Solving Inverse Problems with Generative Priors: From Low-rank to Diffusion Models

Yuejie Chi

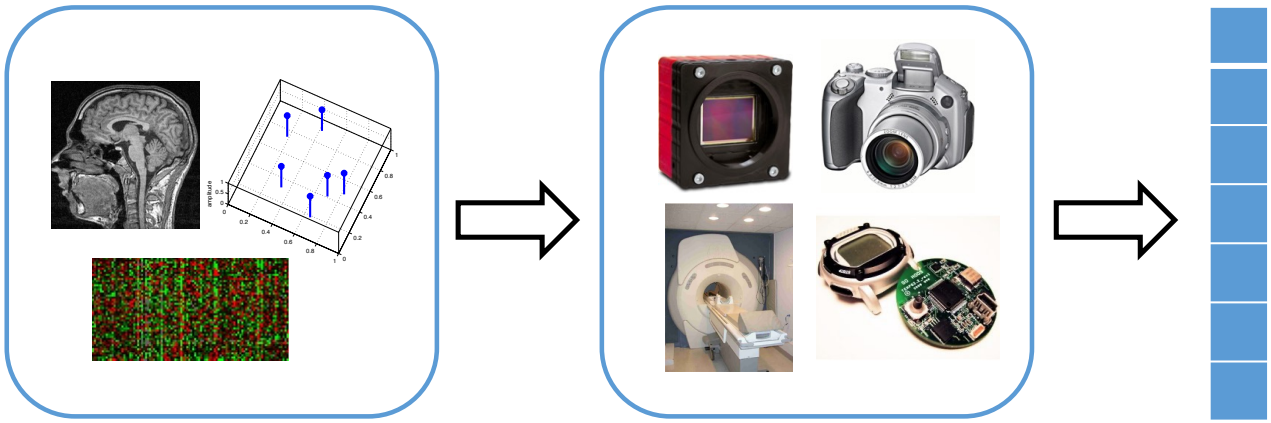
Carnegie Mellon University

NIST/IEEE Conference on Computational Imaging Using Synthetic Apertures
May 2024

Inverse problems

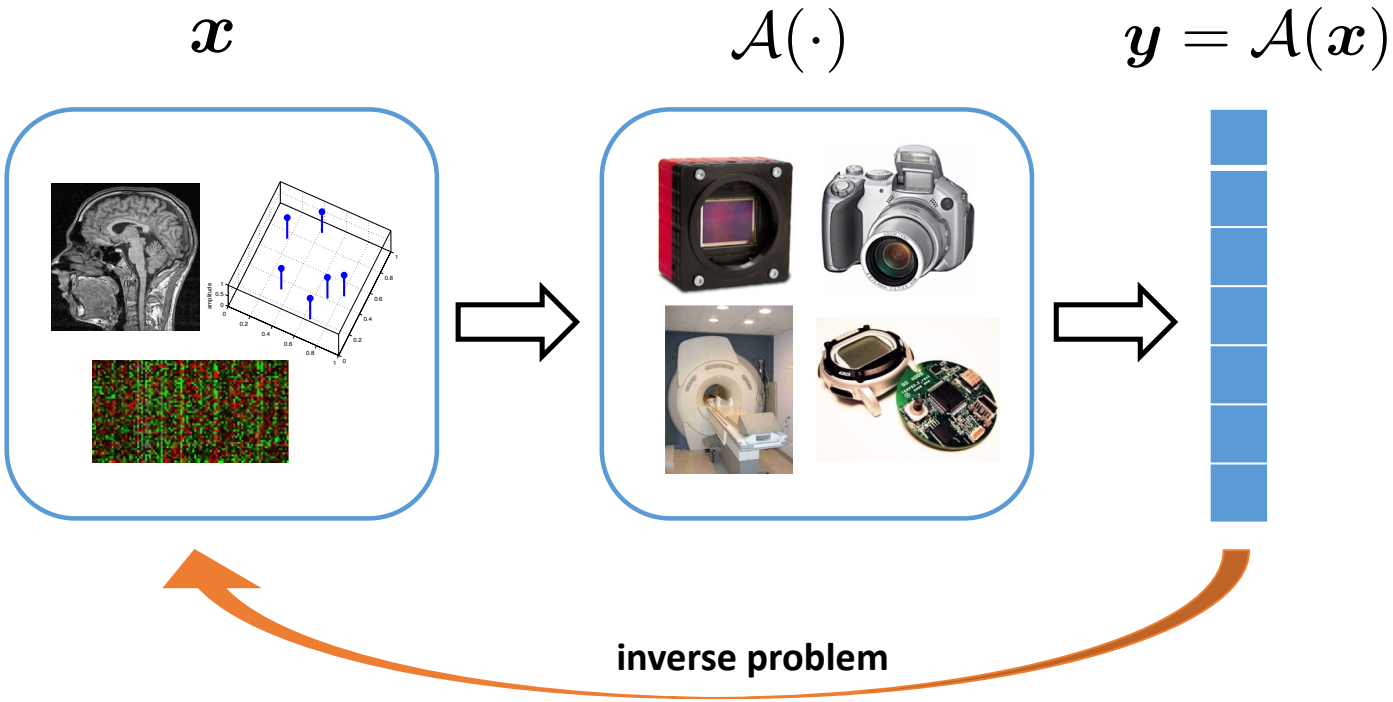
Forward model: we interrogate the signal of interest $\boldsymbol{x}$ through forward model $\mathcal{A}$ and make measurements $\boldsymbol{y}$.

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



Inverse problems

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



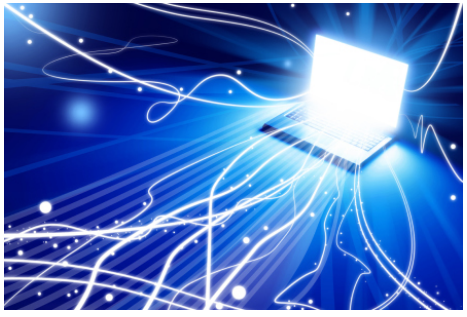
Inverse problem: recover the signal of interest x from y.

Ubiquitous in sensing and imaging applications

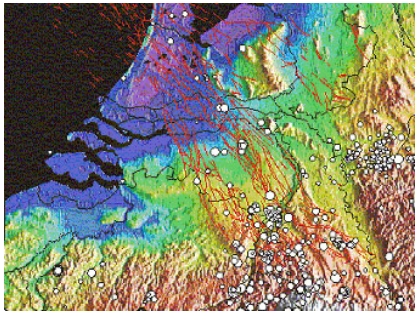
healthcare

Radio astronomy

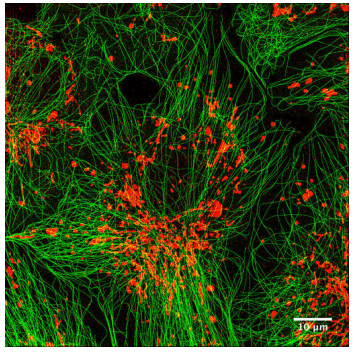
hyperspectral



Internet traffic



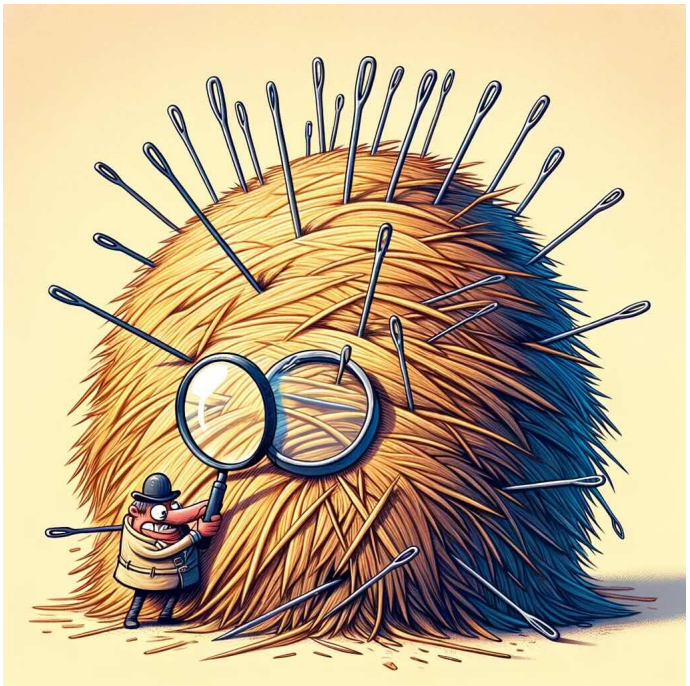
seismic imaging



microscopy

Challenges: finding needles in a haystack

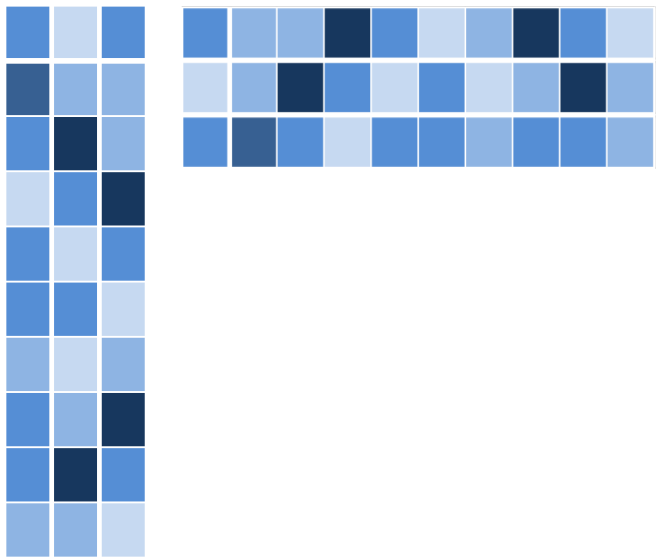
- **Sampling constraints:** sample-starved, low signal-to-noise ratio, nonlinear measurements;
- **Ill-conditioned sources:** weak and fine-grained information;
- **Resiliency:** miscalibration, missing data, corruptions, etc.



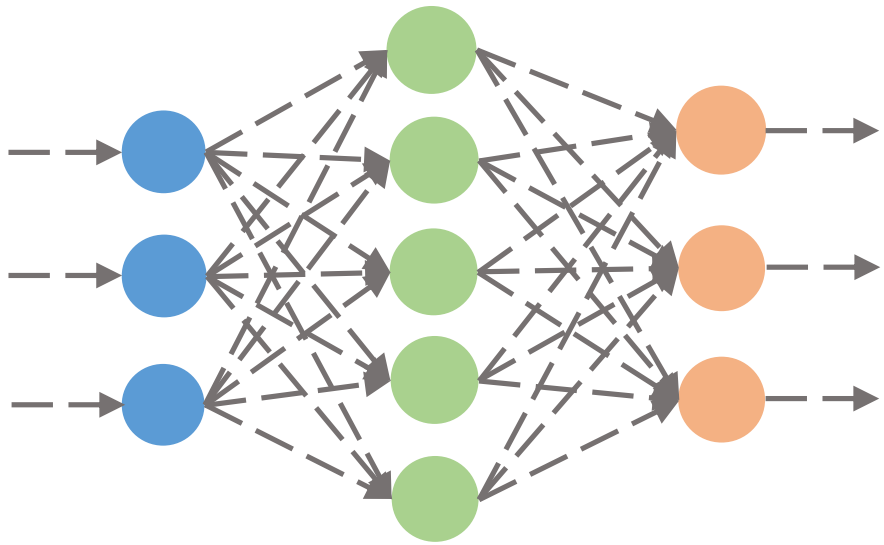
DALLE generated with the prompt "finding needles in the haystack"

Geometry as a prior: from low-rank to generative models

How do we learn effectively leveraging the data priors?



Subspace models:
Sparsity, low-rank, …



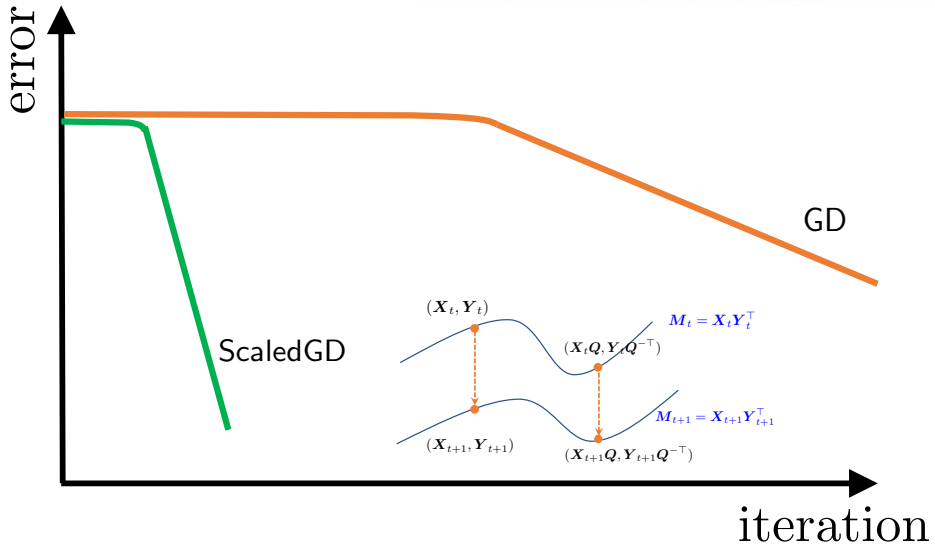
Neural networks:
GAN, VAE, diffusion models…

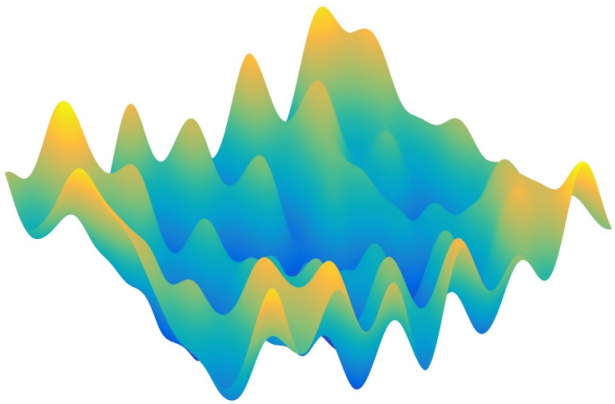
First vignette: preconditioning for low-rank learning

An optimization vignette: preconditioning to accelerate nonconvex ill-conditioned low-rank estimation

preconditioning

overparameterization

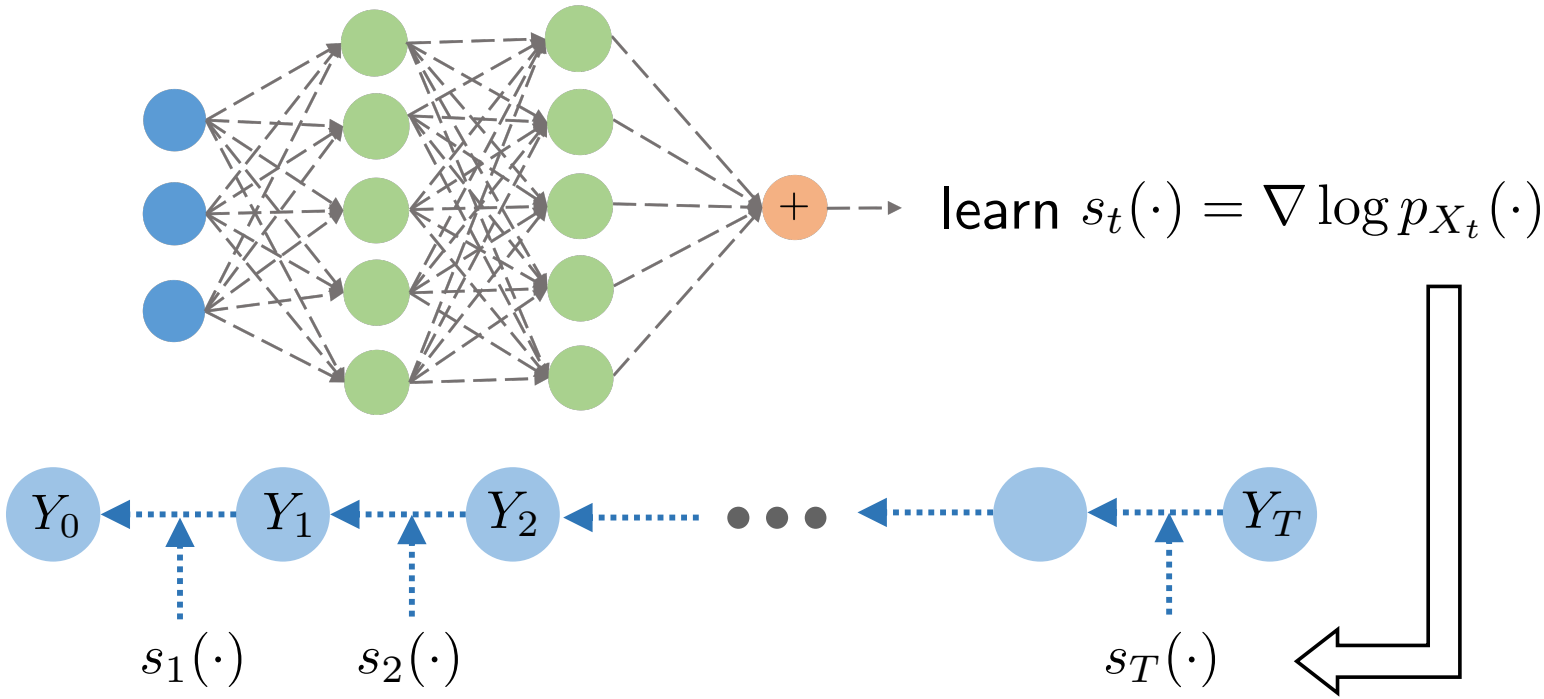




(JMLR 2020, TSP 2021, JMLR 2022, I&I 2023, ICML 2023).

Second vignette: diffusion models for inverse problems

A sampling vignette: how can we leverage score-based generative models for solving inverse problems, efficiently and provably?

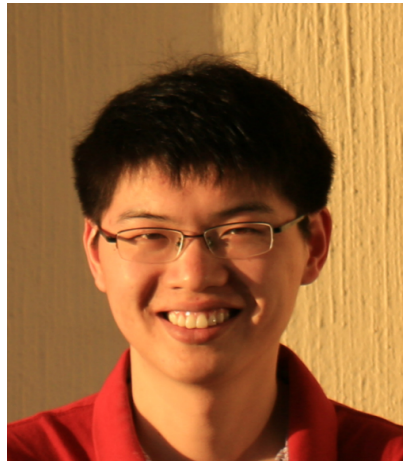


(Xu and Chi, arXiv:2403.17042)

Accelerating gradient descent for ill-conditioned low-rank estimation

Tian Tong

CMU→Amazon

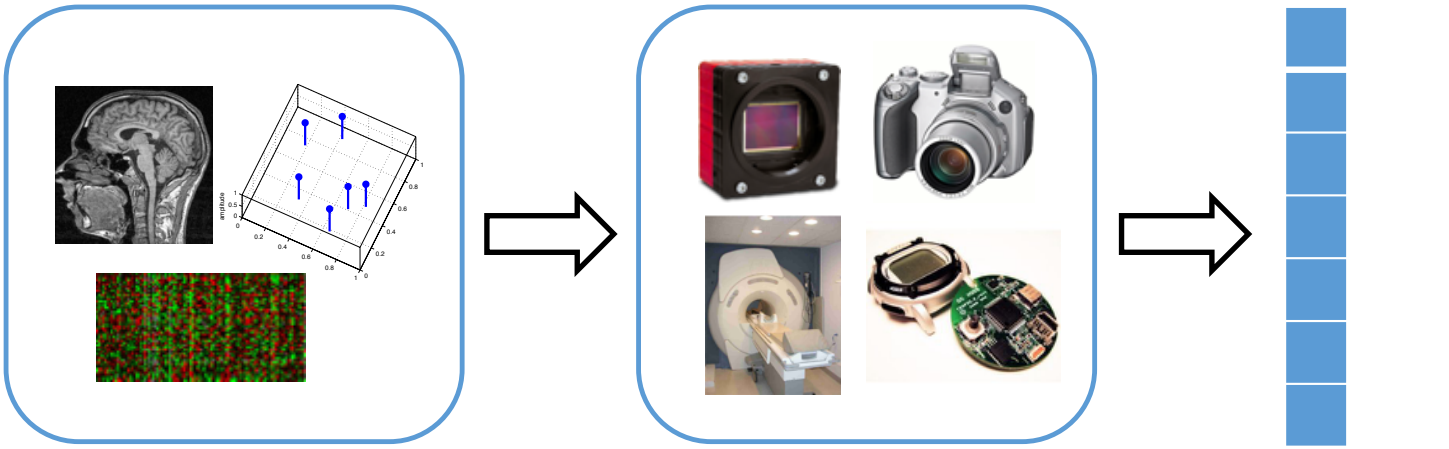


Cong Ma

UChicago

A canonical problem: low-rank matrix sensing

$$\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2} \quad \mathrm{rank}(\boldsymbol{M}) = r \qquad \mathcal{A}(\cdot) \text{ linear map} \qquad \boldsymbol{y} \in \mathbb{R}^m$$



$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{M}) + \text{noise}$$

Recover M in the sample-starved regime:

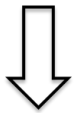
$$\underbrace{(n_1 + n_2)r}_{\text{degree of freedom}} \lesssim \underbrace{m}_{\text{sensing budget}} \ll \underbrace{n_1 n_2}_{\text{ambient dimension}}$$

Low-rank matrix factorization

$$\min_{\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}} \quad \mathsf{rank}(\boldsymbol{Z}) \qquad \text{s.t.} \quad \boldsymbol{y} \approx \mathcal{A}(\boldsymbol{Z})$$

Low-rank matrix factorization

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}} \ \mathsf{rank}(\boldsymbol{Z}) \qquad \text{s.t.} \quad \boldsymbol{y}\approx\mathcal{A}(\boldsymbol{Z})$$



$$\min_{\mathrm{rank}(\boldsymbol{Z})=r} \ \frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

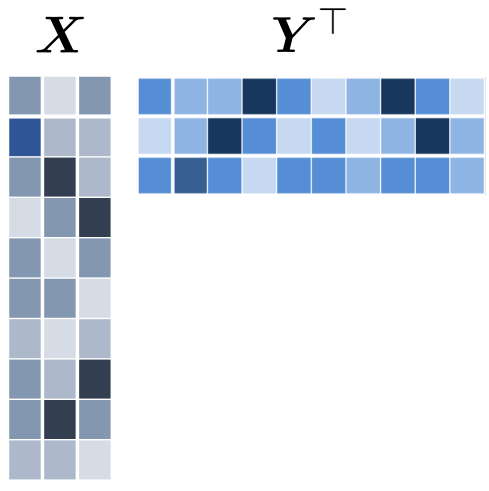
Low-rank matrix factorization

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}} \quad \text{rank}(\boldsymbol{Z}) \qquad \text{s.t.} \quad \boldsymbol{y}\approx\mathcal{A}(\boldsymbol{Z})$$

$$\Downarrow$$

$$\min_{\text{rank}(\boldsymbol{Z})=r} \quad \frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

$$\Downarrow \qquad \boldsymbol{Z} =$$



$$\min_{\boldsymbol{X}\in\mathbb{R}^{n_1\times r},\boldsymbol{Y}\in\mathbb{R}^{n_2\times r}} \quad f(\boldsymbol{X},\boldsymbol{Y})=\frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

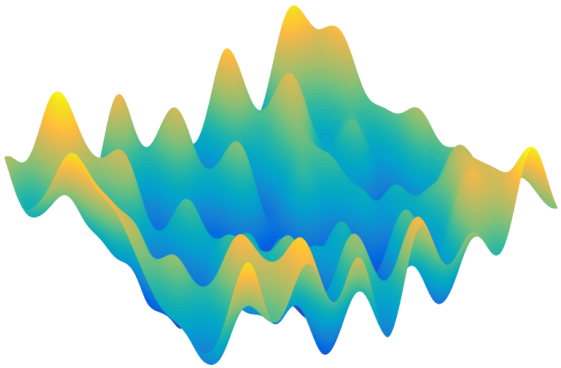
Low-rank matrix factorization

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}} \ \operatorname{rank}(\boldsymbol{Z}) \qquad \text{s.t.} \quad \boldsymbol{y}\approx\mathcal{A}(\boldsymbol{Z})$$

$$\Downarrow$$

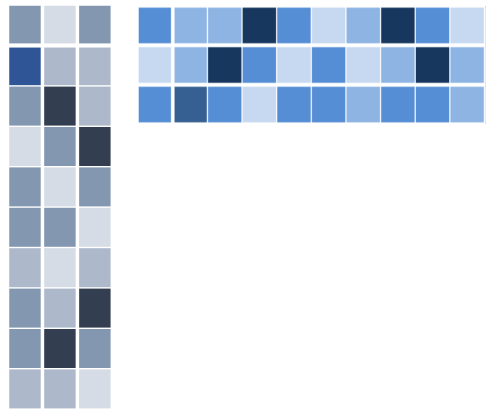
$$\min_{\operatorname{rank}(\boldsymbol{Z})=r} \ \frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

scalable, but nonconvex!



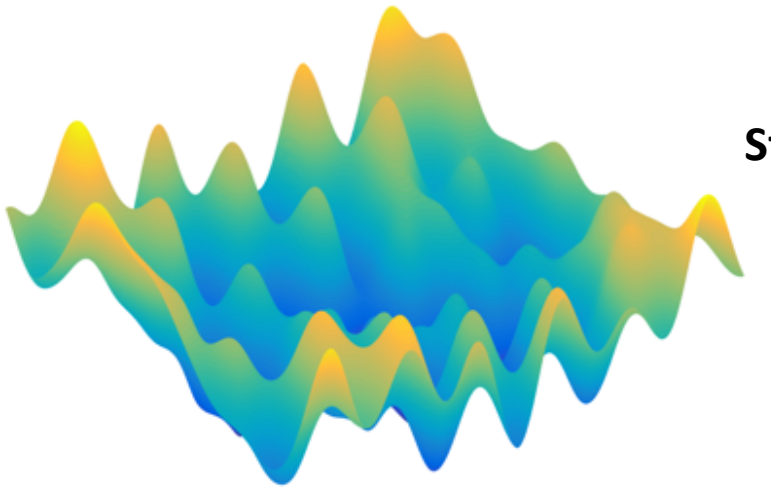
$$\Downarrow$$

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}^\top$$



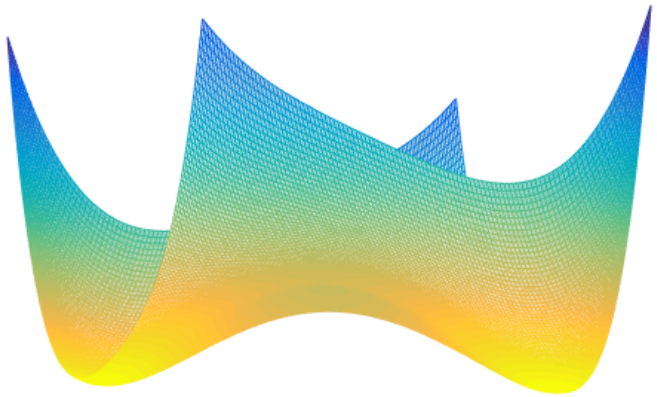
$$\min_{\boldsymbol{X}\in\mathbb{R}^{n_1\times r},\boldsymbol{Y}\in\mathbb{R}^{n_2\times r}} \ f(\boldsymbol{X},\boldsymbol{Y})=\frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

Statistics meets optimization



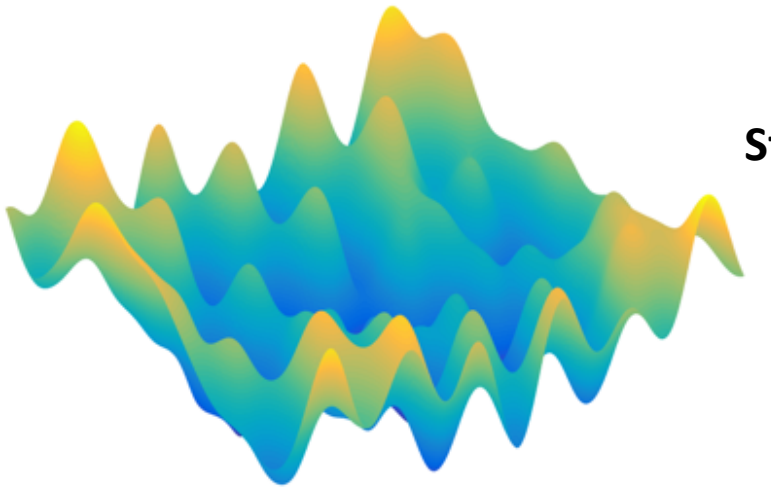
worst case

Statistical model



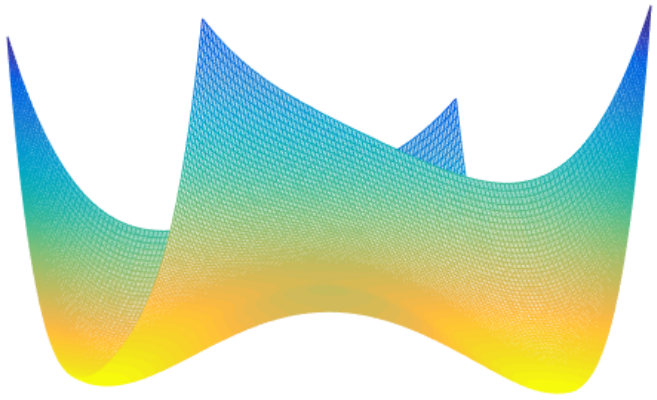
average case

Statistics meets optimization



worst case

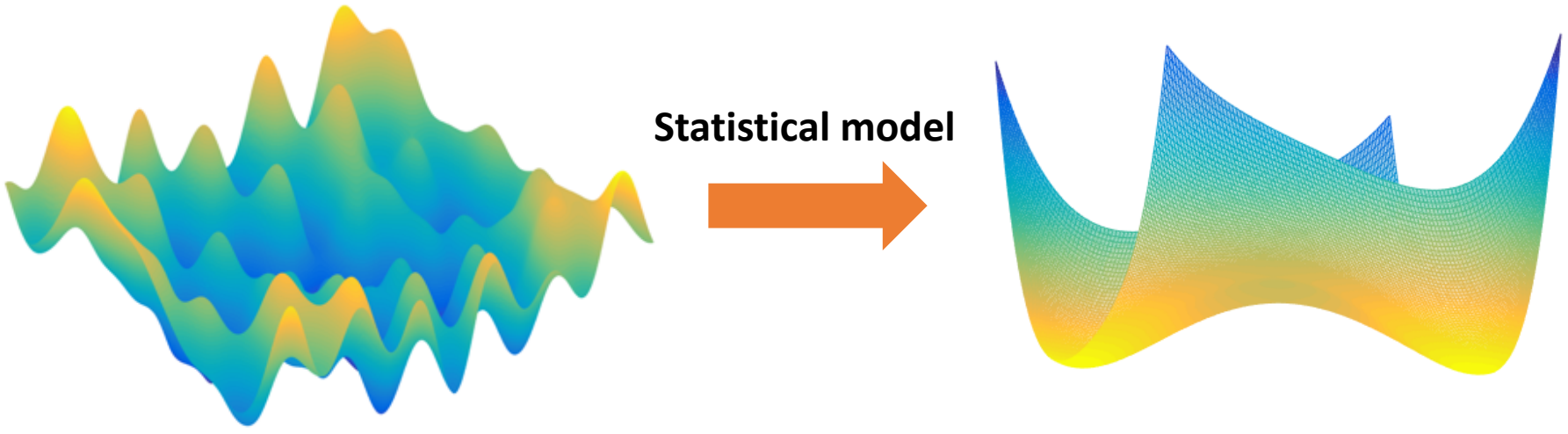
Statistical model



average case

Simple algorithms can be efficient for nonconvex problems!

Statistics meets optimization



worst case

average case

Simple algorithms can be efficient for nonconvex problems!

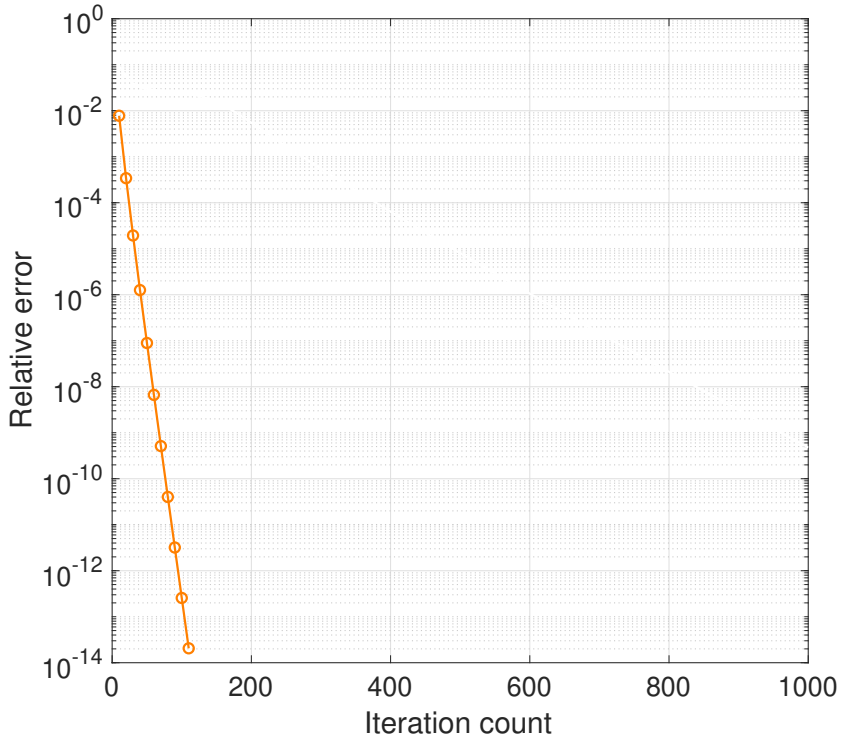
Vanilla gradient descent (GD):

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta \nabla_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$

for $t = 0, 1, \ldots$ from a carefully chosen (e.g., spectral) initialization.

Benign nonconvexity: global linear convergence

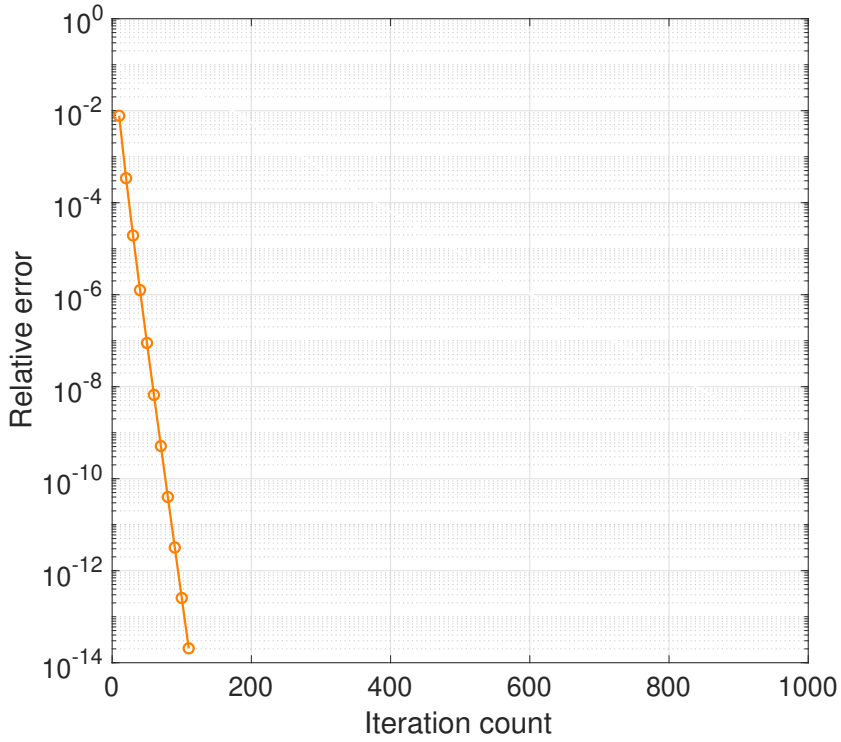
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$



Vanilla GD converges in $O\big(\log \frac{1}{\varepsilon}\big)$ iterations from a spectral initialization with barely enough samples information-theoretically.

Benign nonconvexity: global linear convergence

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$

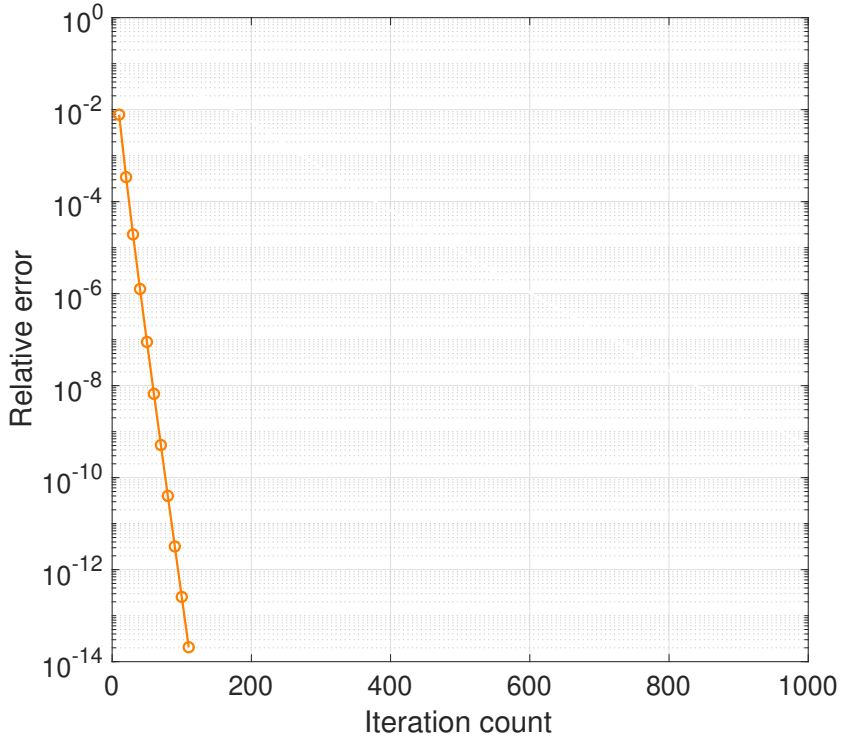


Similar results hold for many low-rank problems...

(Tu et al. '16, Netrapalli et al. '13, Candès, Li, Soltanolkotabi '14, Sun and Luo '15, Chen and Wainwright '15, Zheng and Lafferty '15, Ma et al. '17, ...)

What could go wrong?

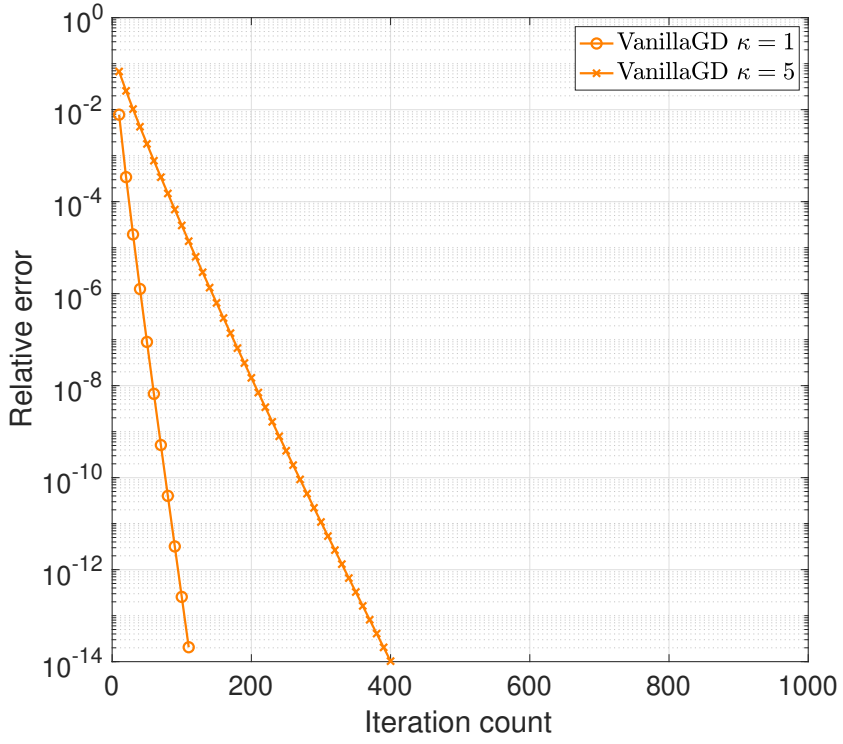
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$



Let us increase the condition number $\kappa(\boldsymbol{M}) = \frac{\sigma_1(\boldsymbol{M})}{\sigma_r(\boldsymbol{M})}$.

What could go wrong?

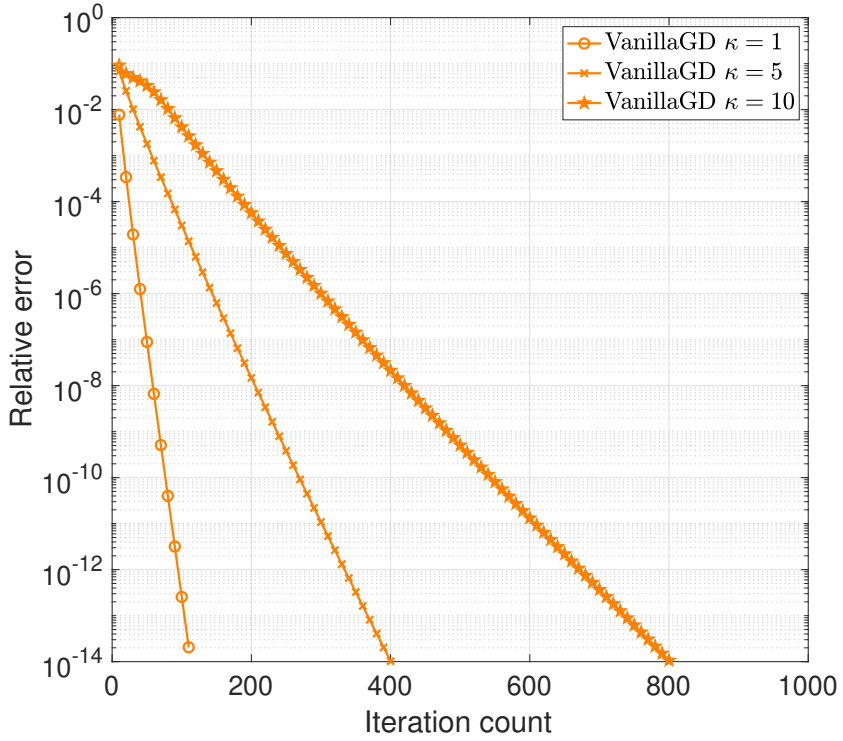
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$



Let us increase the condition number $\kappa(\boldsymbol{M}) = \dfrac{\sigma_1(\boldsymbol{M})}{\sigma_r(\boldsymbol{M})}$.

What could go wrong?

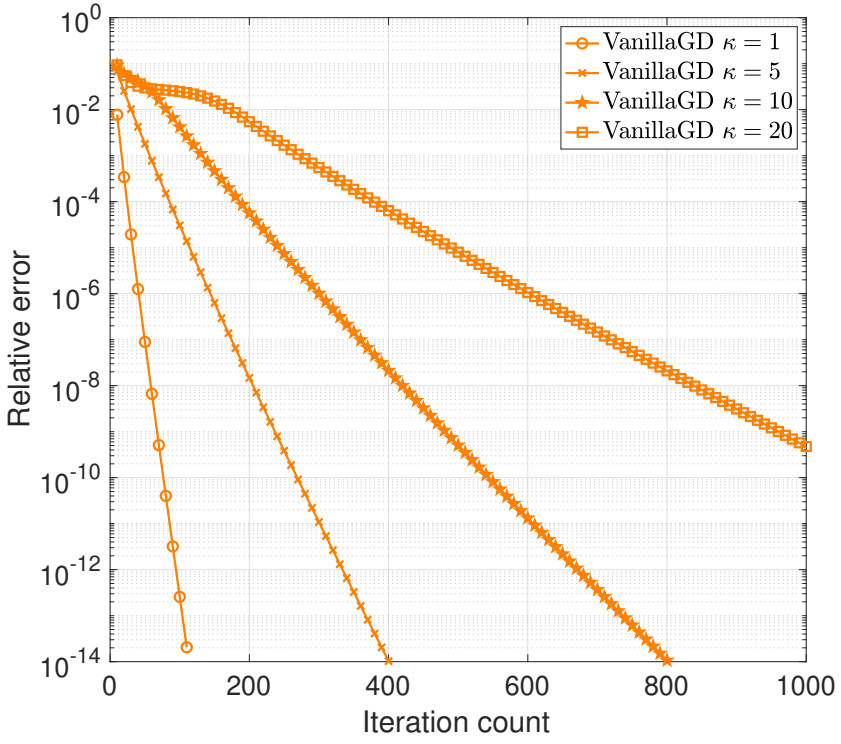
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$



Let us increase the condition number $\kappa(\boldsymbol{M}) = \dfrac{\sigma_1(\boldsymbol{M})}{\sigma_r(\boldsymbol{M})}$.

What could go wrong?

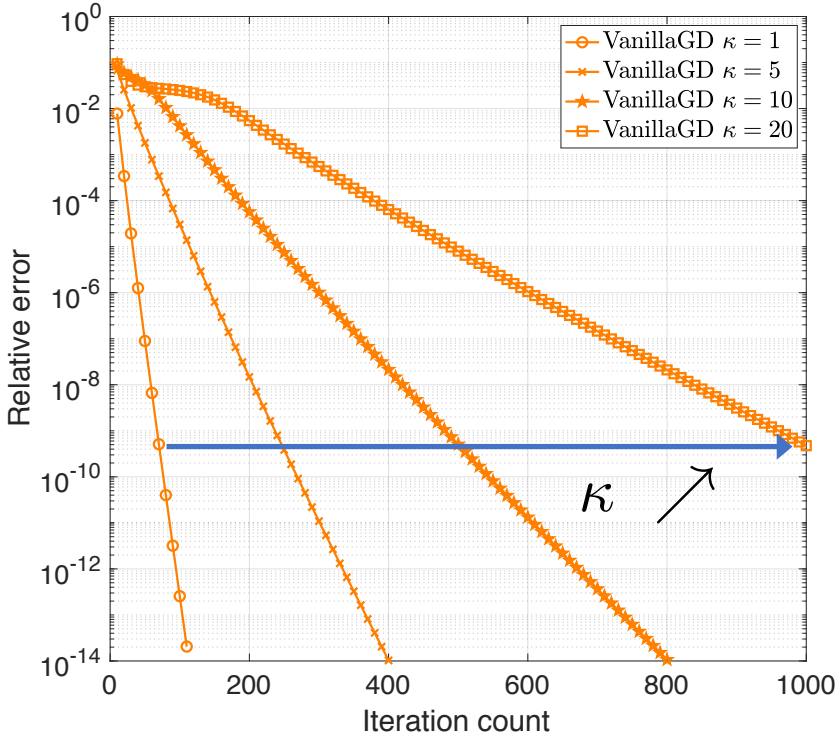
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$



Let us increase the condition number $\kappa(\boldsymbol{M}) = \frac{\sigma_1(\boldsymbol{M})}{\sigma_r(\boldsymbol{M})}$.

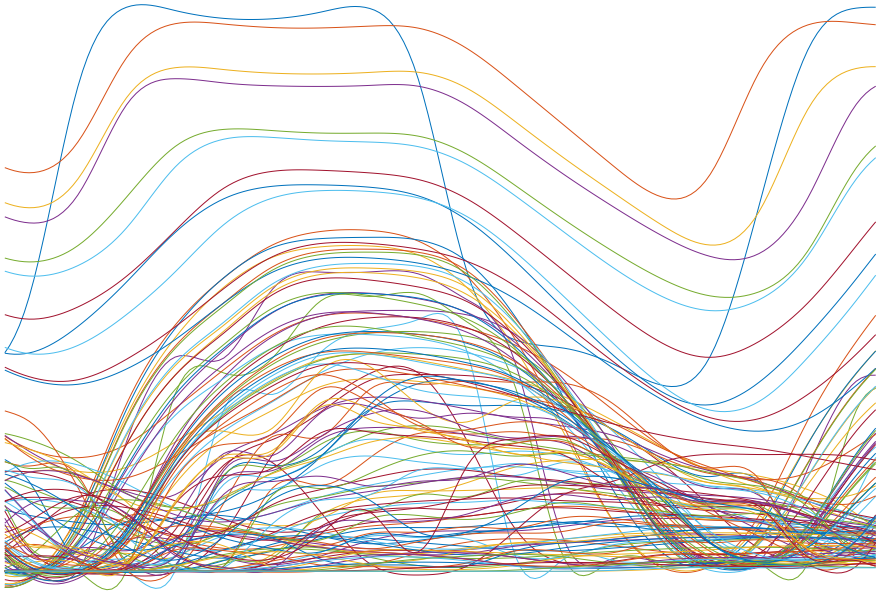
What could go wrong?

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{Y}^{\top} - \boldsymbol{M})\right\|_{\mathrm{F}}^{2}$$

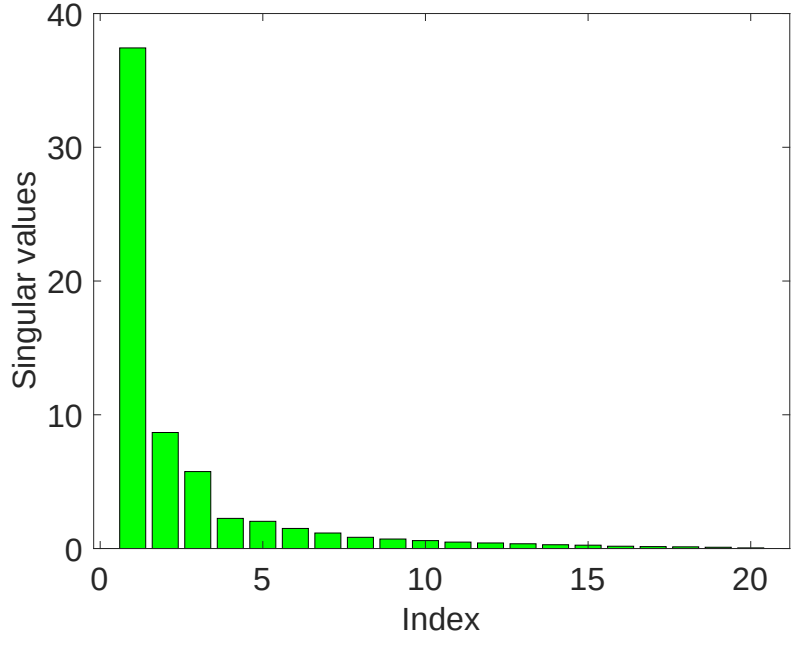


Vanilla GD converges in $O\left(\kappa \log \frac{1}{\varepsilon}\right)$ iterations.

Condition number can be large



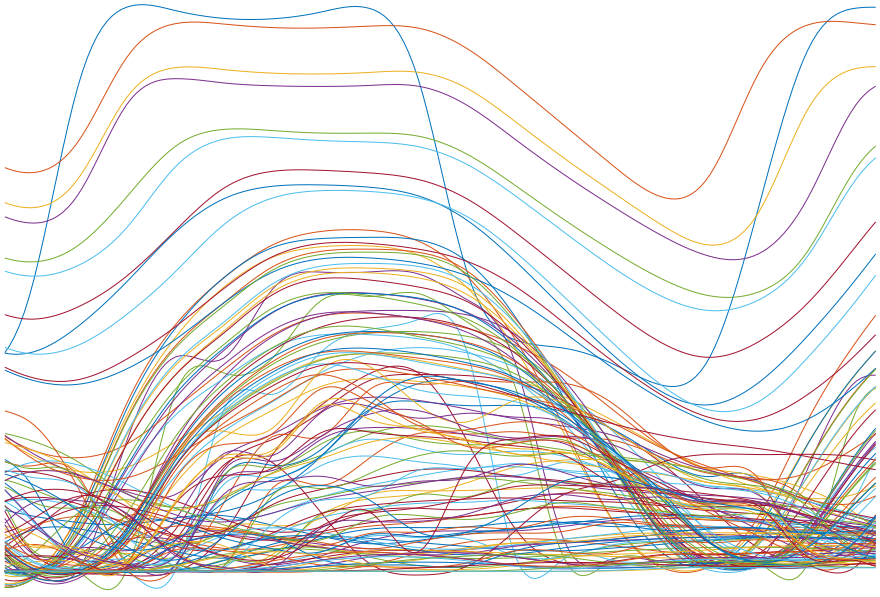
chlorine concentration levels
120 junctions, 180 time slots



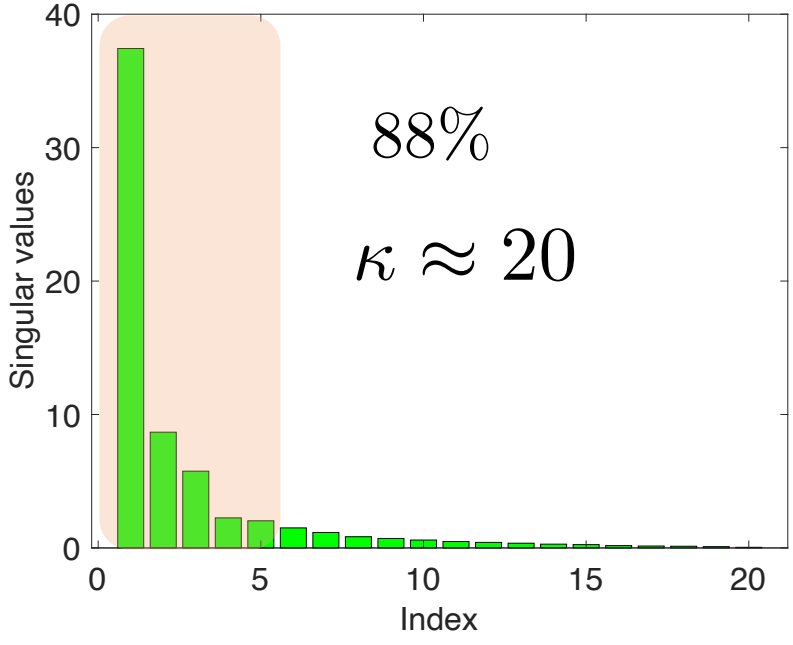
power-law spectrum

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



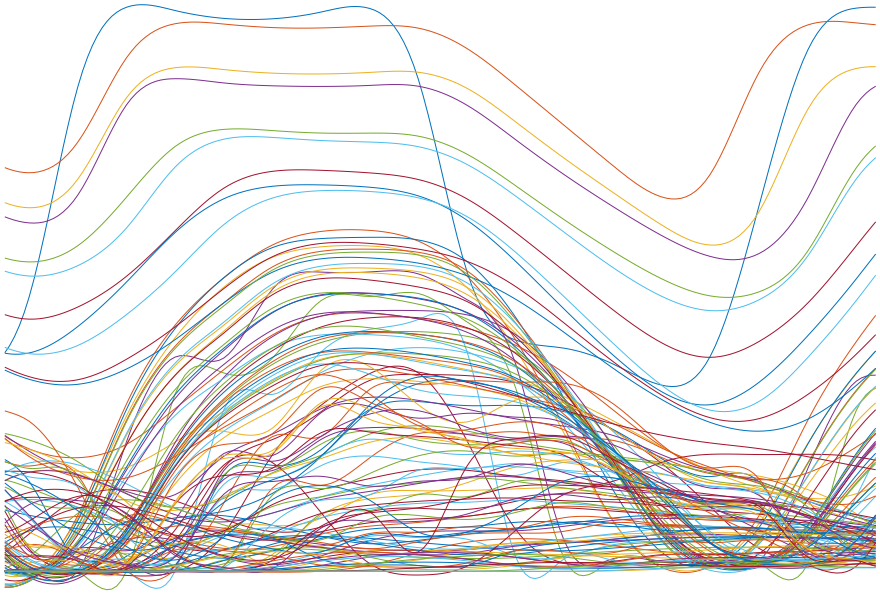
chlorine concentration levels
120 junctions, 180 time slots



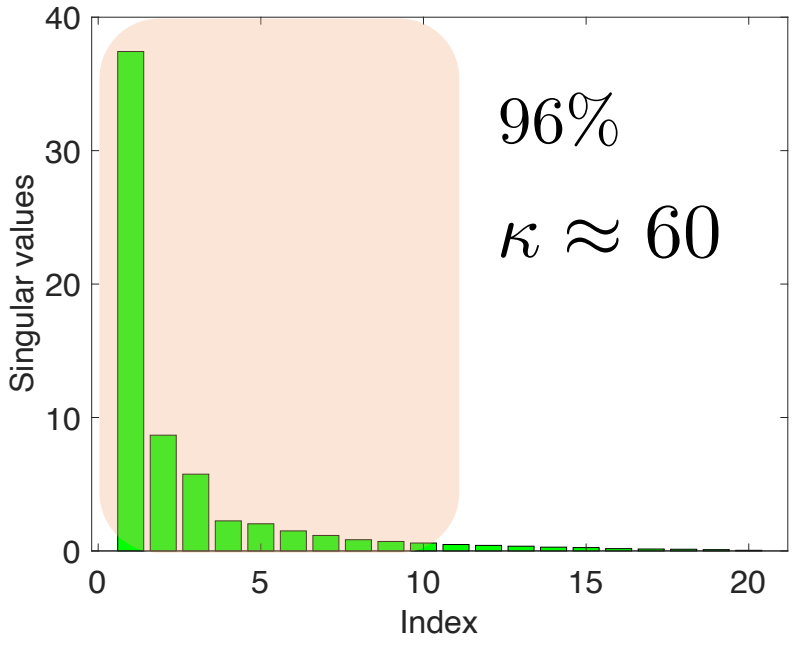
rank-5 approximation

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



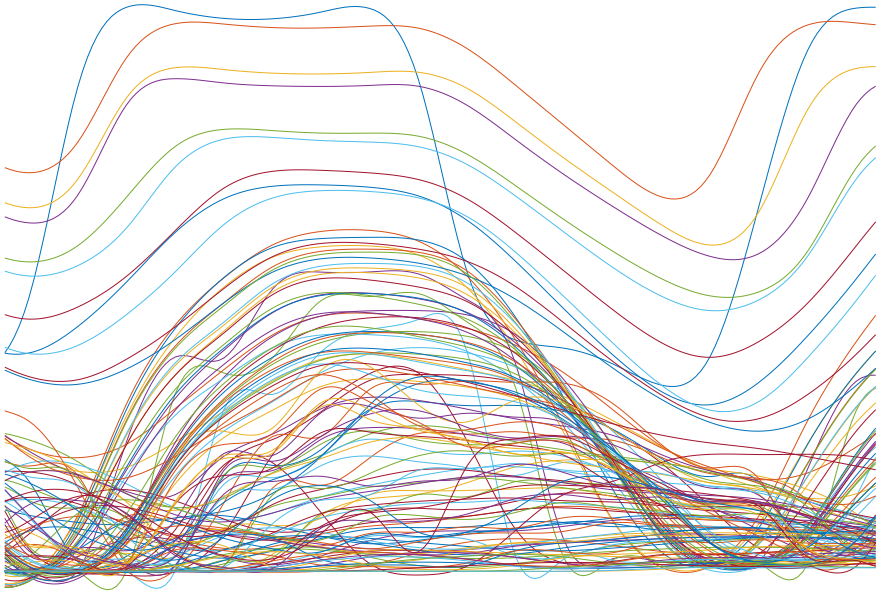
chlorine concentration levels
120 junctions, 180 time slots



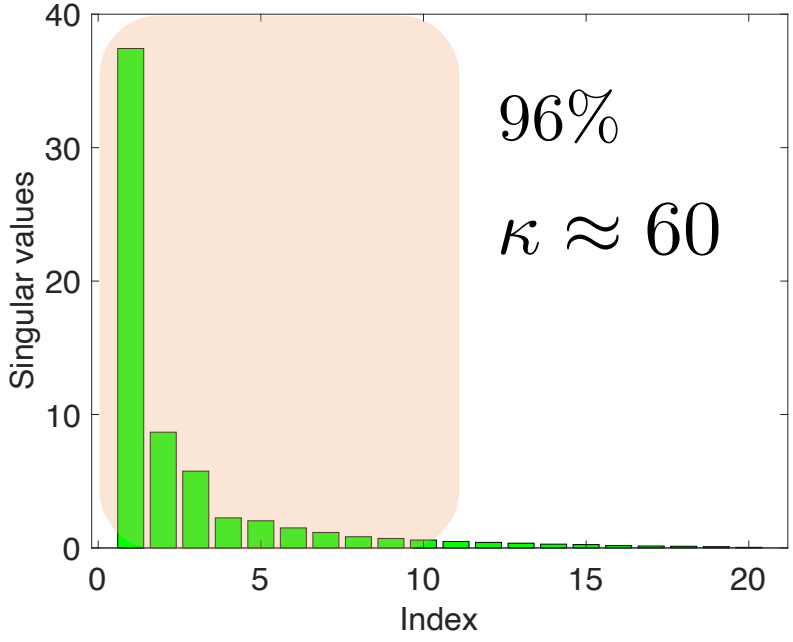
rank-10 approximation

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



chlorine concentration levels
120 junctions, 180 time slots

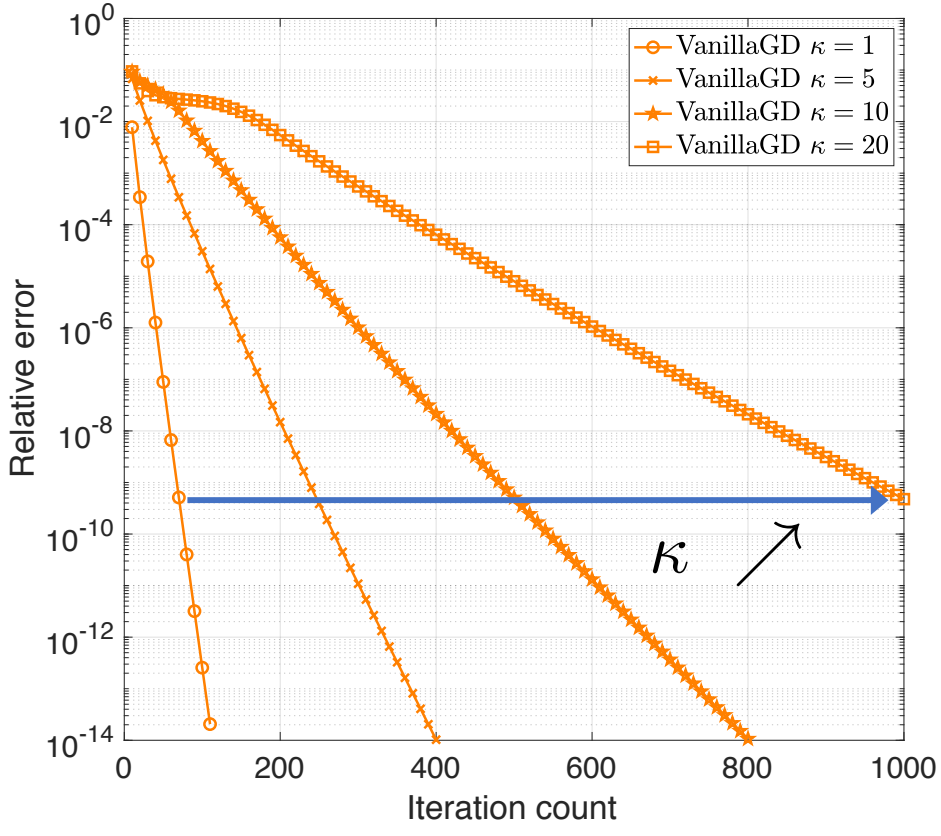


rank-10 approximation

Must mind the condition number!

Data source: `www.epa.gov/water-research/epanet`

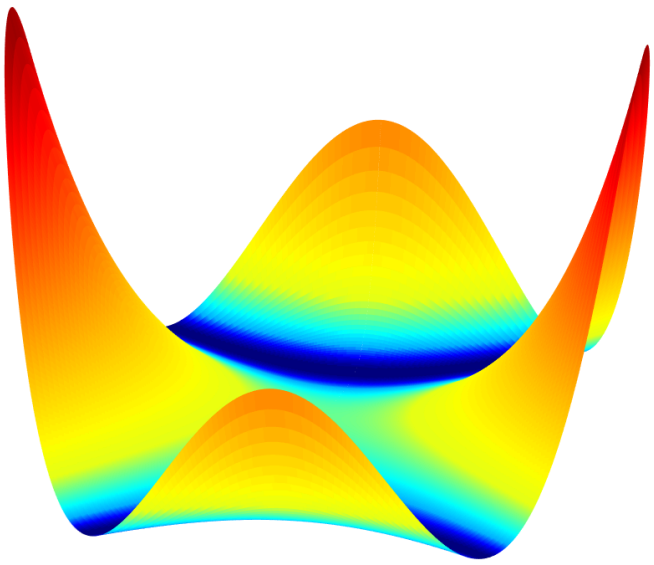
Getting rid of the condition number?



Can we accelerate the convergence rate of GD to $O(\log \frac{1}{\varepsilon})$*?*

Our recipe: scaled gradient descent (ScaledGD)

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top) \right\|_2^2$$



- **Spectral initialization:** find an initial point in the "basin of attraction".
- **Scaled gradient iterations:**

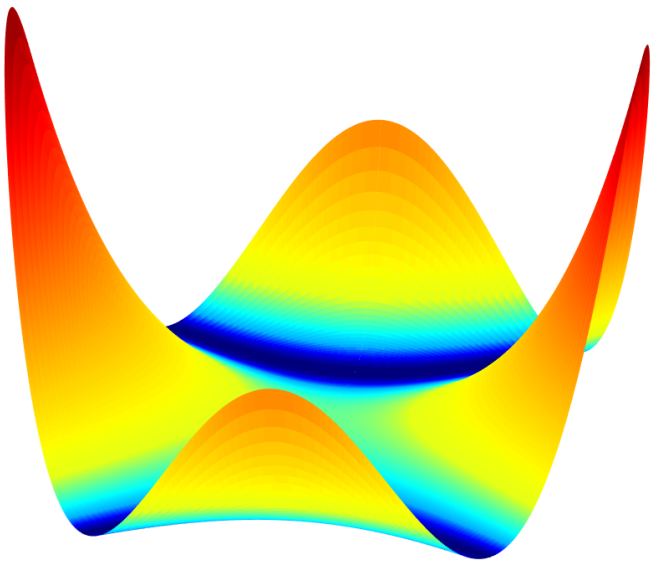
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta \nabla_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

for $t = 0, 1, \ldots$

Our recipe: scaled gradient descent (ScaledGD)

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top) \right\|_2^2$$



- **Spectral initialization:** find an initial point in the "basin of attraction".
- **Scaled gradient iterations:**

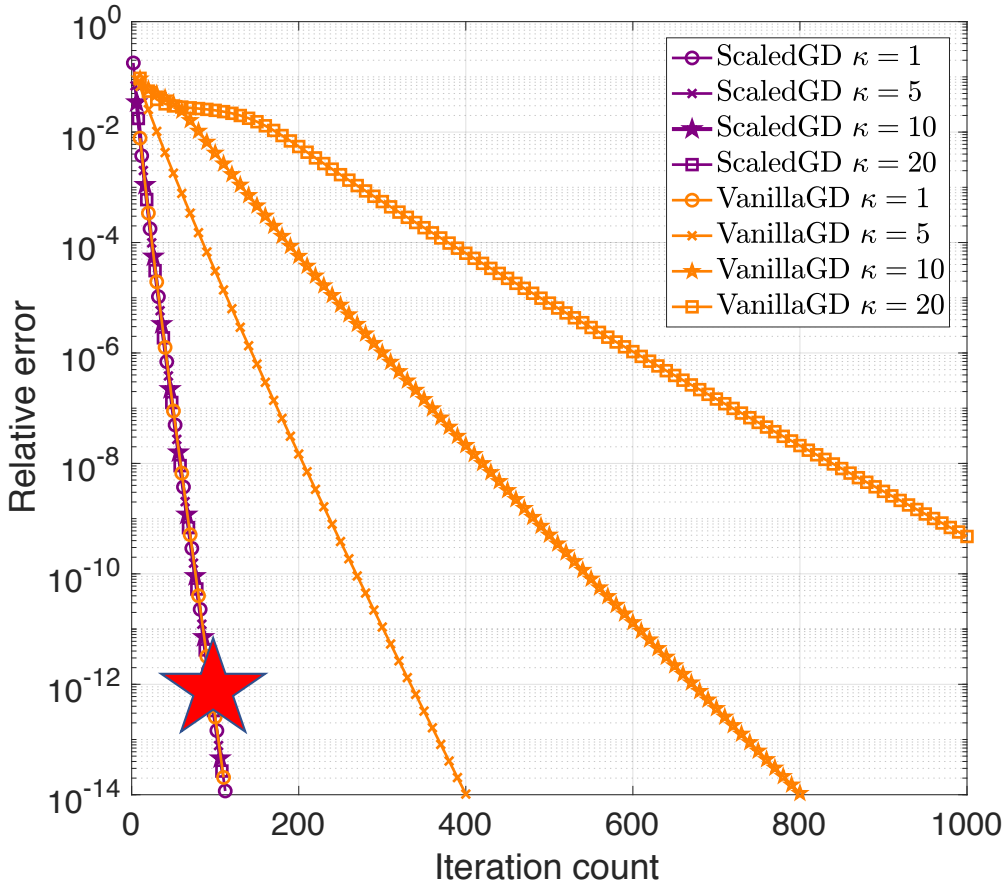
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta \nabla_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

for $t = 0, 1, \ldots$

ScaledGD is a *preconditioned* gradient method *without* balancing regularization!

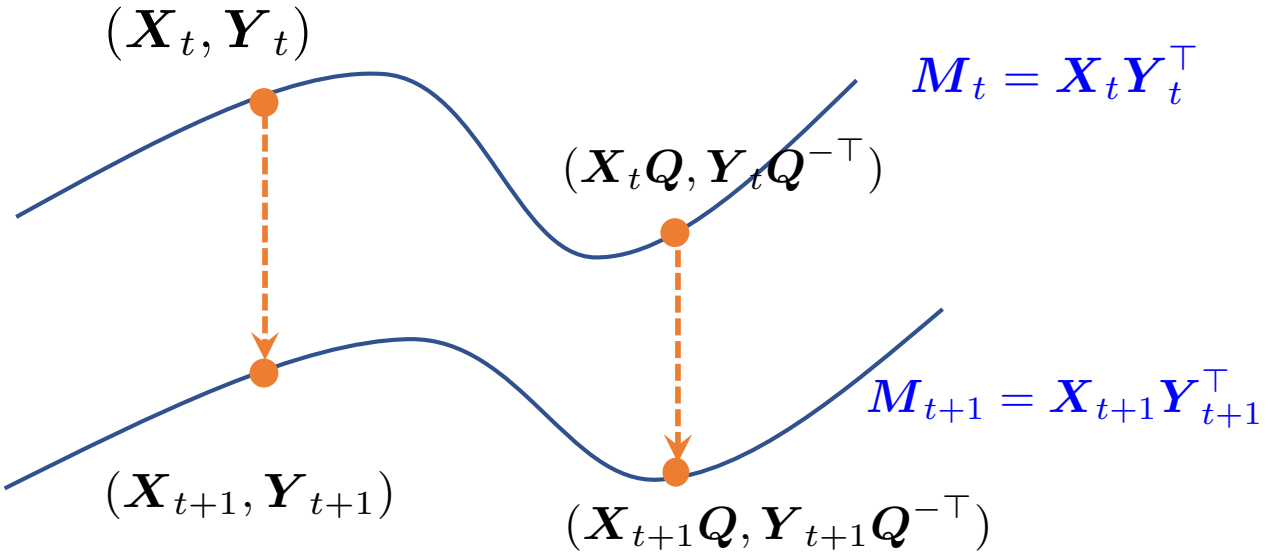
ScaledGD for low-rank matrix completion



Huge computational saving: ScaledGD converges in an κ-independent manner with a minimal overhead!

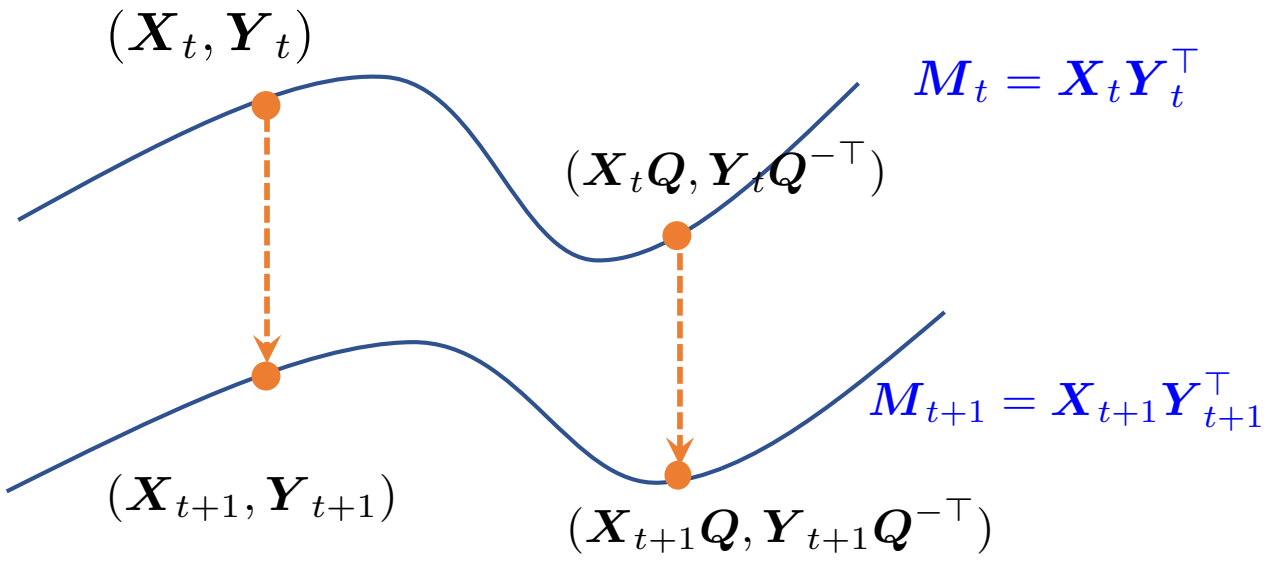
A closer look at ScaledGD

Invariance to invertible transforms: (Tanner and Wei, '16; Mishra '16)



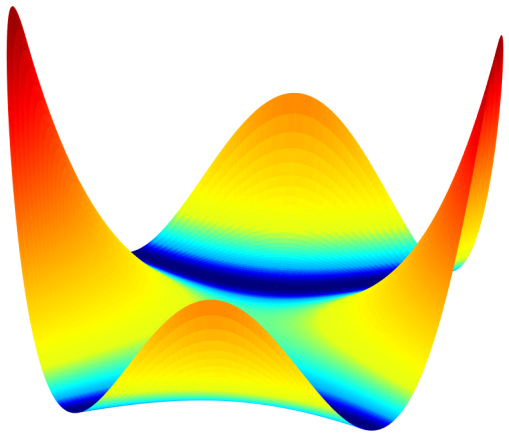
A closer look at ScaledGD

Invariance to invertible transforms: (Tanner and Wei, '16; Mishra '16)



New distance metric as Lyapunov function:

$$\operatorname{dist}^2\left(\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{bmatrix}, \begin{bmatrix}\boldsymbol{X}_\star\\ \boldsymbol{Y}_\star\end{bmatrix}\right) = \inf_{\boldsymbol{Q}\in \mathrm{GL}(r)} \left\|(\boldsymbol{X}\boldsymbol{Q} - \boldsymbol{X}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathrm{F}}^2 + \left\|(\boldsymbol{Y}\boldsymbol{Q}^{-\top} - \boldsymbol{Y}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathrm{F}}^2$$



+ a careful trajectory-based analysis

Theoretical guarantees of ScaledGD

Theorem (Tong, Ma and Chi, JMLR 2021)

For low-rank matrix sensing with i.i.d. Gaussian design, ScaledGD with spectral initialization achieves

$$\|\boldsymbol{X}_t\boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *iterations;*
- **Statistical:** *the sample complexity satisfies*

$$m \gtrsim (n_1 + n_2) r^2 \kappa^2.$$

Theoretical guarantees of ScaledGD

Theorem (Tong, Ma and Chi, JMLR 2021)

For low-rank matrix sensing with i.i.d. Gaussian design, ScaledGD with spectral initialization achieves

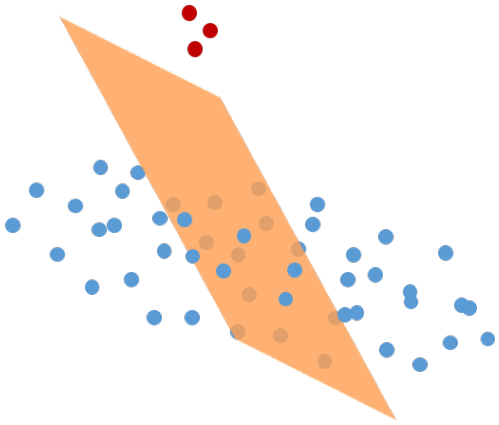
$$\|\boldsymbol{X}_t\boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *iterations;*
- **Statistical:** *the sample complexity satisfies*

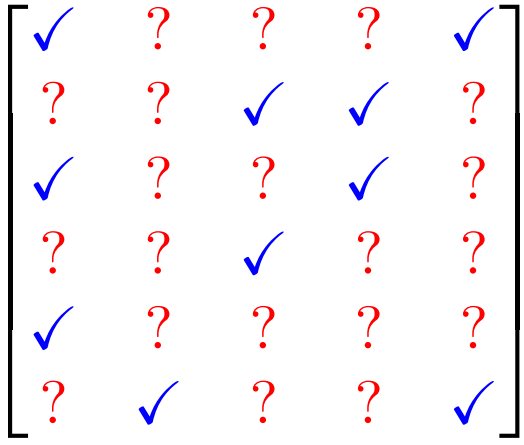
$$m \gtrsim (n_1 + n_2) r^2 \kappa^2.$$

Strict improvement over vanilla GD: provable acceleration at the same sample complexity!

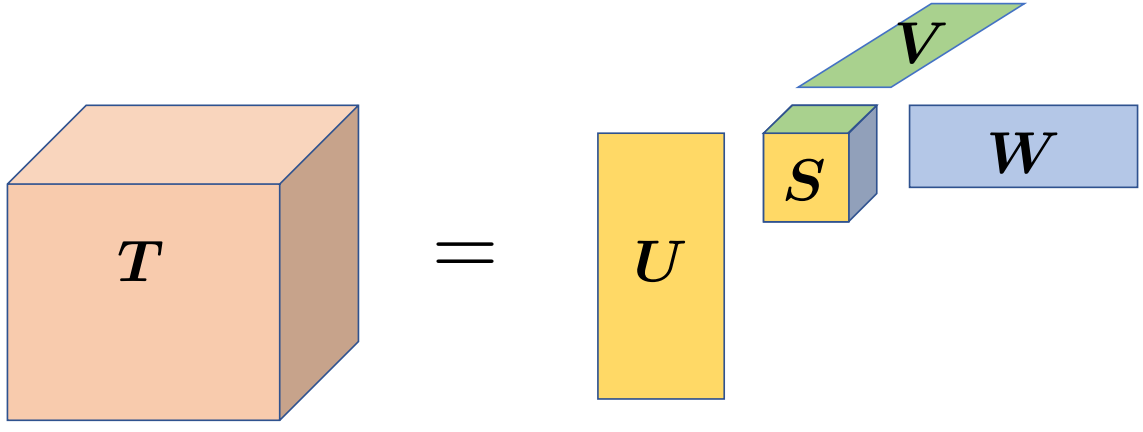
ScaledGD works more broadly



robust PCA



matrix completion

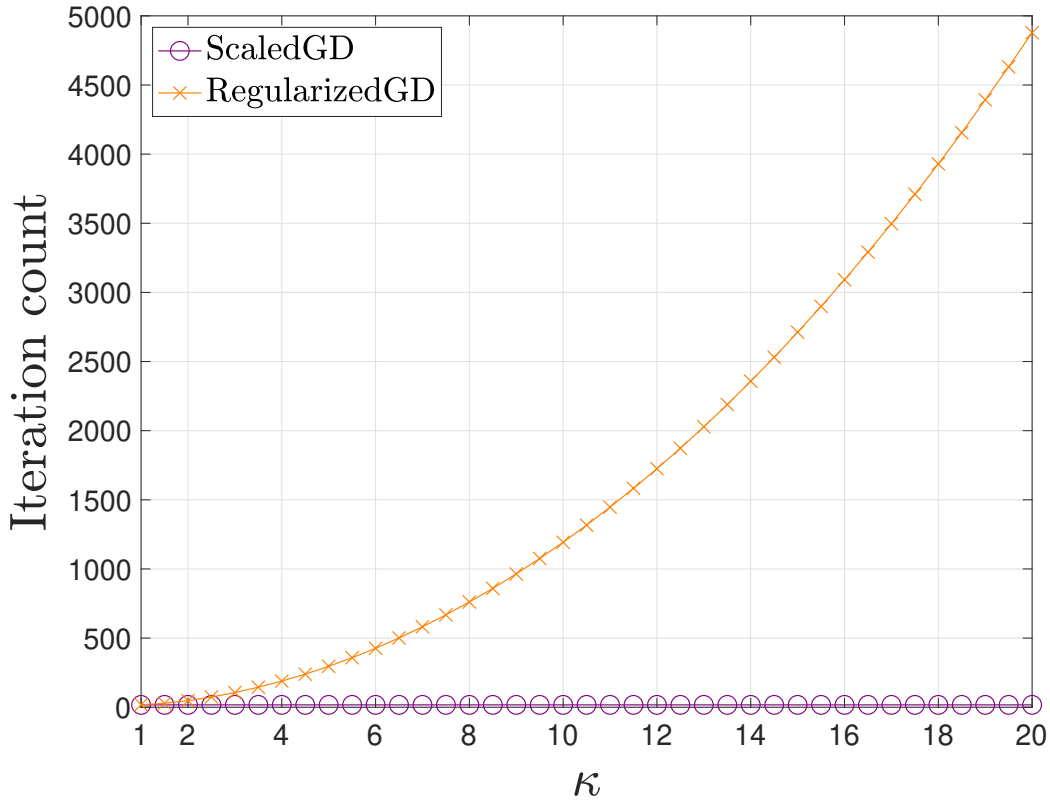


Tucker tensor recovery

Huge computation savings at comparable sample complexities!

More gain for tensors

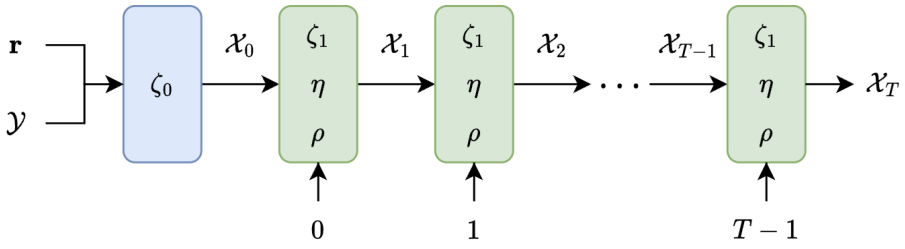
$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \; f(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{T})\right\|_{\mathrm{F}}^2$$



The benefit of ScaledGD is even more evident for tensors!

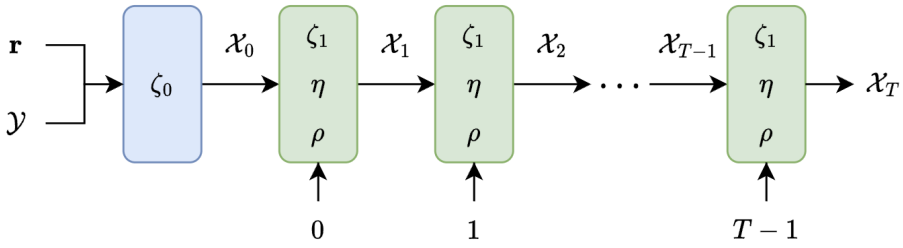
Saliency detection in materials data

Unrolling ScaledGD + self-supervised learning for tensor RPCA



Saliency detection in materials data

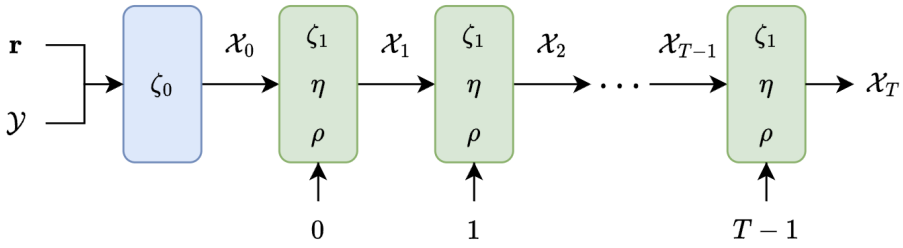
Unrolling ScaledGD + self-supervised learning for tensor RPCA



some materials data

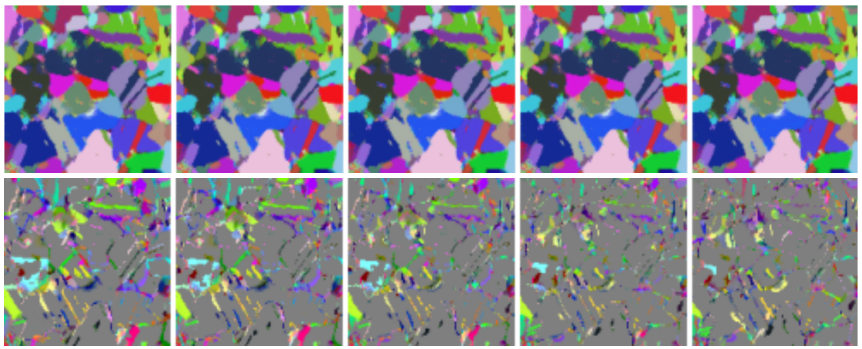
Saliency detection in materials data

Unrolling ScaledGD + self-supervised learning for tensor RPCA



low-rank + sparse decomposition

some materials data

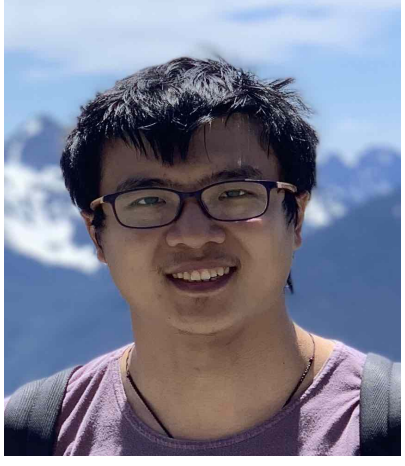


"Deep Unfolded Tensor Robust PCA with Self-supervised Learning", Dong, Shah, Donegan, and Chi, ICASSP 2023.

Preconditioning meets generalization in overparameterized low-rank matrix sensing

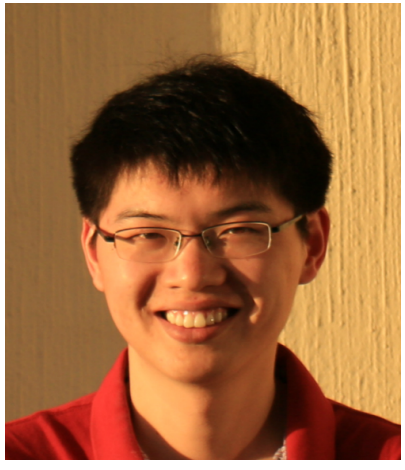
Xingyu Xu

CMU



Yandi Shen

Yale



Cong Ma

UChicago

What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

Misspecification by overparameterization:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n \times r'}, \qquad r' > r$$

What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

Misspecification by overparameterization:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n \times r'}, \qquad r' > r$$

ScaledGD:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

analysis break down and might be unstable...

What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

Misspecification by overparameterization:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n\times r'}, \qquad r' > r$$

ScaledGD(λ):

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t + \lambda \boldsymbol{I})^{-1}}_{\texttt{preconditioner}}$$

add regularization to stabilize the preconditioner

Does preconditioning hurt generalization?

- Infinitely many global minima, not all generalize
- Can we still guarantee generalization?

optimization

generalization

When Does Preconditioning Help or Hurt Generalization?

*Shun-ichi Amari[1], Jimmy Ba[2,3], Roger Grosse[2,3], Xuechen Li[4], Atsushi Nitanda[5,6], Taiji Suzuki[5,6], Denny Wu[2,3], Ji Xu[7]

[1]RIKEN CBS, [2]University of Toronto, [3]Vector Institute, [4]Google Research, Brain Team, [5]University of Tokyo, [6]RIKEN AIP, [7]Columbia University

amari@brain.riken.jp, {jba,rgrosse,lxuechen,dennywu}@cs.toronto.edu, {nitanda,taiji}@mist.i.u-tokyo.ac.jp, jixu@cs.columbia.edu

Theoretical guarantees

Theorem (Xu, Shen, Ma, Chi, ICML 2023)

For low-rank matrix sensing with i.i.d. Gaussian design, ScaledGD(λ) with $\lambda \asymp \sigma_{\min}(\boldsymbol{M})$, $\eta \asymp 1$, and small random initialization $\boldsymbol{X}_0 \sim \alpha\mathcal{N}(0, 1/n)$ with sufficiently small α achieves

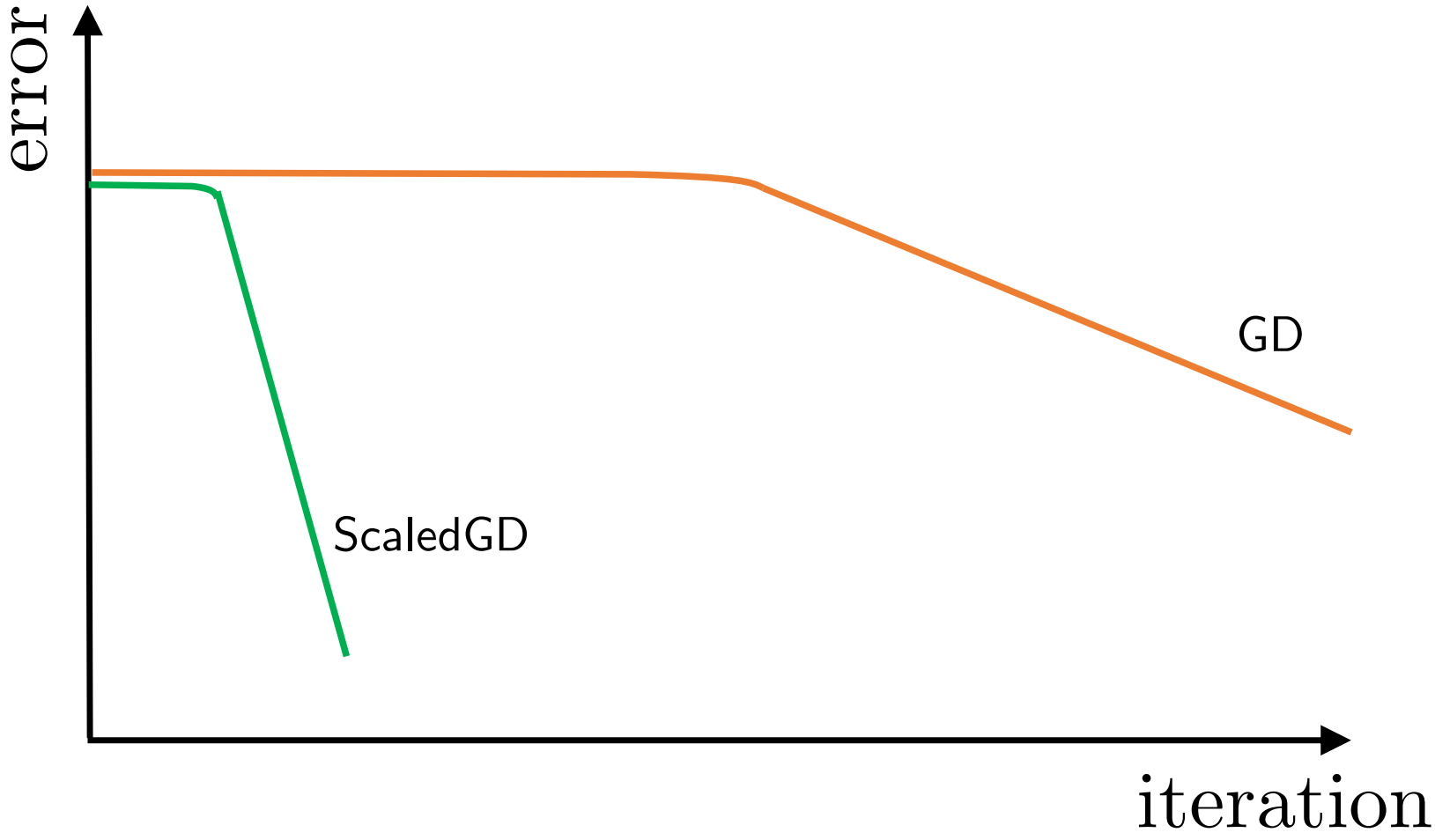
$$\|\boldsymbol{X}_t\boldsymbol{X}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **Computational:** *within* $O\big(\log\kappa \log(\kappa n) + \log\frac{1}{\varepsilon}\big)$ *iterations;*
- **Statistical:** *the sample complexity satisfies*

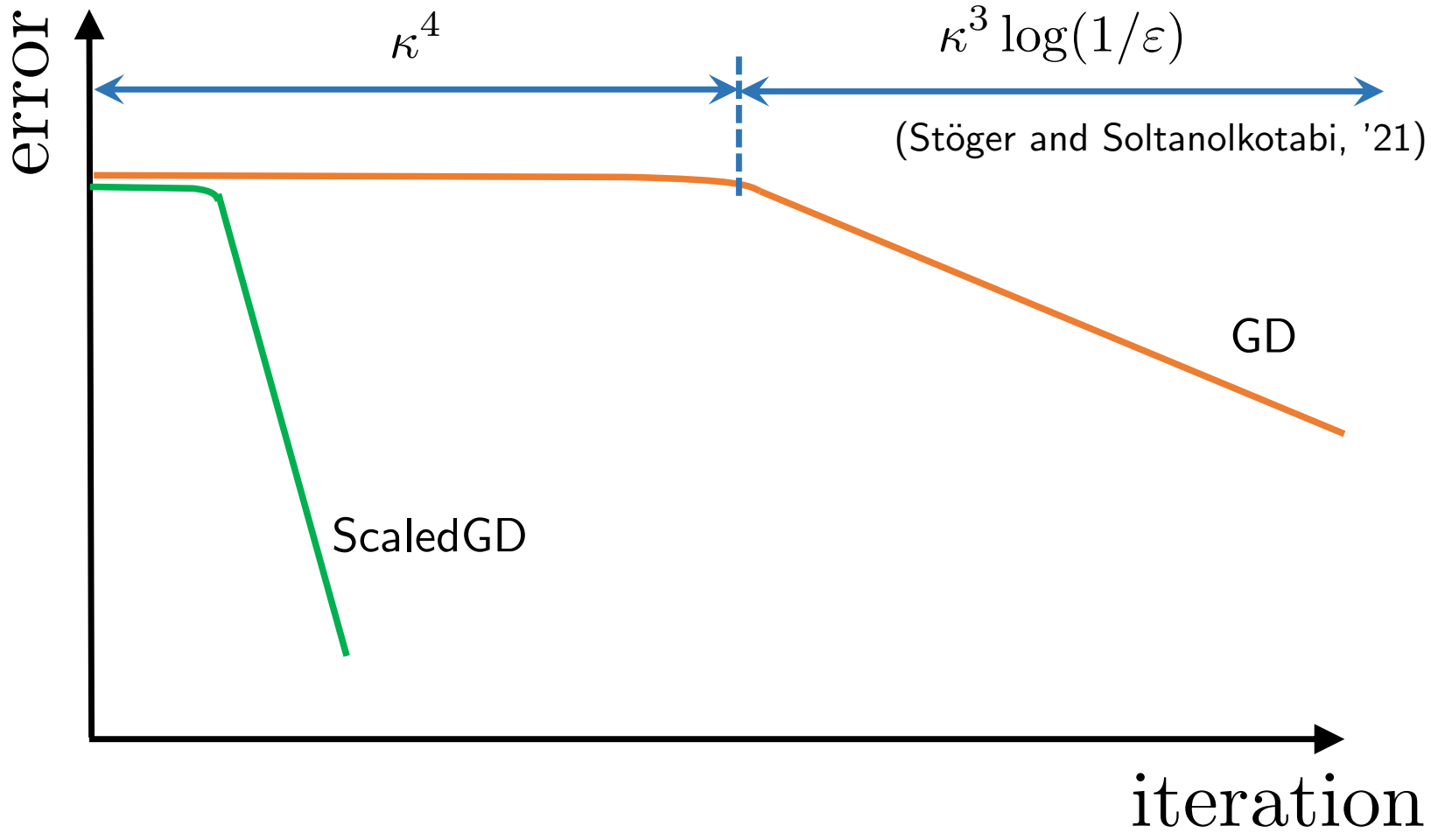
$$m \gtrsim nr^2 \mathit{poly}(\kappa).$$

- Our analysis also enables exact convergence under random initialization with correct rank specification.

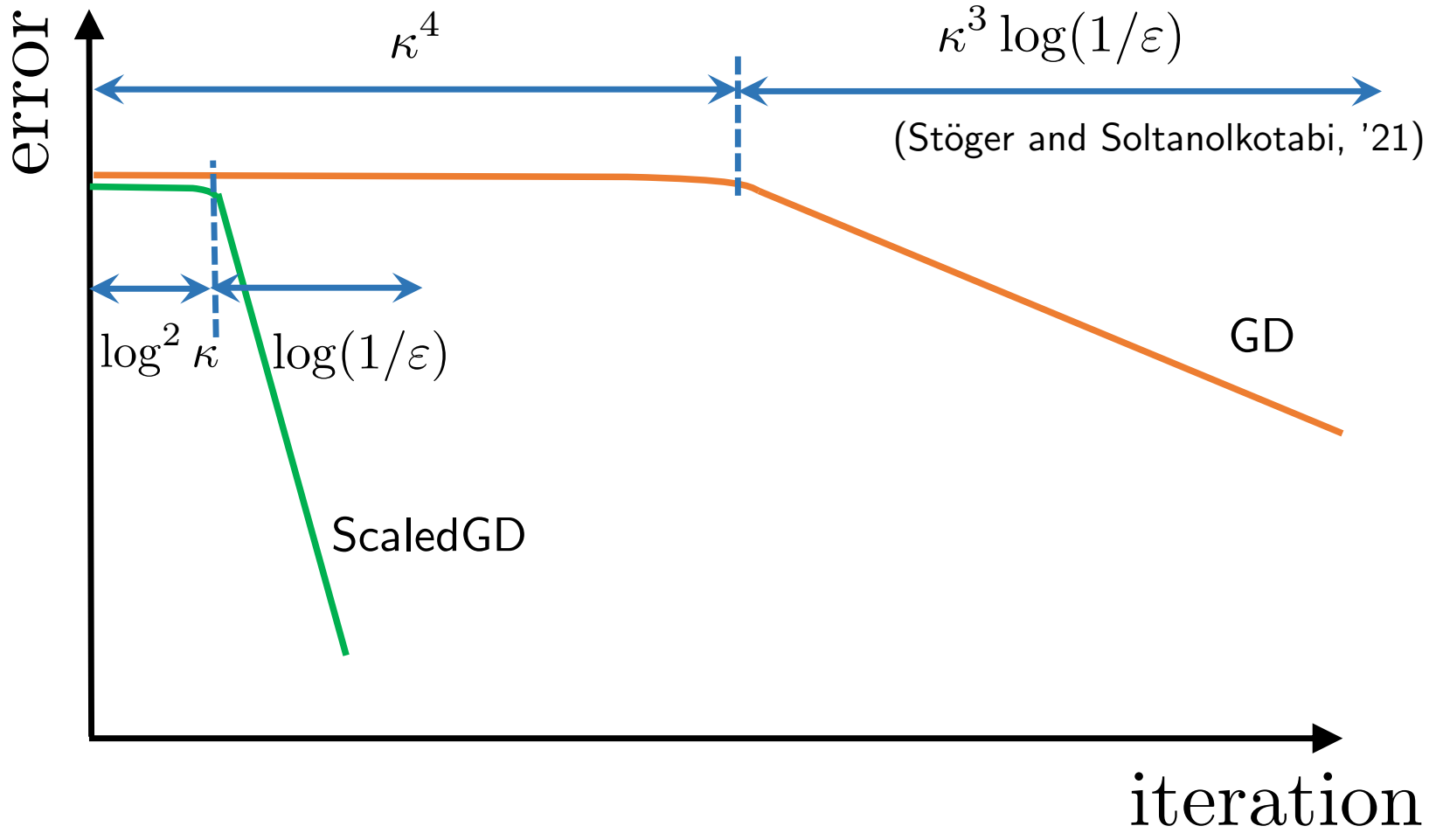
Comparison with overparameterized GD



Comparison with overparameterized GD

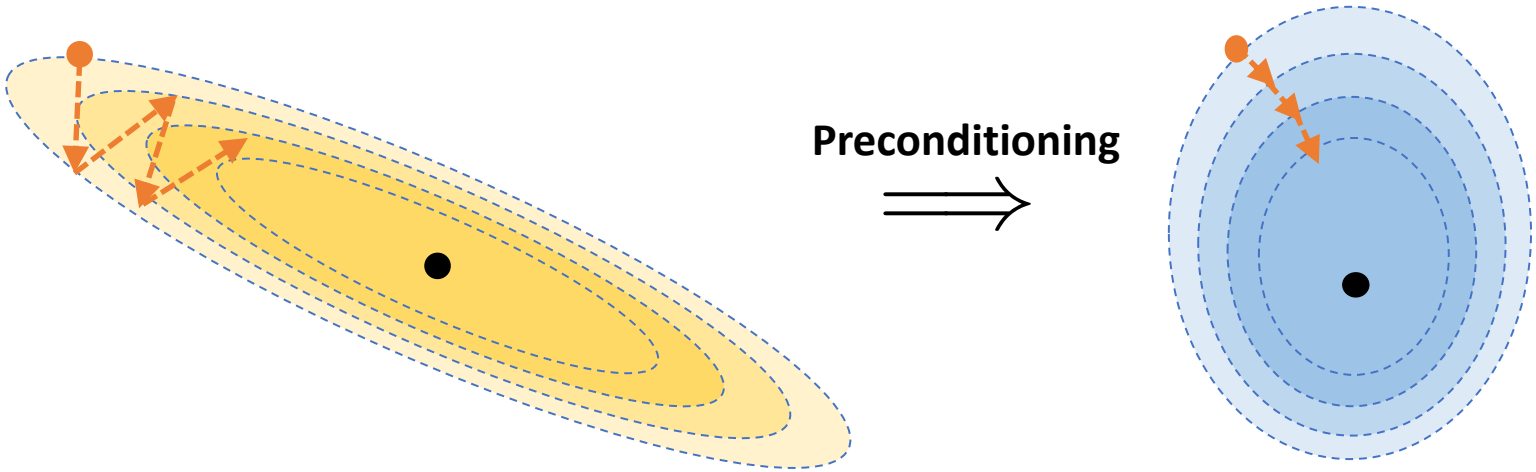


Comparison with overparameterized GD



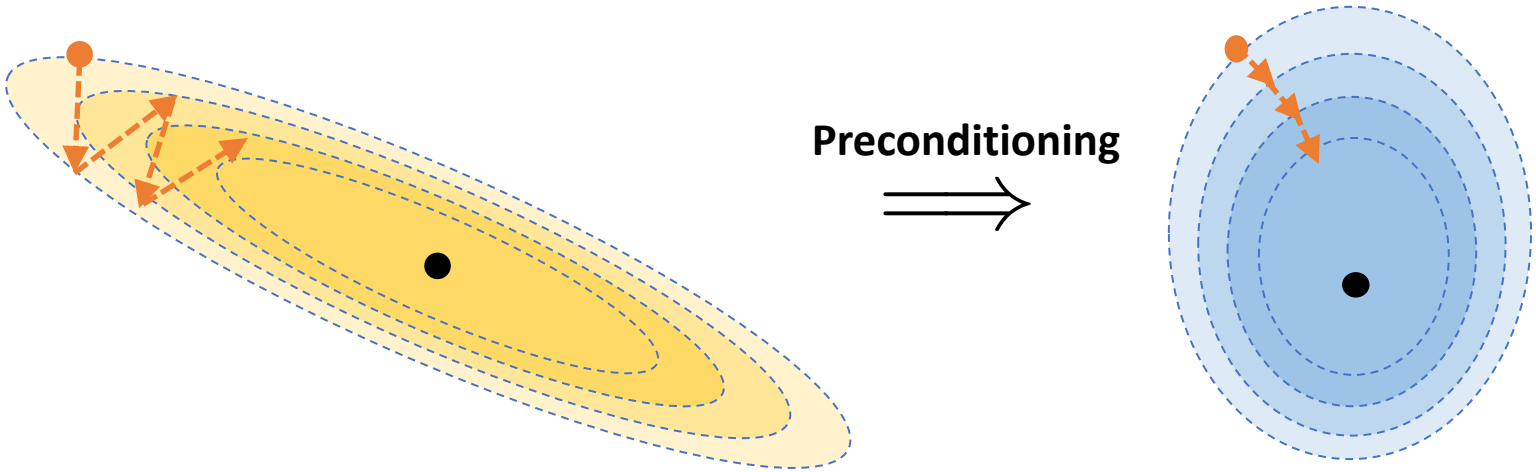
ScaledGD picks up the signal component much faster than GD even from small random initialization!

Summary: preconditioning helps!



Preconditioning can dramatically increase the computational efficiency of vanilla gradient methods without hurting statistical efficiency

Summary: preconditioning helps!



Preconditioning can dramatically increase the computational efficiency of vanilla gradient methods without hurting statistical efficiency

Future directions:

- generalizing the idea of ScaledGD to other learning problems

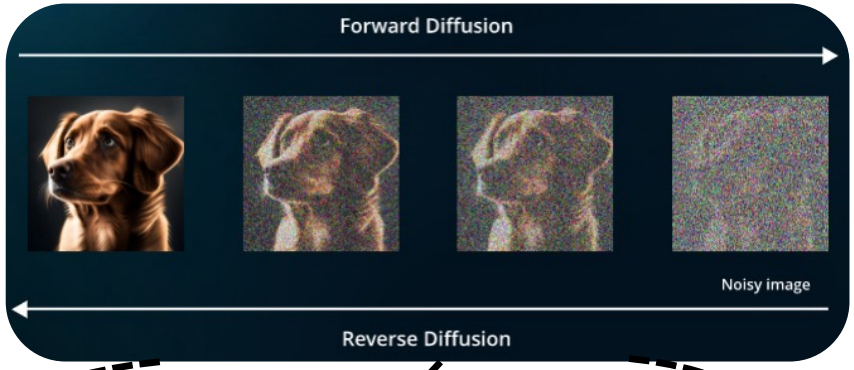
Score-based diffusion models for inverse problems

Xingyu Xu
CMU

State-of-the-art diffusion models

Inspired by nonequilibrium thermodynamics

Diffusion models

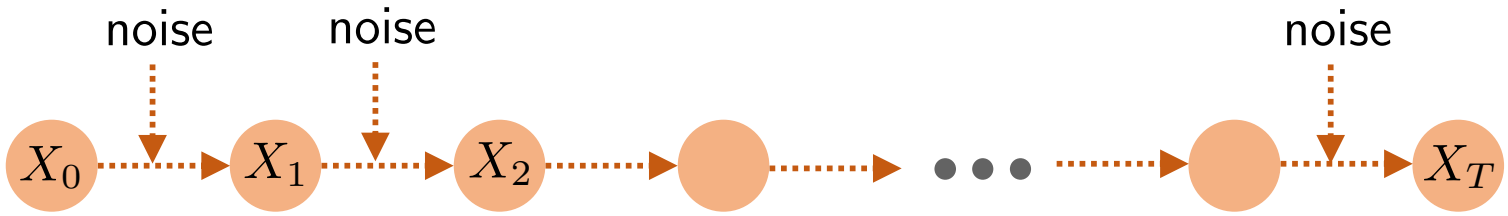


Stable Diffusion

DALLE

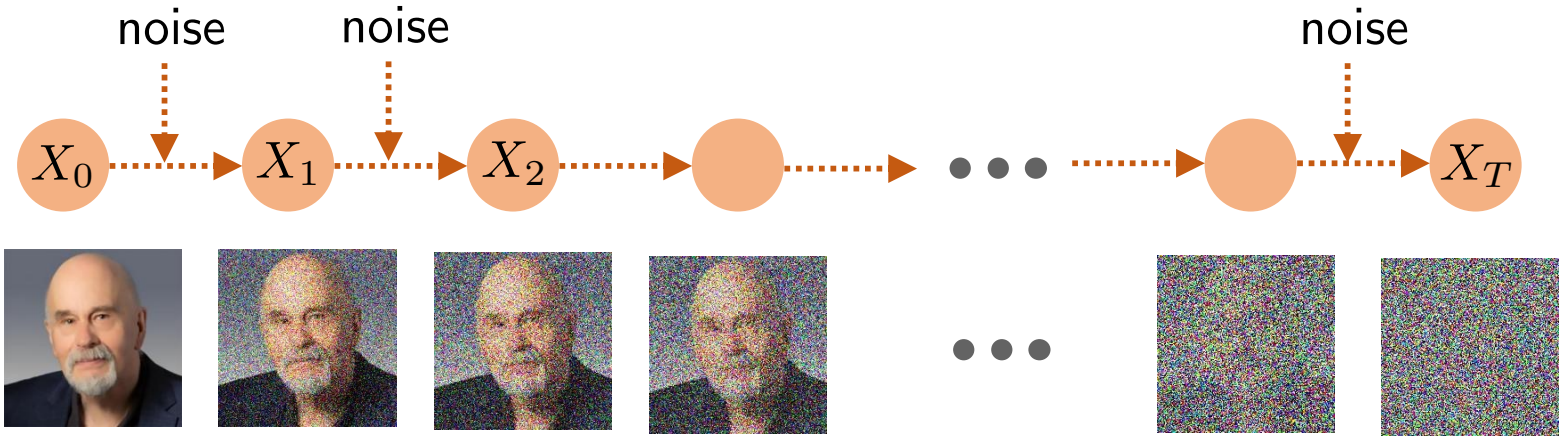
Sora

A high-level description of diffusion models



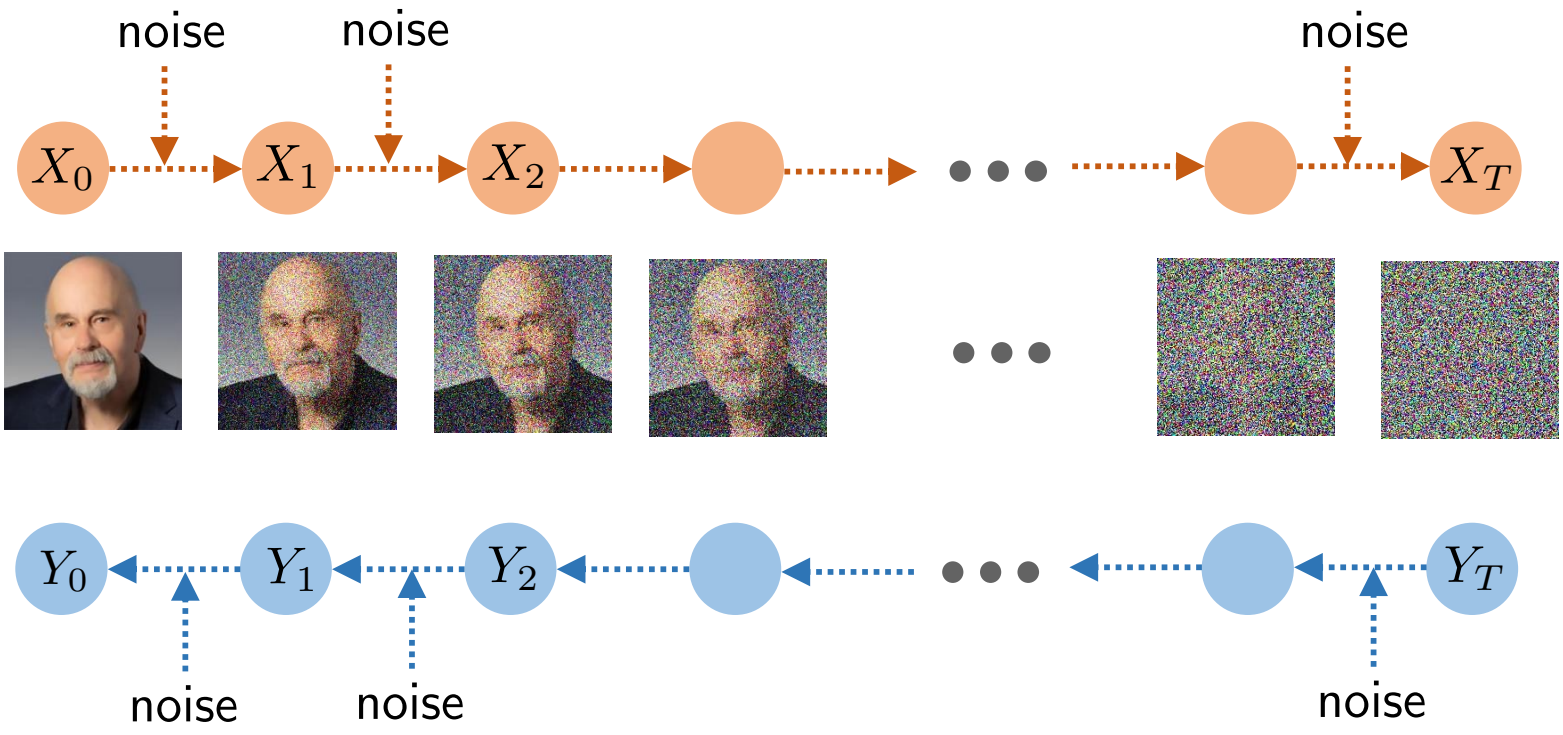
- **forward process:** (progressively) diffuse data into noise

A high-level description of diffusion models



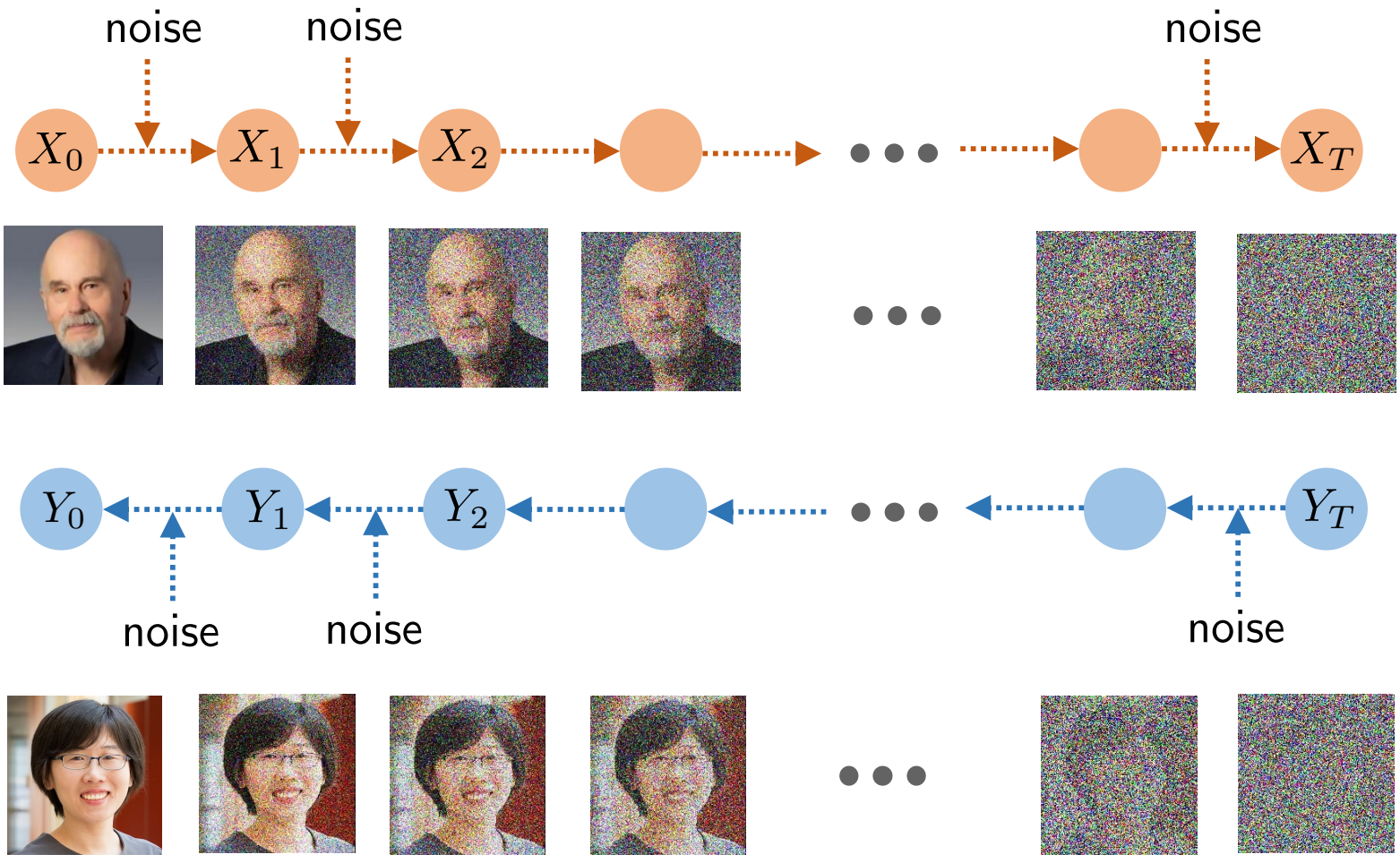
- **forward process:** (progressively) diffuse data into noise

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



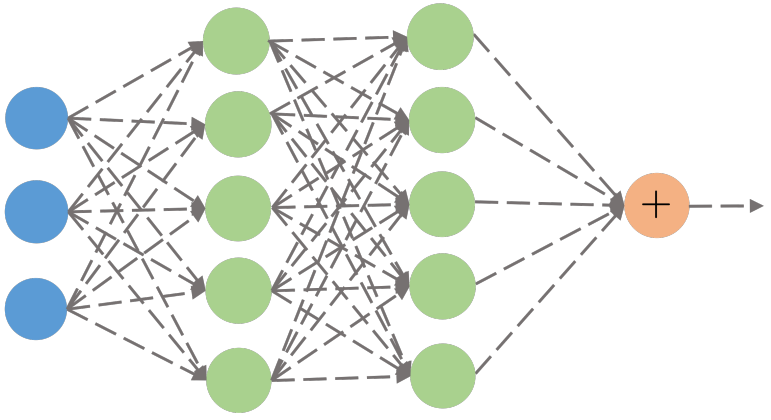
- **forward process:** (progressively) diffuse data into noise
- **reverse process:** convert pure noise into data-like distributions

Score is all you need (Anderson'82)

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

Score is all you need (Anderson'82)

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

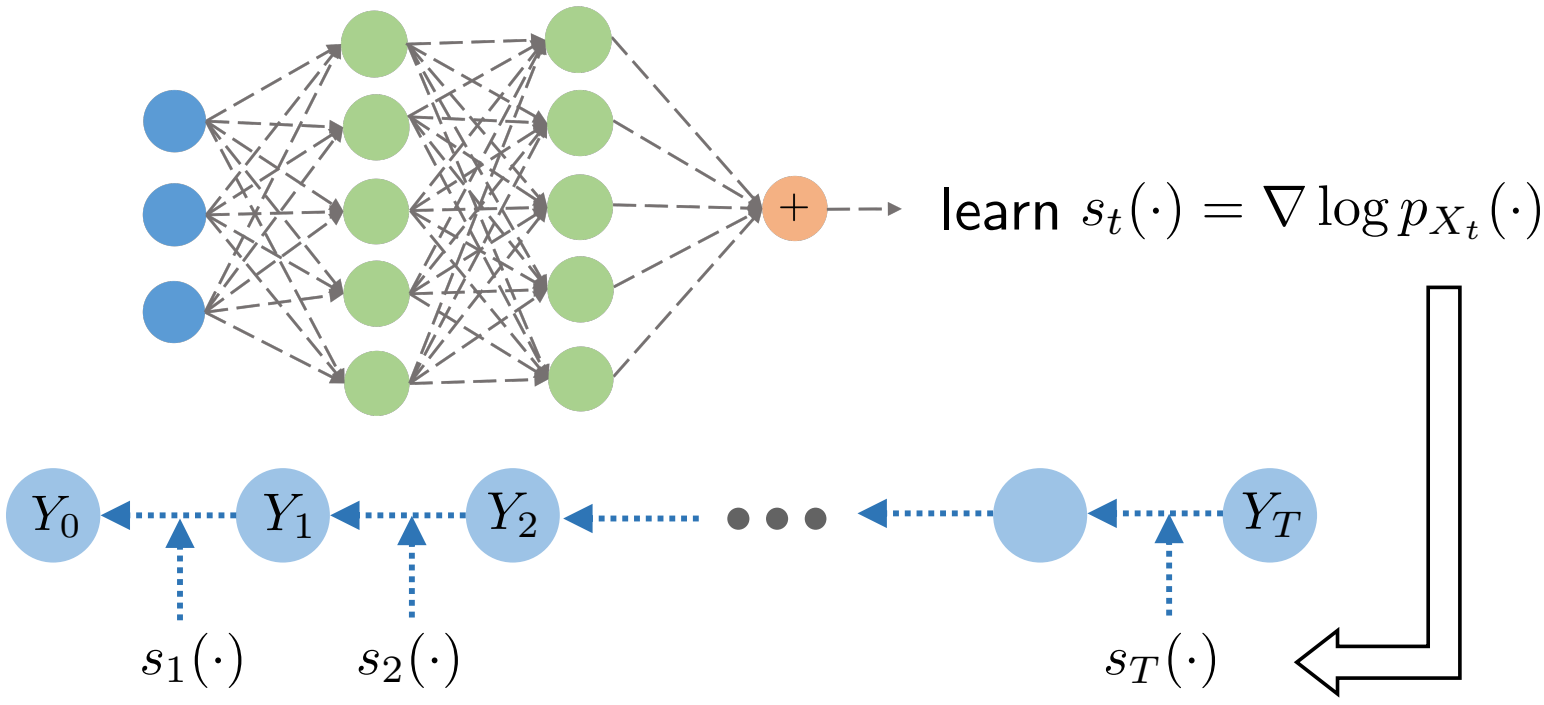


learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

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

Score is all you need (Anderson'82)

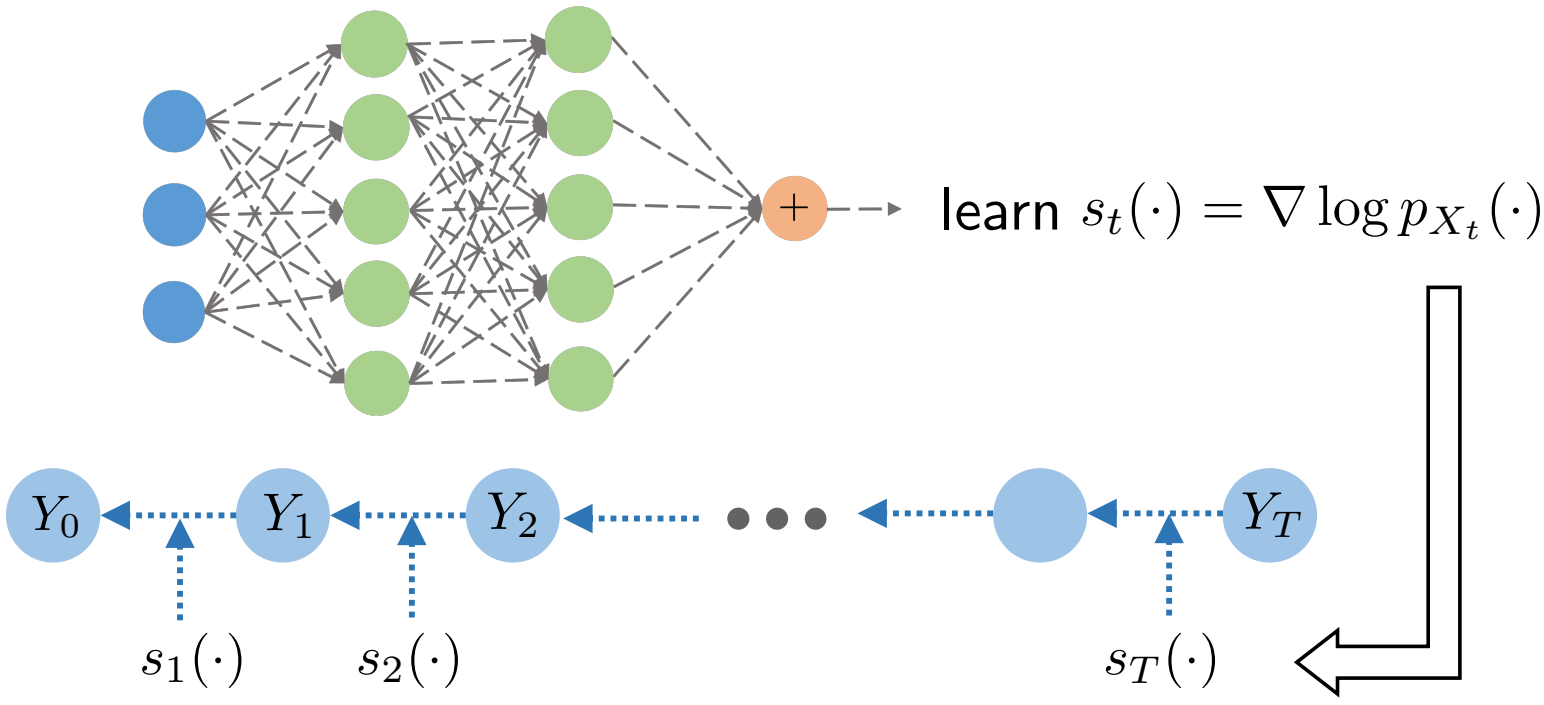
- **score functions** of marginals of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$



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 is all you need (Anderson'82)

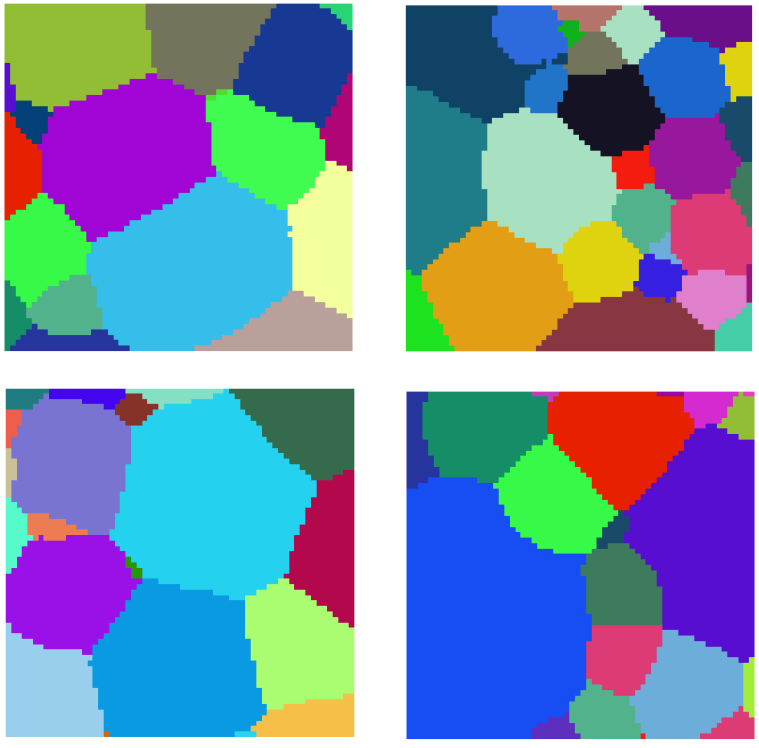
- **score functions** of marginals of forward process: $\underbrace{\nabla \log p_{X_t}(X)}_{\text{w.r.t. } X}$



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 $\{s_t(\cdot)\}$

Generating materials imagery using diffusion models

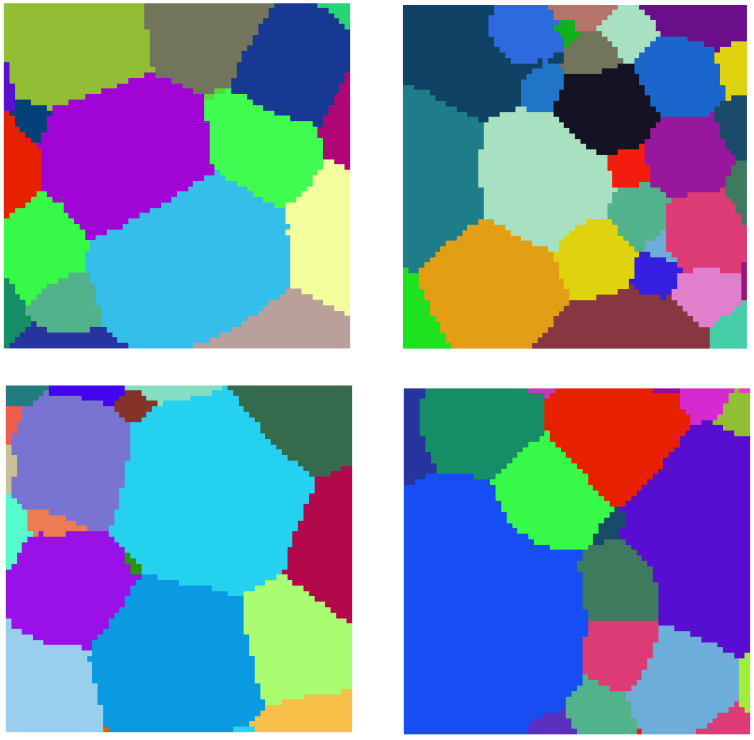
Diffusion model generates EBSD imagery



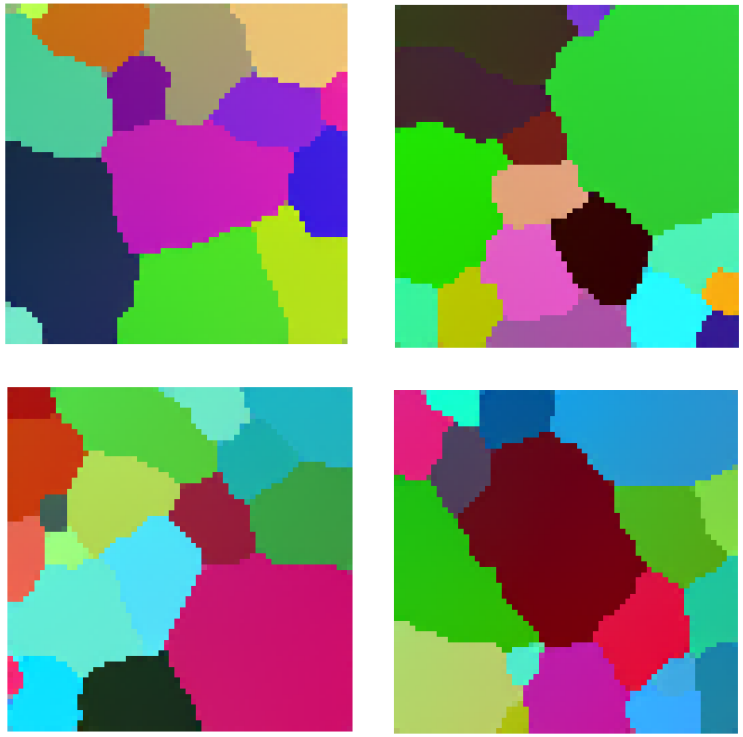
Generated by physical modeling

Generating materials imagery using diffusion models

Diffusion model generates EBSD imagery



Generated by physical modeling



Generated by diffusion models

Non-asymptotic complexity of generation

Theorem (Li, Wei, Chen, Chi, ICLR 2024)

Under mild data assumptions, suppose we are given **perfect score estimates:** $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$ *for all* t.

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

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{T} \qquad \textit{up to log factor}$$

- *For the stochastic sampler (DDPM-type),*

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{\sqrt{T}} \qquad \textit{up to log factor}$$

Non-asymptotic complexity of generation

Theorem (Li, Wei, Chen, Chi, ICLR 2024)

Under mild data assumptions, suppose we are given **perfect score estimates:** $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$ *for all* t.

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

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{T} \qquad \textit{up to log factor}$$

- *For the stochastic sampler (DDPM-type),*

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{\sqrt{T}} \qquad \textit{up to log factor}$$

- first polynomial-time bounds for *plain* probability flow ODE

Non-asymptotic complexity of generation

Theorem (Li, Wei, Chen, Chi, ICLR 2024)

Under mild data assumptions, suppose we are given **perfect score estimates:** $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$ *for all* t.

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

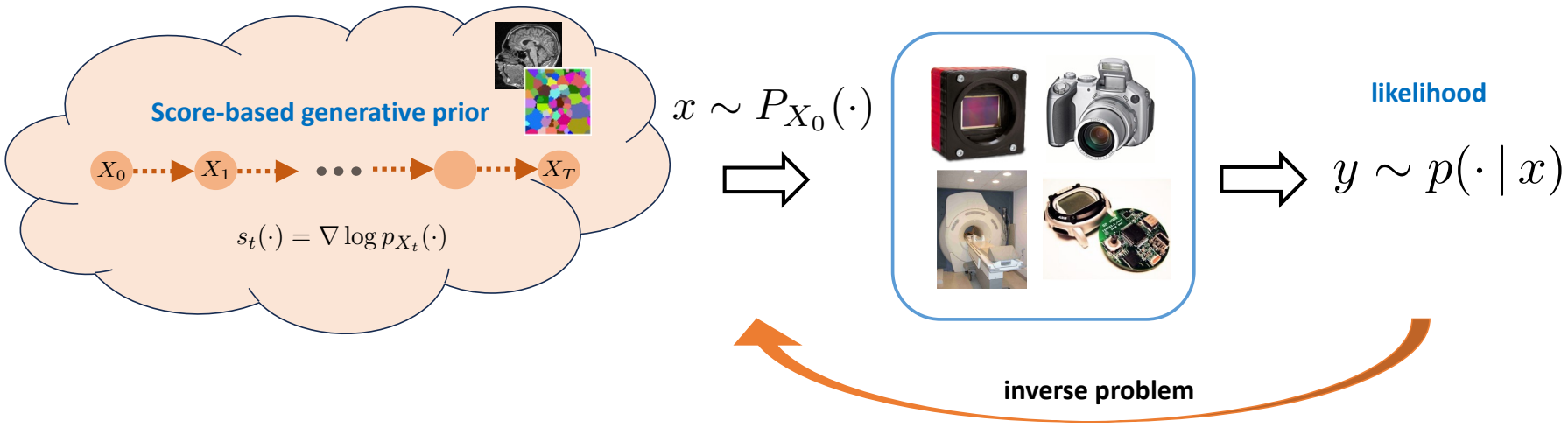
$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{T} \qquad \textit{up to log factor}$$

- *For the stochastic sampler (DDPM-type),*

$$\mathsf{TV}\big(p_{X_1}, p_{Y_1}\big) \lesssim \frac{d^2}{\sqrt{T}} \qquad \textit{up to log factor}$$

- first polynomial-time bounds for *plain* probability flow ODE
- Similar rates extend in the presence of score estimation errors.

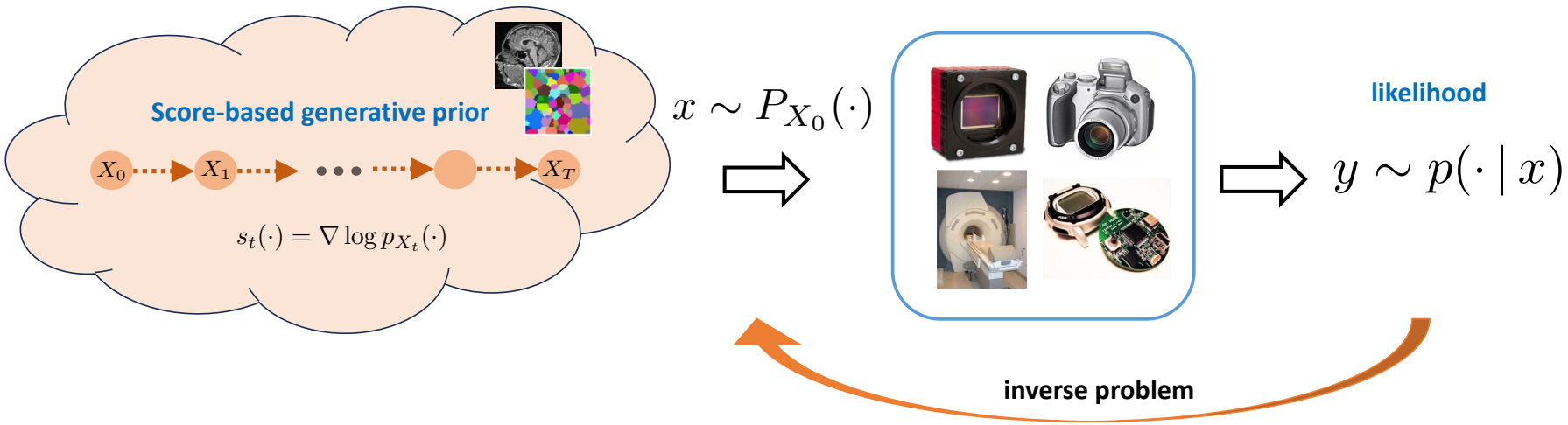
Score-based diffusion model for inverse problems



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 diffusion model for inverse problems



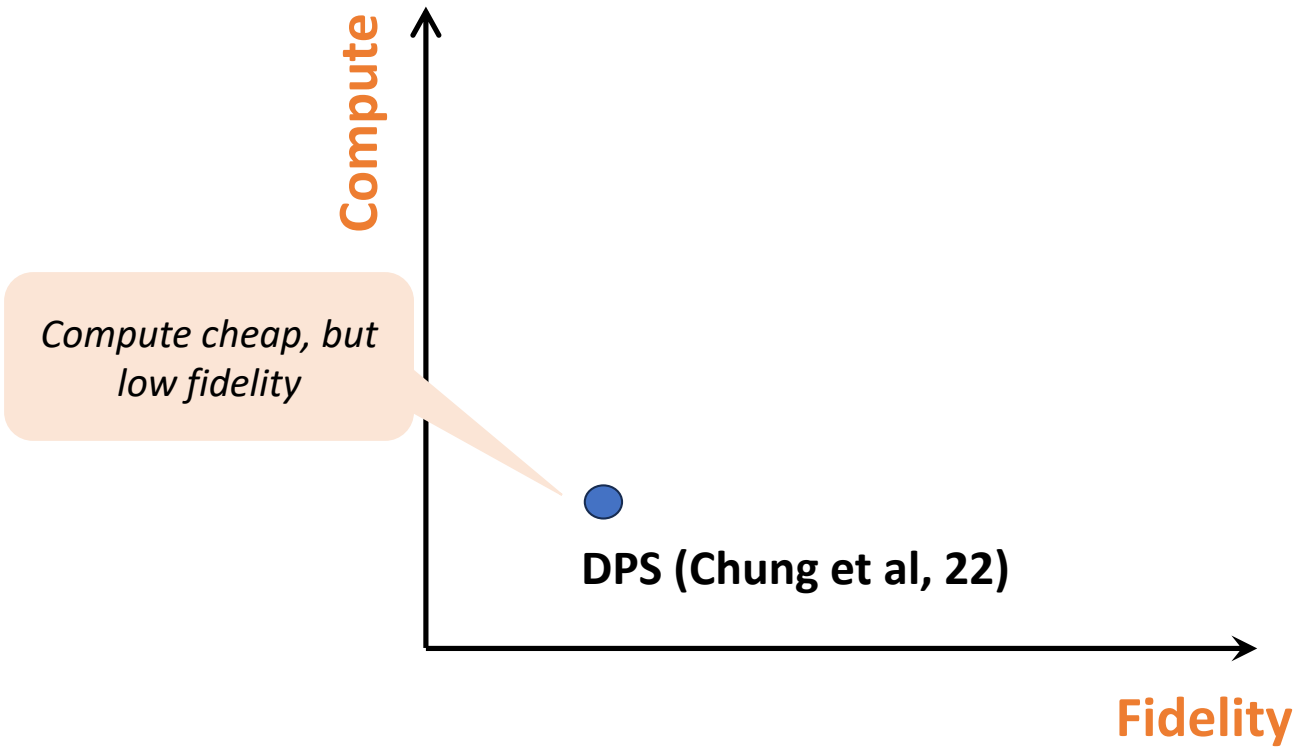
Posterior sampling: sample from

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

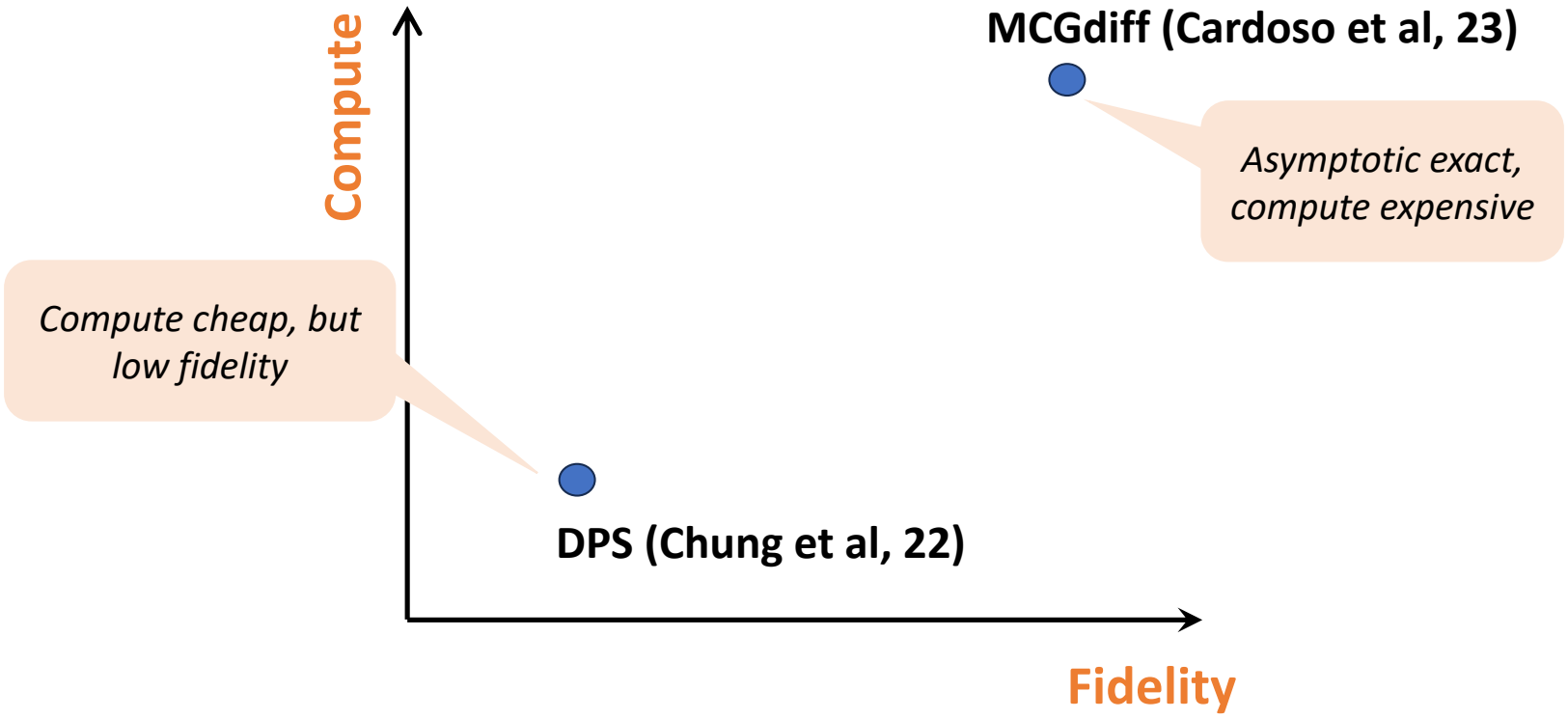
Score-based implicit prior: the data prior $p(\cdot)$ is accessed through its *unconditional* score functions $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$.

Towards provably efficient and accurate inversion

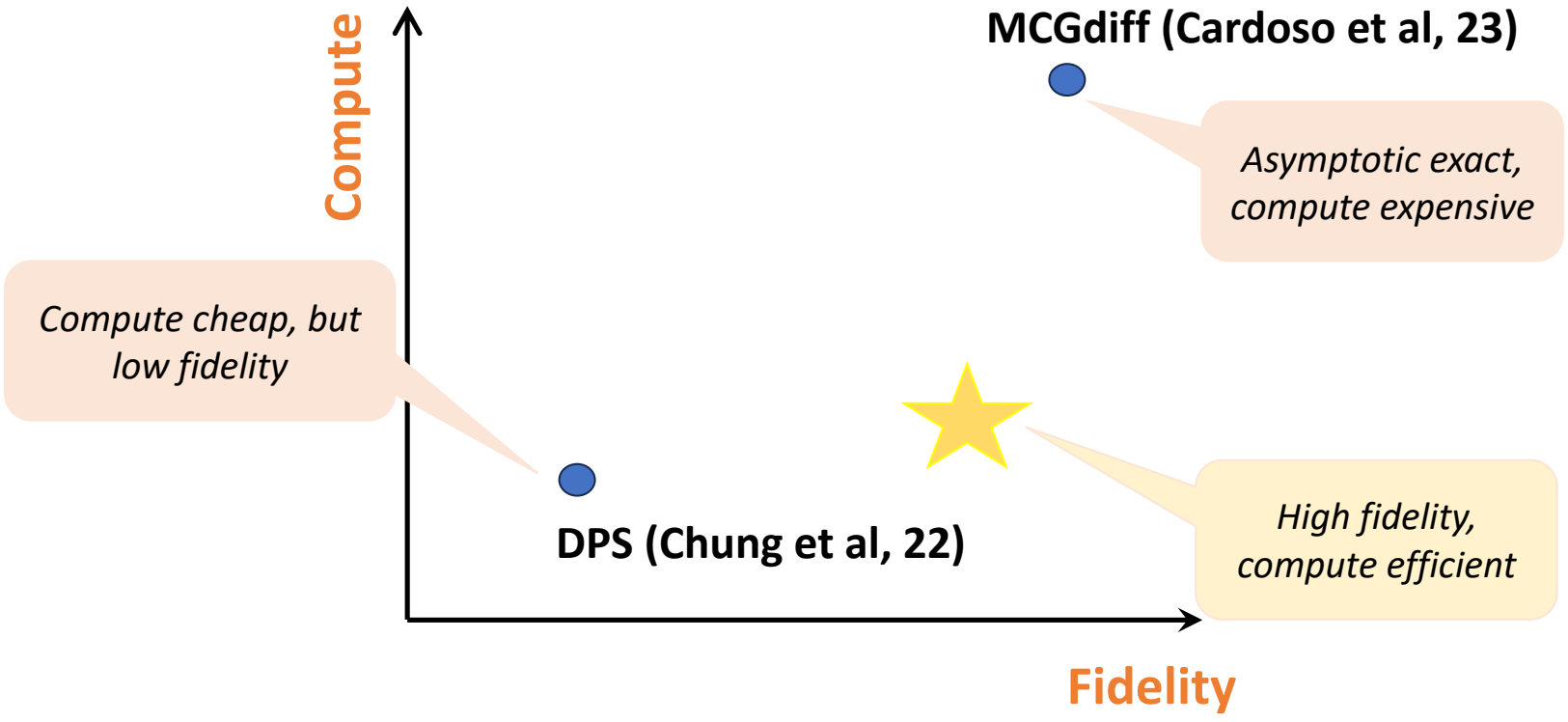
Towards provably efficient and accurate inversion



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)

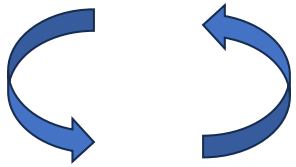
Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

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

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)

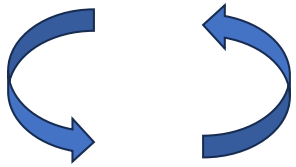
Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

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

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

Proximal consistency sampler:

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



Diffusion denoising sampler:

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

✓ Readily implementable by, e.g., MALA

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

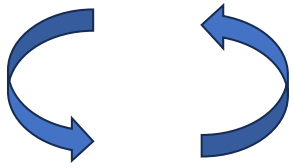
Inspired by (Bouman and Buzzard, 2023; Vono et al., 2019; Lee et al., 2021)

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

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:

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

Readily implementable by, e.g., MALA

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:

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

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:

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

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 $s_t(\cdot)$.
- The resulting update rules are similar to, but not the same as, the ones used for generation.

Our theory

Theorem (Xu and Chi, 2024)

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

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

where $\epsilon \sim \mathcal{N}(0, I_d)$ *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*

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

where $\epsilon \sim \mathcal{N}(0, I_d)$ *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*

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

where $\epsilon \sim \mathcal{N}(0, I_d)$ *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*

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

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

Numerical experiments

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

DPnP recovers the fine-grained details more faithfully.

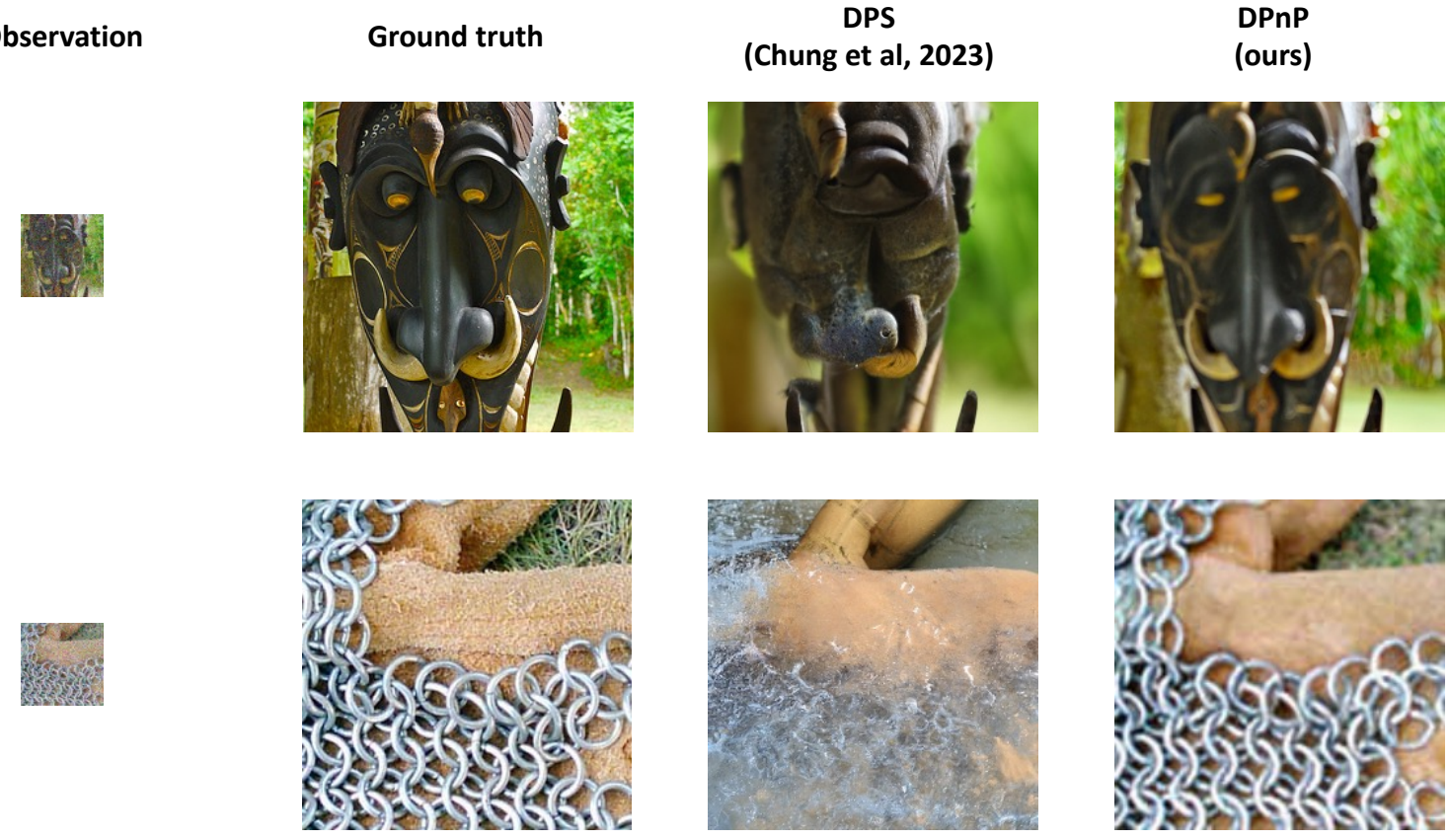
Numerical experiments

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

DPnP recovers the fine-grained details more faithfully.

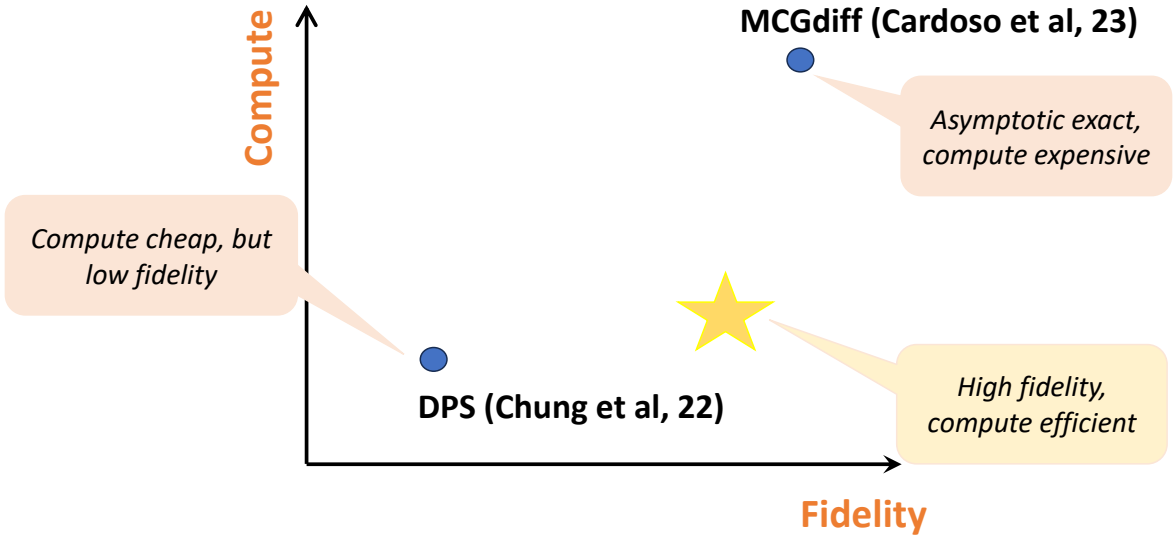
Numerical experiments

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



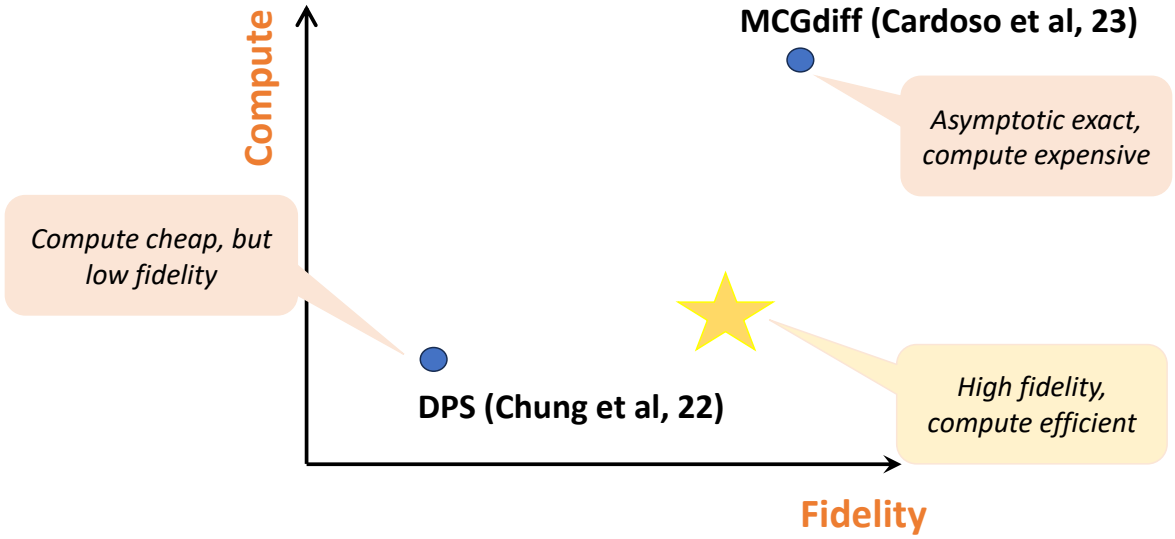
DPnP recovers the fine-grained details more faithfully.

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.

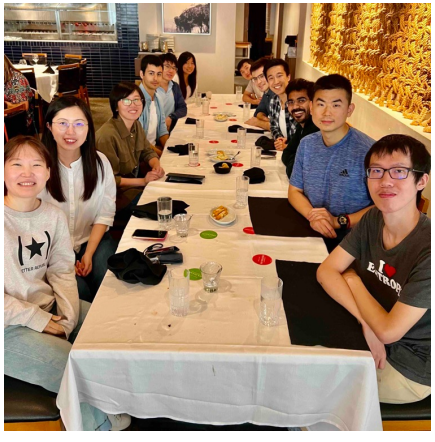
Future directions:

- 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!

- Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent, *Journal of Machine Learning Research*, 2021.
- The Power of Preconditioning in Overparameterized Low-Rank Matrix Sensing, short version at ICML 2023.
- Towards Faster Non-Asymptotic Convergence for Diffusion-Based Generative Models, arXiv: 2306.09251, short version at ICLR 2024.
- 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/