Diffusion Model Journal Club

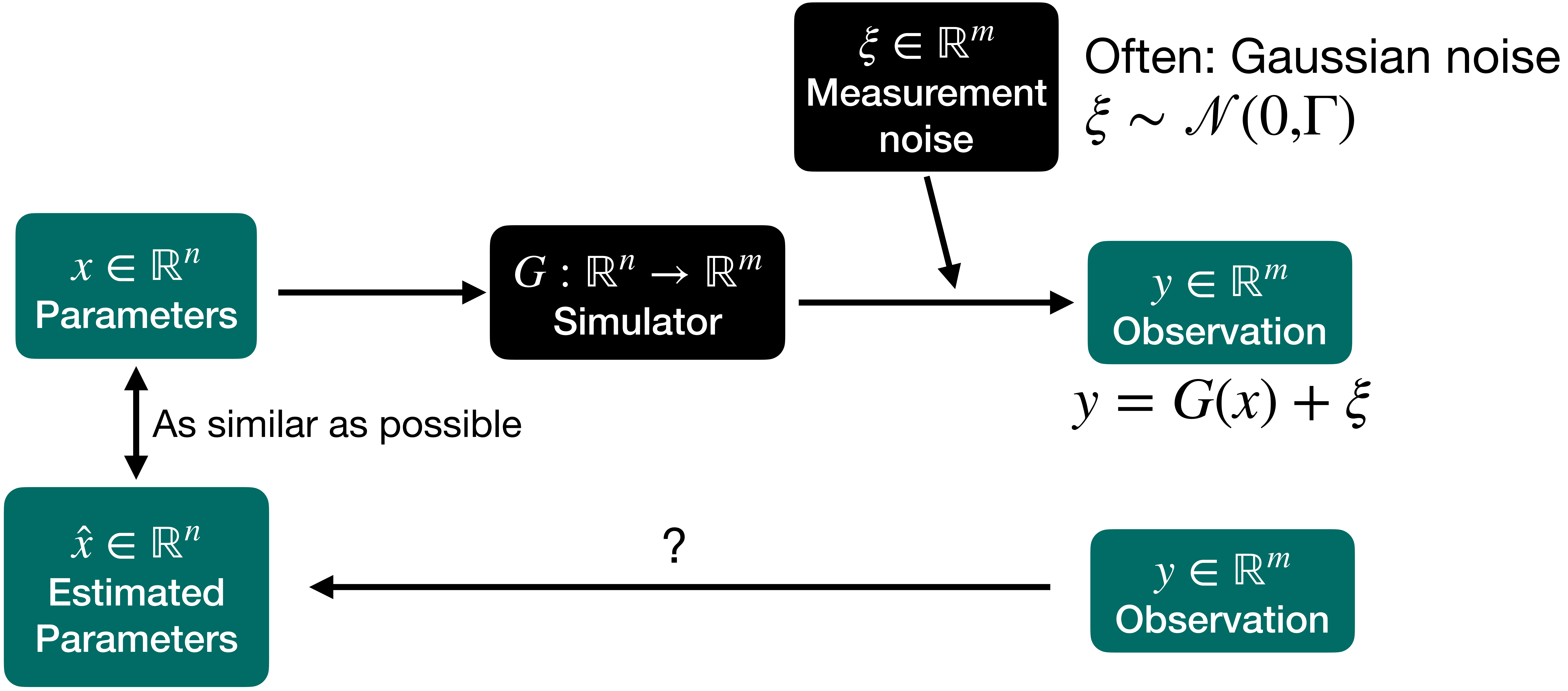
Ensemble Kalman Diffusion Guidance: A Derivative-free Method for Inverse Problems

Annalena Kofler, 30.10.2024

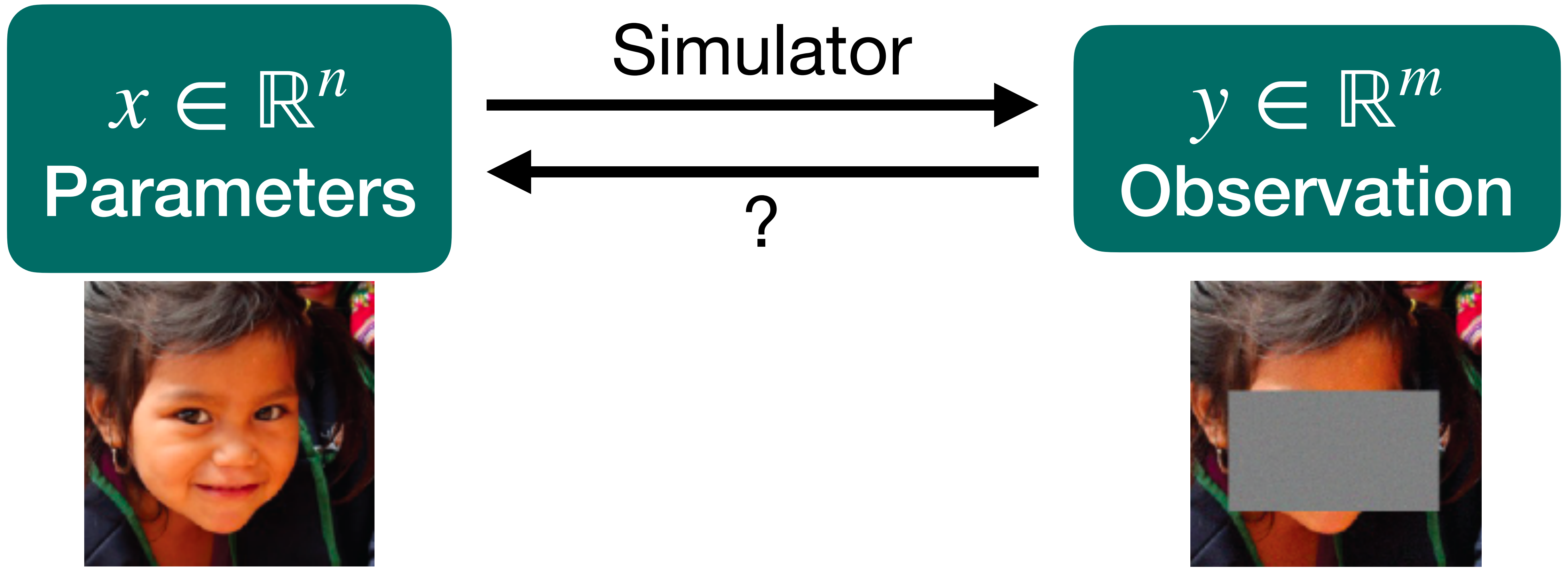
Overview

- What are inverse problems? How to solve them?
- Diffusion Notation
- What is diffusion guidance?
- Why do we need derivative-free optimization?
- Interpreting guidance as prediction-correction

What are inverse problems?



What are inverse problems?



Examples:

- Image inverse problems:
  - Inpainting:
    $x =$ complete image,
    $y =$ image with box mask
  - Superresolution:
    $x =$ high resolution image, $y =$ low resolution image
- Scientific applications:
  - Navier-Stokes: $x =$ Initial vorticity field, $y =$ later noisy, sparse observation
  - Black Hole Imaging: $x =$ black hole image,
    $y =$ noisy telescope visibility measurement

How to solve inverse problems?

$x \in \mathbb{R}^n$ Parameters $\xrightarrow{\text{Simulator}}$ $y \in \mathbb{R}^m$ Observation; $y \in \mathbb{R}^m$ Observation $\xrightarrow{?}$ $x \in \mathbb{R}^n$ Parameters

- Rephrase in terms of probability distributions:

  Solving inverse problem = finding x that has high posterior value $p(x \mid y)$

- Bayes' theorem:

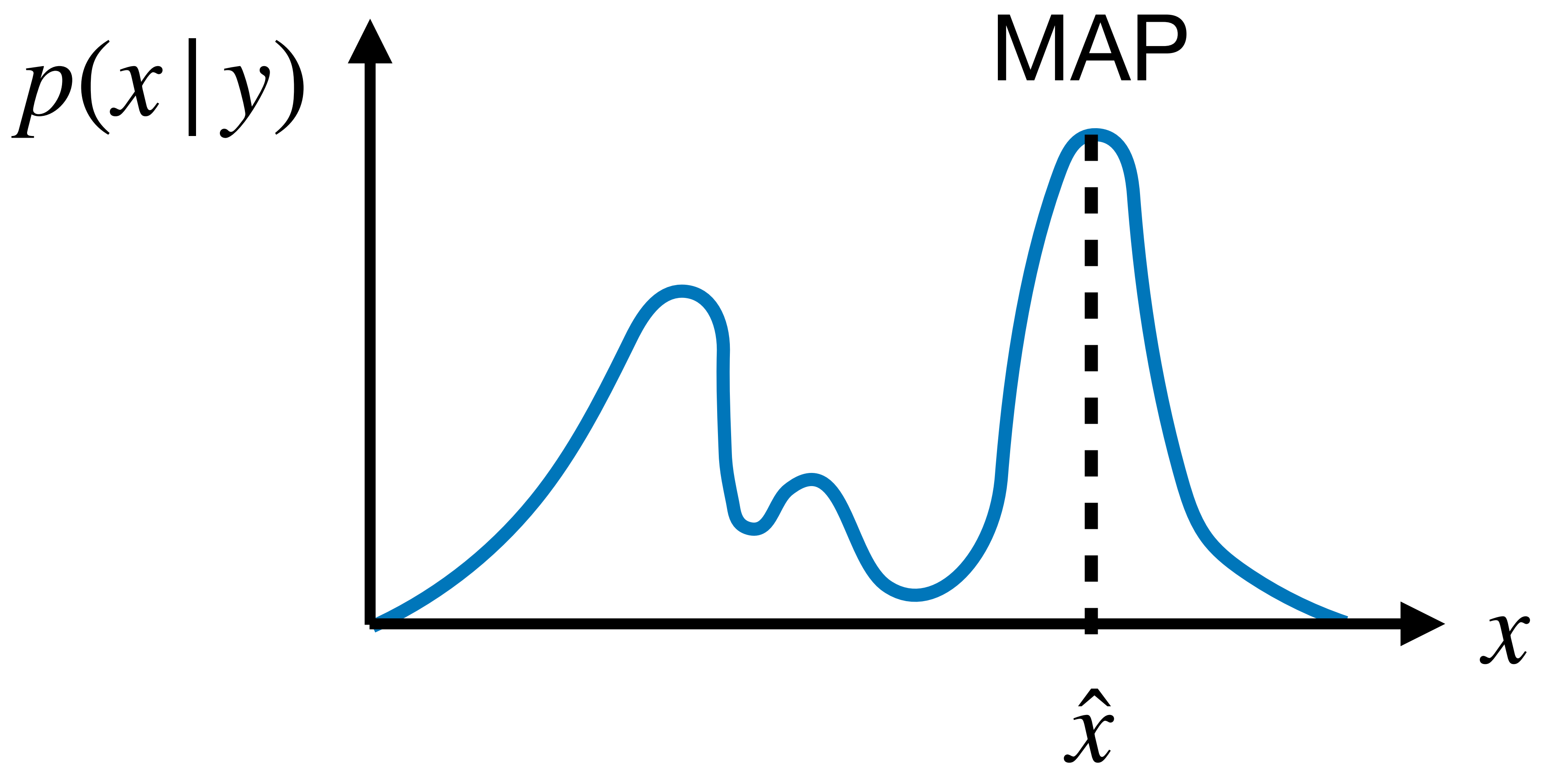
$$p(x \mid y) = \frac{p(y \mid x)}{p(y)} p(x) \propto p(y \mid x) p(y)$$

Prior $p(x)$		Likelihood $p(y \mid x)$		Posterior $p(x \mid y)$
$x \in \mathbb{R}^n$ Parameters	Simulator →	$y \in \mathbb{R}^m$ Observation	? →	$\hat{x} \in \mathbb{R}^n$ Estimated Parameters
Pre-trained diffusion model				

Specific case of this paper: MAP

- Here: simulator = black box
  → sample from likelihood $p(y \mid x)$, but no density evaluation
  → goal: find maximum-a-posteriori estimate $\hat{x}$ for $p(x \mid y)$

$$\hat{x} = \arg\max_x p(y \mid x)\ p(y)$$



Follows Karras 2022 paper

Diffusion Notation

- Capture prior $p(x)$ by diffusion

$t = 0$

data $x_0 \sim p_{\text{data}}$

Forward process
= noising

Reverse process
= denoising

$t = T$

Noise $x_T \sim \mathcal{N}(0, \sigma^2(T)\mathbf{I})$

- Reverse process: **probability flow ODE** / reverse-time stochastic process

$$\mathrm{d}x_t = -\dot{\sigma}(t)\sigma(t)\,\nabla_{x_t}\log p_t(x_t)\,\mathrm{d}t$$

Score function

→ Assume already trained diffusion model

What is diffusion guidance?

- Guidance = conditional inverse process
  $\rightarrow$ solve inverse problems

- Diffusion guidance for inverse problems
  $\rightarrow$ conditional reverse diffusion model targeting posterior $p(x \mid y)$

$$\mathrm{d}x_t = -\dot{\sigma}(t)\sigma(t)\,\nabla_{x_t}\log p_t(x_t \mid y)\ \mathrm{d}t$$

Bayes' theorem

$$= -\dot{\sigma}(t)\sigma(t)\,\underbrace{\nabla_{x_t}\log p_t(x_t)}_{\text{Prior}}\ \mathrm{d}t - \dot{\sigma}(t)\sigma(t)\,\underbrace{\nabla_{x_t}\log p_t(y \mid x_t)}_{\text{Likelihood}}\ \mathrm{d}t$$

$$= -\dot{\sigma}(t)\sigma(t)\underbrace{s_\theta(x_t, t)}_{\text{Diffusion model}}\ \mathrm{d}t - w_t\,\underbrace{\nabla_{x_t}\log \hat{p}_t(y \mid x_t)}_{\text{Problematic!}}\ \mathrm{d}t$$

Why is the gradient of the likelihood a problem?

- Likelihood = probability distribution of forward simulator G

  $$\nabla_{x_t} \log \hat{p}_t(y \mid x_t)$$

- Often:
  - Simulator is very complex and not differentiable
  - Likelihood not analytically accessible
- Solution: Derivative-free optimization

How does derivative-free optimization work?

- Use only output of black-box forward model to define gradient
- E.g. Gaussian smoothing
  - Use Gaussian approximation of function f

$$f_\mu(x) = \frac{1}{\kappa} \int f(x + \mu u) \exp\left(-\frac{1}{2}\|u\|^2\right) \mathrm{d}u$$

  - For which the derivative can be defined as

$$\nabla f_\mu(x) = \frac{1}{\kappa} \int \frac{f(x + \mu u) - f(x - \mu u)}{\mu} \exp\left(-\frac{1}{2}\|u\|^2\right) Bu \ \mathrm{d}u$$

Nesterov and Spokoiny, Random Gradient-free minimization of convex functions, 2015

How does derivative-free optimization work?

- Ensemble Kalman Inversion
    - Robust, easy to implement, comparable accuracy to least squares approaches
    - Ensemble of particles $\{x_i^{(j)}\}_{j=1}^{J}$ with empirical covariance matrix $C_{xx}^{(i)}$
    - Iteratively update ensemble
        1. Prediction step: $\hat{x}_{i+1}^{(j)} = G(x_i^{(j)})$, update ensemble mean and covariance
        2. “Analysis” step: Update ensemble members based on Kalman gain (comparison of mapped ensemble with noise-perturbed data)

Iglesias+, Ensemble Kalman Methods for Inverse Problems, 2013

Interpreting guidance as prediction-correction

- Discretization scheme: $t_N = 0 \rightarrow t_{N-1} \rightarrow \ldots \rightarrow t_1 \rightarrow t_0 = T$

  data $x_N \sim p_{\text{data}}$

  Noise $x_0 \sim \mathcal{N}(0, \sigma^2(T)\mathbf{I})$

- Prior prediction step
  - One step along unconditional ODE trajectory via numerical integration

$$x_i' = x_i - \dot{\sigma}(t_i)\sigma(t_i)s_\theta(x_i, t_i)(t_{i+1} - t_i)$$

- Log likelihood evaluation at x_i $\quad \log \hat{p}(y \mid x_i) \approx \log p(y \mid x_i)$

- Guidance correction step

$$x_{i+1} = \arg\min_{x_{i+1}} \frac{1}{2w_i} \|x_{i+1} - x_i'\|_2^2 - \log \hat{p}(y \mid x_{i+1})$$

Generic guidance algorithm

Algorithm 1 Generic Guidance-based Method (ODE version)

Require: $G, \boldsymbol{y}, s_\theta, \{t_i\}_{i=1}^N, \{w_i\}_{i=1}^N$

sample $\boldsymbol{x}_0 \sim \mathcal{N}(0, \sigma^2(t_0)\boldsymbol{I})$

for $i \in \{0, \ldots, N-1\}$ **do**

$\quad \boldsymbol{x}_i' \leftarrow \boldsymbol{x}_i - \dot{\sigma}(t_i)\sigma(t_i)s_\theta(\boldsymbol{x}_i, t_i)(t_{i+1} - t_i)$ ▷ Prior prediction step

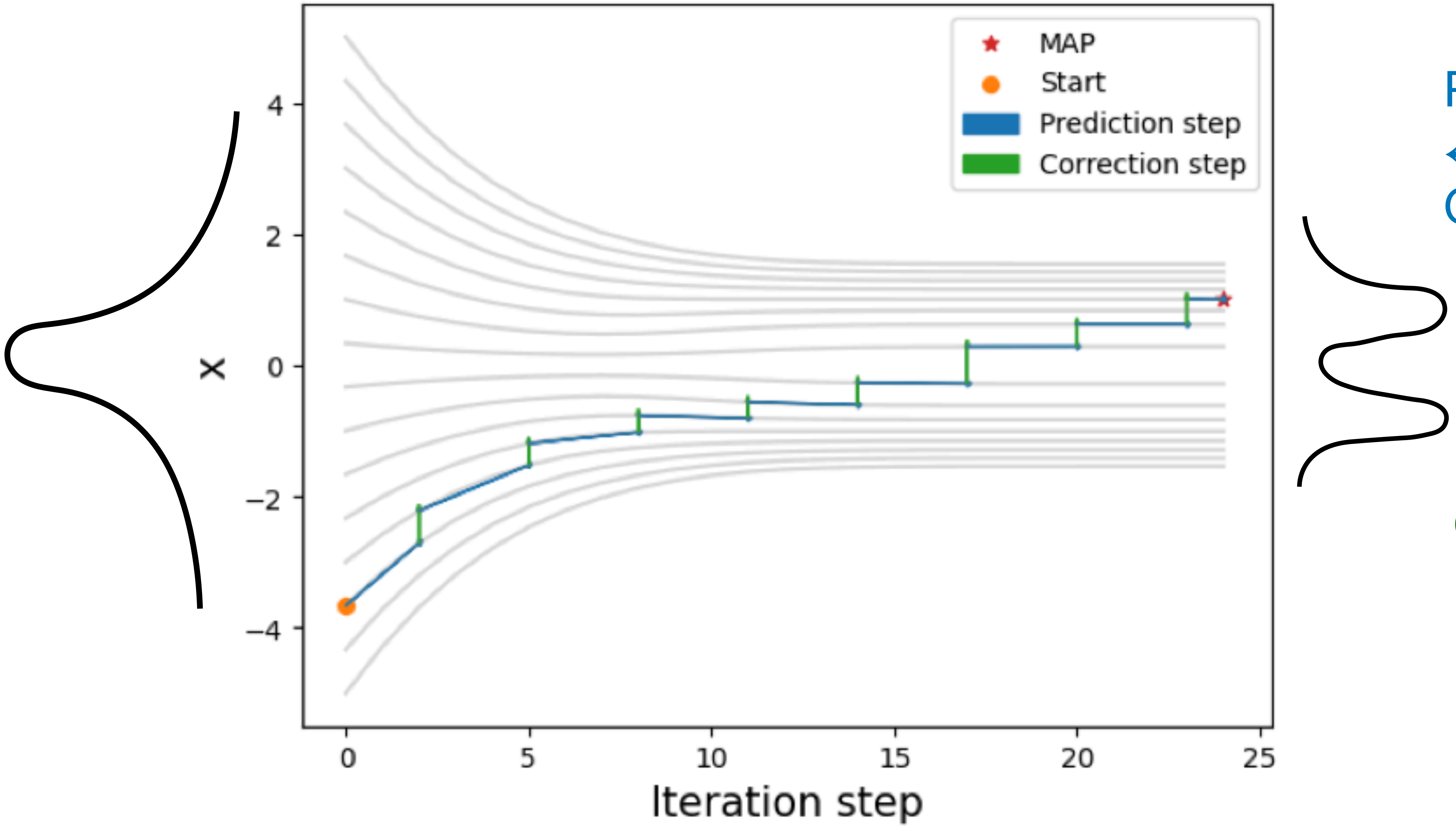
$\quad \log \hat{p}(\boldsymbol{y}|\boldsymbol{x}_t) \approx \log p(\boldsymbol{y}|\boldsymbol{x}_t)$ ▷ Log-likelihood estimation

$\quad \boldsymbol{x}_{i+1} \leftarrow \arg\min_{\boldsymbol{x}_{i+1}} \frac{\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i'\|_2^2}{2w_i} - \log \hat{p}(\boldsymbol{y}|\boldsymbol{x}_{i+1})$ ▷ Guidance correction step

end for

return $\boldsymbol{x}_N$

Illustration of prediction-correction interpretation



Prediction step:
↔ move along unconditional ODE trajectory

Correction step:
↕ Locally find best x_{i+1} that maximizes the likelihood

Simplifying guidance correction

- Guidance correction step $x_{i+1} = \arg\min_{x_{i+1}} \frac{1}{2w_i} \|x_{i+1} - x_i'\|_2^2 - \log \hat{p}(y \mid x_{i+1})$
- First-order Taylor expansion

$$\log \hat{p}(y \mid x_{i+1}) = \log \hat{p}(y \mid x_i') + \nabla^T \log \hat{p}(y \mid x_i')(x_{i+1} - x_i') + \mathcal{O}(\Delta x_i^2)$$

$$x_{i+1} \approx \arg\min_{x_{i+1}} \frac{1}{2w_i} \|x_{i+1} - x_i'\|_2^2 - \log \hat{p}(y \mid x_i) - \nabla^T \log \hat{p}(y \mid x_i')(x_{i+1} - x_i')$$

$$0 = \frac{\partial}{\partial x_{i+1}} \left(\frac{1}{2w_i} \|x_{i+1} - x_i'\|_2^2 - \log \hat{p}(y \mid x_i) - \nabla^T \log \hat{p}(y \mid x_i')(x_{i+1} - x_i') \right)$$

$$0 = \frac{1}{w_i}(x_{i+1} - x_i')_2 - \nabla^T \log \hat{p}(y \mid x_i') \longrightarrow x_{i+1} = x_i' + w_i \nabla \log \hat{p}(y \mid x_i')$$

Simplifying guidance correction

- Guidance correction step $$x_{i+1} = \arg\min_{x_{i+1}} \frac{1}{2w_i} \|x_{i+1} - x_i'\|_2^2 - \log \hat{p}(y \mid x_{i+1})$$
- First-order Taylor expansion

$$x_{i+1} = x_i' + w_i \, \nabla \log \hat{p}(y \mid x_i')$$

- Problem: still dependent on gradient of log-likelihood

Ensemble Kalman Diffusion Guidance

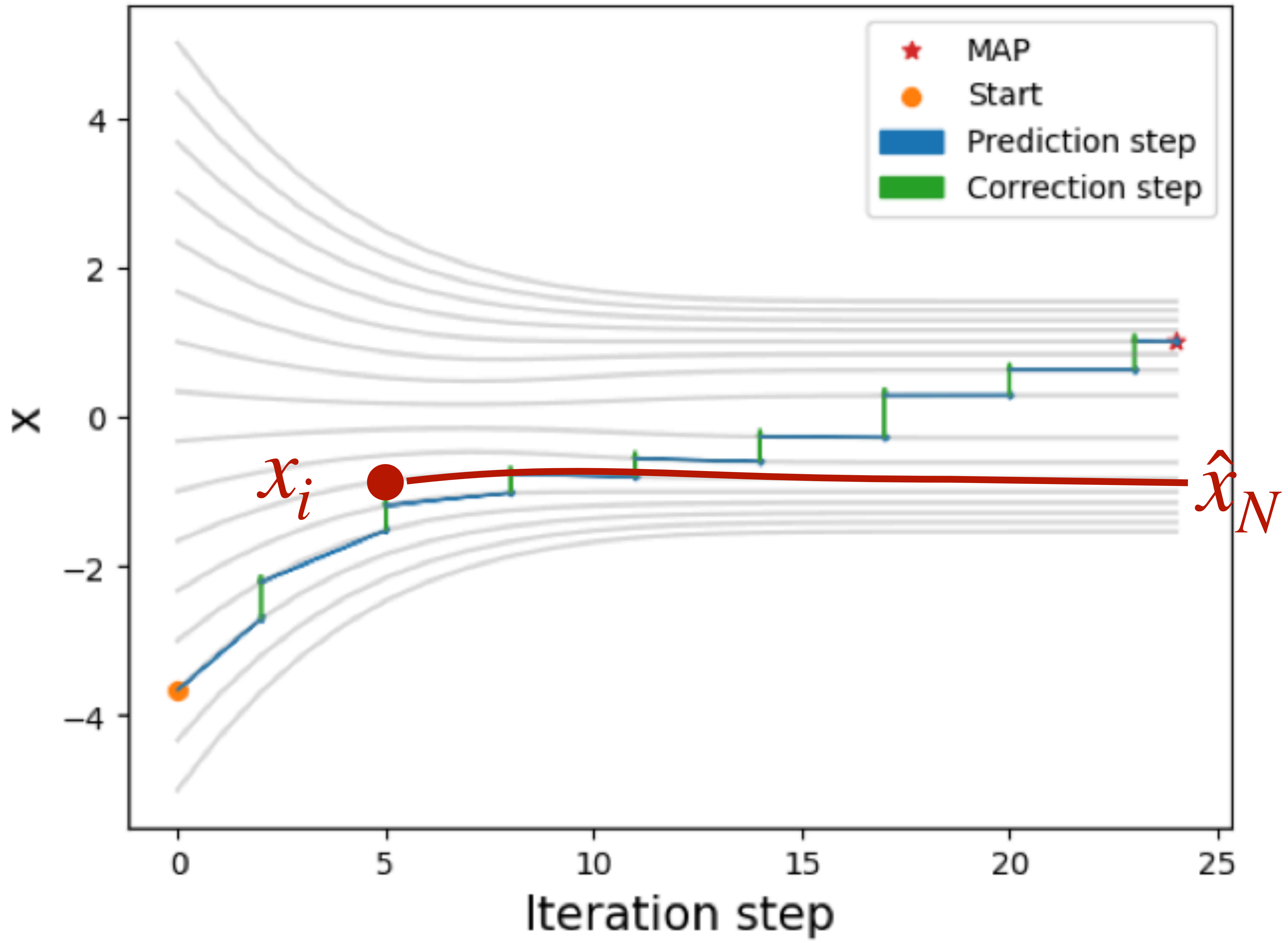
- Goal: Write correction step in derivative-free manner
- Step 1: How to estimate the likelihood?

Factorize likelihood & MC approximation:

$$p(y\,|\,x_i) = \int p(y\,|\,x_N)p(X_N\,|\,x_i)\ \mathrm{d}x_N$$

$$= \mathbb{E}_{x_N \sim p(x_N|x_i)}[p(y\,|\,x_N)] \approx p(y\,|\,\hat{x}_N)$$

$\rightarrow \hat{x}_N$ = run ODE solver $\phi(x_i)$



EnKG: Define a derivative-free correction step?

- Step 2: How to define a derivative-free correction step?
  - Kalman Ensemble of particles $\{x_i^{(j)}\}_{j=1}^{J}$ with empirical covariance matrix $C_{xx}^{(i)}$ and mean $\bar{x}_i$
  - Replace scalar weight with weighting matrix $w_i C_{xx}^{(i)}$

$$x_{i+1}^{(j)} = x_i^{\prime(j)} + w_i C_{xx}^{(i)} \ \nabla \log \hat{p}(y \,|\, x_i^{\prime(j)})$$

= gradient step preconditioned on $C_{xx}^{(i)}$

  - Valid as gradient because of statistical linearization (gradient corresponds to best linear estimator of gradient mean in large ensemble limit and based on Gaussian approximation)

Schillings and Stuart, Analysis of the Ensemble Kalman Filter for Inverse problems, 2016

EnKG: Which assumptions do we need to make?

- Assumption 1:
  ODE solver ϕ and forward model G have to have bounded 1st and 2nd-order derivatives, i.e. the join Hessian $H_{G\circ\phi}$ is bounded

- Assumption 2:
  Distance between ensemble particles is bounded

- Assumption 3:
  Empirical covariance matrix does not collapse to zero $tr\left(C_{xx}^{(i)}\right) \neq 0$
  unless $C_{xx}^{(i)} = 0$

EnKG: How to approximate the gradient?

- With these assumptions, we can choose $w_i = 1/\mathrm{tr}\left(C_{xx}^{(i)}\right)$
- Approximation of the gradient:

$$C_{xx}^{(i)} \nabla \log \hat{p}\left(y \mid x_i^{')j)}\right) \approx \frac{1}{J}\sum_{k=1}^{J}\left\langle \psi\left(x_i^{'(k)}\right) - \bar{G},\, y - \psi\left(x_i^{'(j)}\right)\right\rangle_{\Gamma}\left(x_i^{'(j)} - \bar{x}_i\right)$$

$$\bar{G} = \frac{1}{J}\sum_{j=1}^{J} G\left(\hat{x}_N^{(j)}\right) = \frac{1}{J}\sum_{j=1}^{J}\psi\left(x_i^{'(j)}\right)$$

EnKG: How does the algorithm work?

Algorithm 2 Our method: Ensemble Kalman Diffusion Guidance (EnKG).

Require: $G, \boldsymbol{y}, s_\theta$, solver $\phi, \{t_i\}_{i=1}^N, \{w_i\}_{i=1}^N, J$

1: **sample** $\boldsymbol{x}_0^{(j)} \sim \mathcal{N}(0, \sigma^2(t_0)\boldsymbol{I}), j = 1, \ldots, J$ ▷ Initialize particles

2: **for** $i \in \{0, \ldots, N-1\}$ **do**

3: $\quad \boldsymbol{x}_i'^{(j)} \leftarrow \boldsymbol{x}_i^{(j)} - \dot{\sigma}(t_i)\sigma(t_i)s_\theta\left(\boldsymbol{x}_i^{(j)}, t_i\right)(t_{i+1} - t_i)$ ▷ Prior prediction step

4: $\quad \hat{\boldsymbol{x}}_N^{(j)} \leftarrow \phi\left(\boldsymbol{x}_i^{(j)}, t_i\right), j = 1, \ldots, J$

5: $\quad g_i^{(j)} \leftarrow \frac{1}{J}\sum_{k=1}^J \left\langle G(\hat{\boldsymbol{x}}_N^{(k)}) - \bar{G}, \boldsymbol{y} - G(\hat{\boldsymbol{x}}_N^{(j)})\right\rangle_\Gamma \left(\boldsymbol{x}_i^{(j)} - \bar{\boldsymbol{x}}_i\right)$

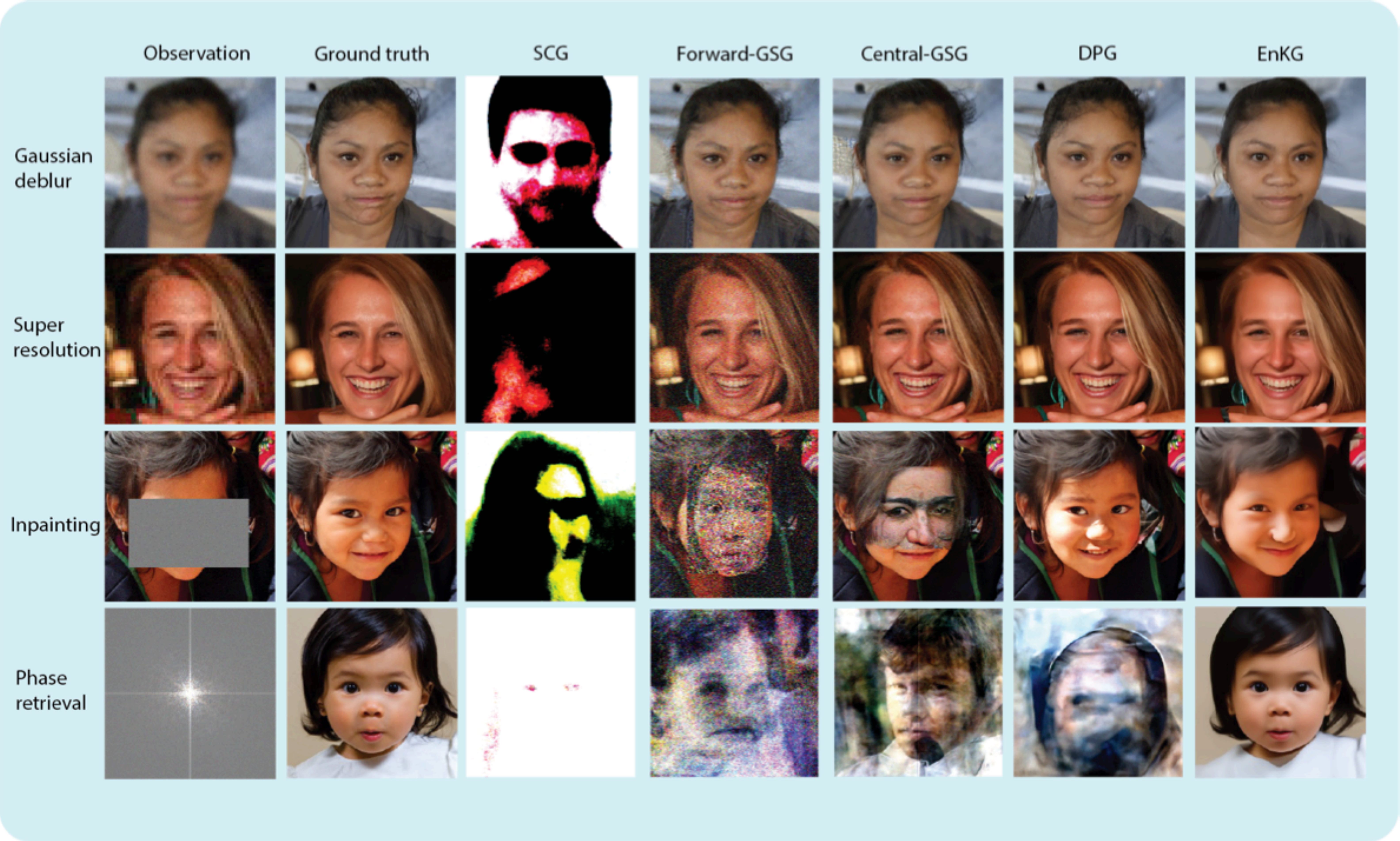
6: $\quad \boldsymbol{x}_{i+1}^{(j)} \leftarrow \boldsymbol{x}_i'^{(j)} + w_i g_i^{(j)}, j = 1, \ldots, J$ ▷ Guidance correction step

7: **end for**

8: **return** $\{\boldsymbol{x}_N^{(j)}\}_{j=1}^J$

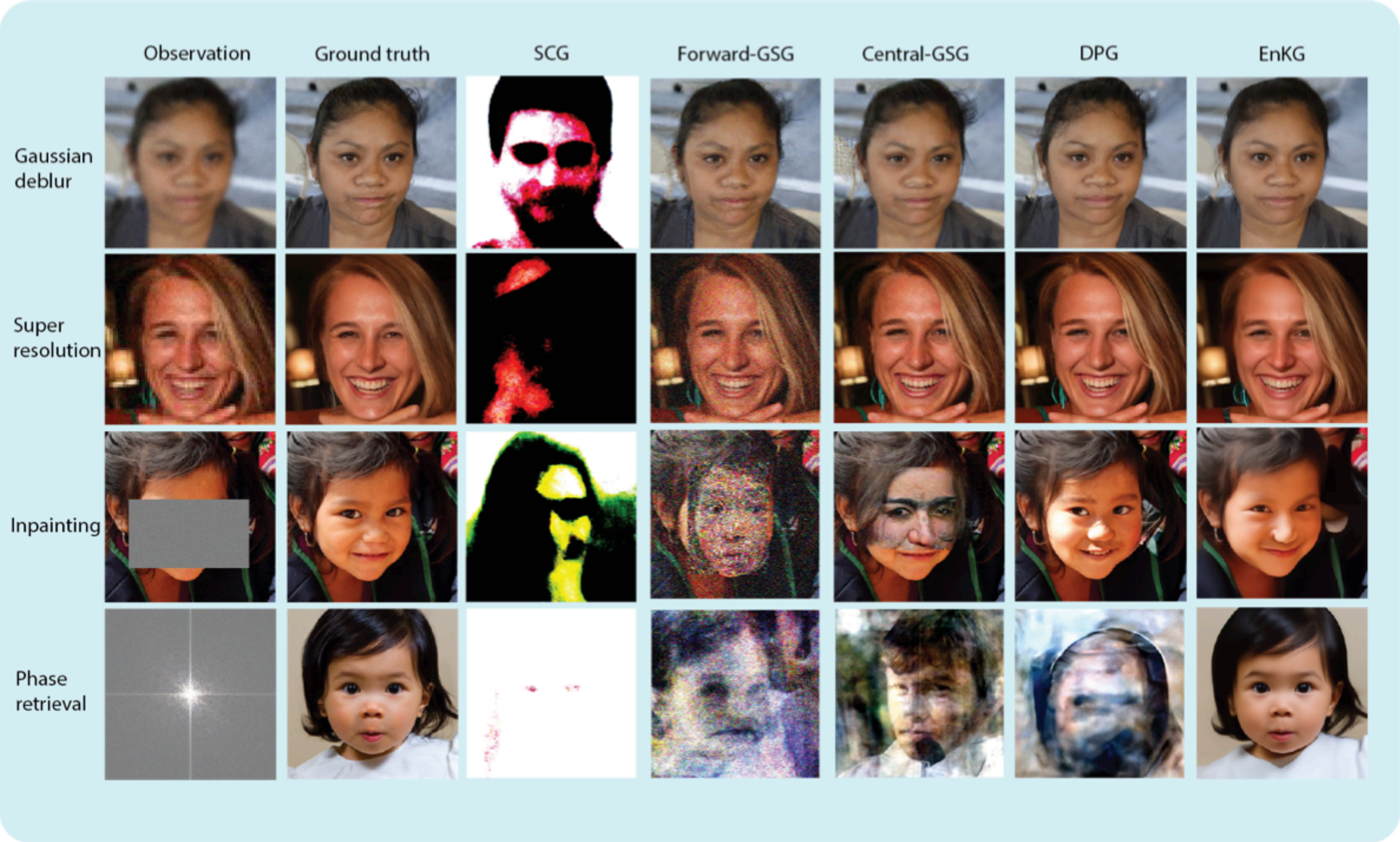
Experiments: Image Inverse Problems

- Inpainting
- Superresolution
- Deblurring of Gaussian noise
- Phase retrieval



Experiments: Image Inverse Problems

- Other baselines:
  - DPS = diffusion posterior sampling
  - SCG = stochastic control guidance
  - Forward-GSG = forward Gaussian smoothed gradient
  - Central-GSG = central GSG
  - DPG = Diffusion policy gradient



Experiments: Image Inverse Problems

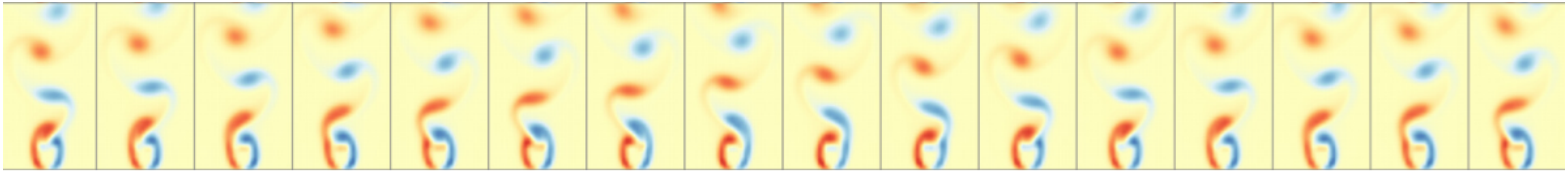
- Numerical Results:

	Inpaint (box)			SR ($\times 4$)			Deblur (Gauss)			Phase retrieval		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Gradient access												
DPS†	**21.77**	**0.767**	0.213	24.90	0.710	0.265	25.46	0.708	0.212	14.14	0.401	0.486
Black-box access												
Forward-GSG	17.82	0.562	0.302	18.08	0.469	0.384	24.43	0.704	0.206	7.88	0.070	0.838
Central-GSG	18.76	0.720	0.229	26.55	0.740	0.169	25.39	0.775	0.180	10.10	0.353	0.691
DPG	20.89	0.752	**0.184**	**28.12**	**0.831**	**0.126**	**26.42**	**0.798**	**0.143**	15.47	0.486	0.495
SCG	4.71	0.302	0.763	4.71	0.302	0.760	4.69	0.300	0.759	4.623	0.294	0.764
EnKG(Ours)	21.70	0.727	0.286	27.17	0.773	0.237	26.13	0.723	0.224	**20.06**	**0.584**	**0.393**

Experiments: Navier-Stokes

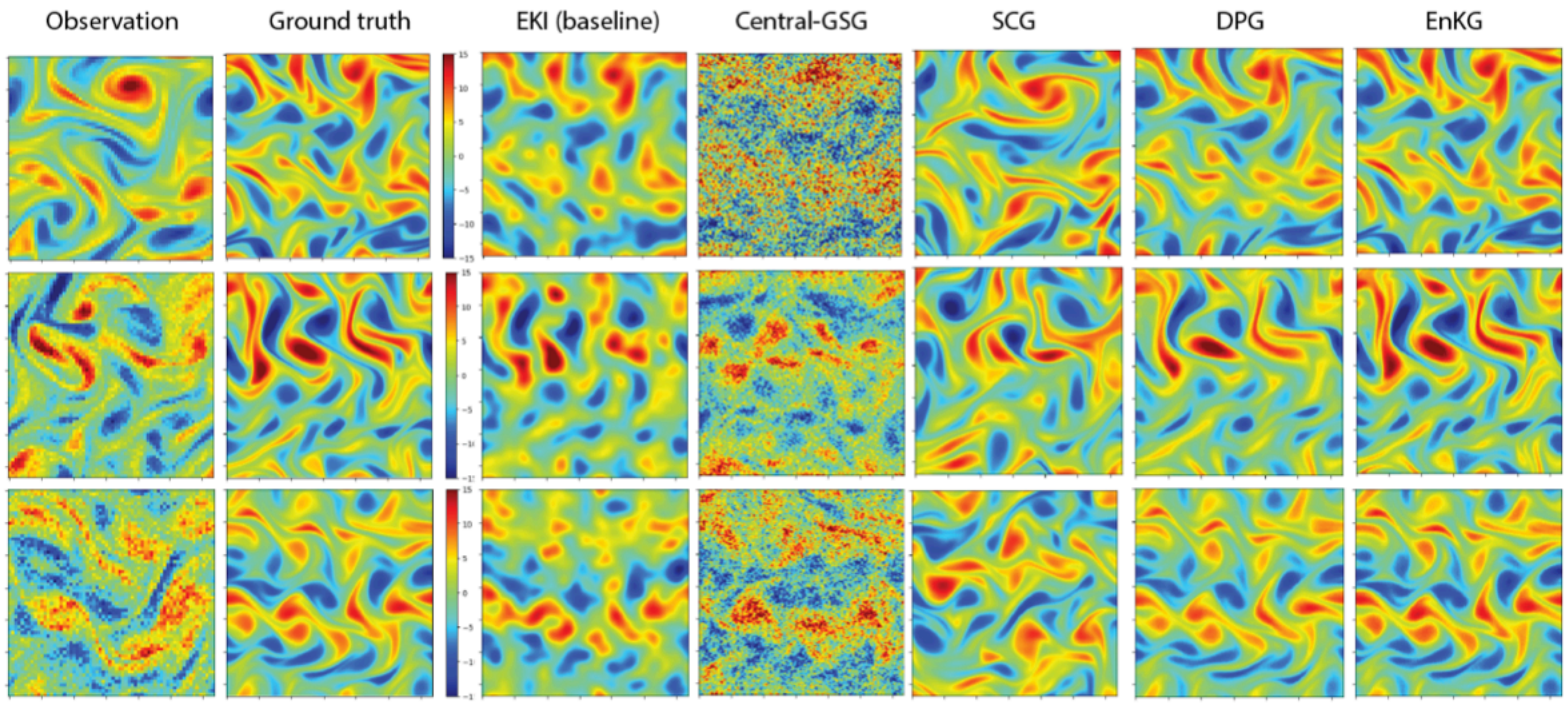
- 2D Navier-Stokes equation of fluid flowing around a torus
  → represent by vorticity field

- Goal: recovar initial vorticity field from noisy, sparse observation at $T = 1$



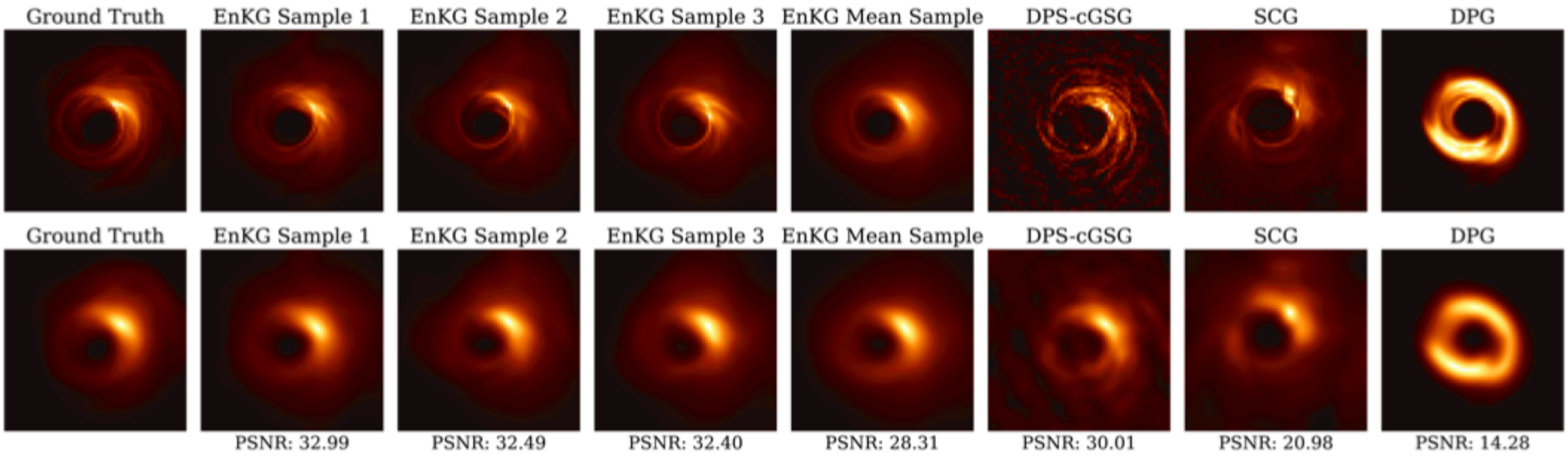
Experiments: Navier-Stokes

- Results



Experiments: Black Hole Imaging

- Empirical Results



Limitations

- Not full posterior, only MAP