)	**0.416**	20.9	0.592	12.8	0.384	22.3	$\sim$ 150s

Table: Performance on the FFHQ 256×256 validation dataset.

Algorithm	Super-resolution (4x, linear)		Phase retrieval (nonlinear)		Quantized sensing (nonlinear)		Time per sample
	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	
DPnP-DDIM (ours)	**0.301**	**24.2**	**0.376**	**22.4**	**0.293**	**24.2**	$\sim$ 90s
DPS	0.331	23.1	0.490	17.4	0.367	21.7	$\sim$ 60s
LGD-MC ($n = 5$)	0.318	23.9	0.522	16.4	0.317	23.9	$\sim$ 60s

More metrics

Table: Performance on the ImageNet 256×256 validation dataset.

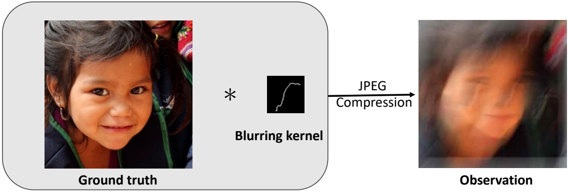
Algorithm	Super-resolution (4x, linear)		Phase retrieval (nonlinear)		Quantized sensing (nonlinear)		Time per sample
	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	
DPnP-DDIM (ours)	**0.416**	**21.6**	**0.562**	**13.4**	**0.363**	**23.0**	~ 240s
DPS	0.473	20.2	0.677	**13.4**	0.542	18.7	~ 150s
LGD-MC ($n = 5$)	**0.416**	20.9	0.592	12.8	0.384	22.3	~ 150s

Table: Performance on the FFHQ 256×256 validation dataset.

Algorithm	Super-resolution (4x, linear)		Phase retrieval (nonlinear)		Quantized sensing (nonlinear)		Time per sample
	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	
DPnP-DDIM (ours)	**0.301**	**24.2**	**0.376**	**22.4**	**0.293**	**24.2**	~ 90s
DPS	0.331	23.1	0.490	17.4	0.367	21.7	~ 60s
LGD-MC ($n = 5$)	0.318	23.9	0.522	16.4	0.317	23.9	~ 60s

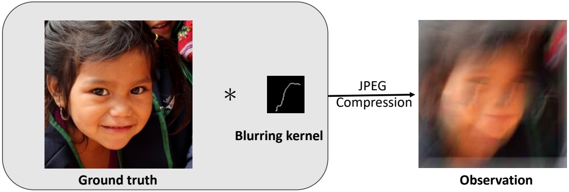
DPnP achieves better performance with a bit more compute.

Blind delurring with JPEG compression (w/ T. Efimov):

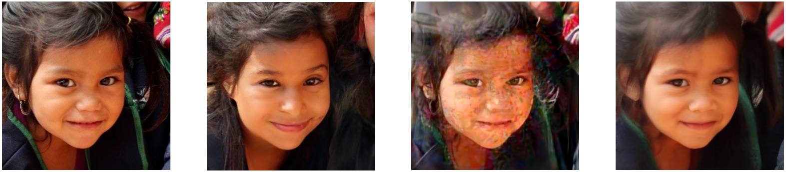


Extension to blind nonlinear inverse problems

Blind delurring with JPEG compression (w/ T. Efimov):

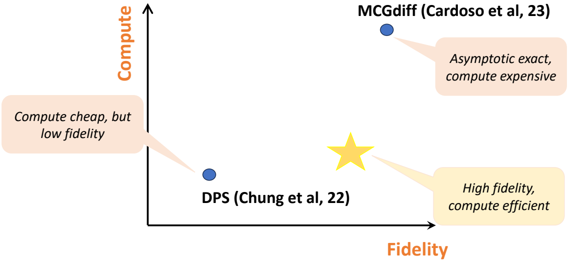


Ongoing work:



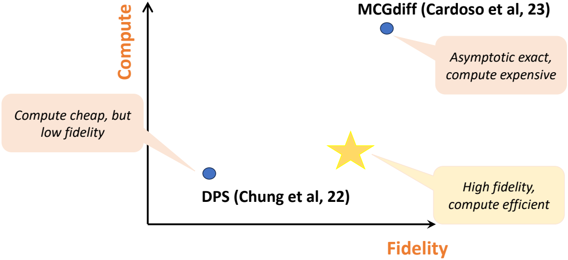
Ground truth | BlindDPS | GibbsDDRM | BlindDPnP (ours)

Summary: diffusion models



Diffusion models are showing great promise in generative AI for Science.

Summary: diffusion models



Diffusion models are showing great promise in generative AI for Science.

Future directions:

- Algorithm and theory for diffusion-based inverse problems: provable guarantees, compute/fidelity trade-offs.
- Applications in imaging science and beyond: 3D/4D imaging, sequence reconstruction, scalability.

Thanks!

- Towards Non-Asymptotic Convergence for Diffusion-Based Generative Models, ICLR 2024.
- Accelerating Convergence of Score-Based Diffusion Models, Provably, ICML 2024.
- A Sharp Convergence Theory for The Probability Flow ODEs of Diffusion Models, arXiv:2408.02320.
- Provably Robust Score-Based Diffusion Posterior Sampling for Plug-and-Play Image Reconstruction, arXiv:2403.17042.

Thanks!

The χ Group

https://users.ece.cmu.edu/~yuejiec/