

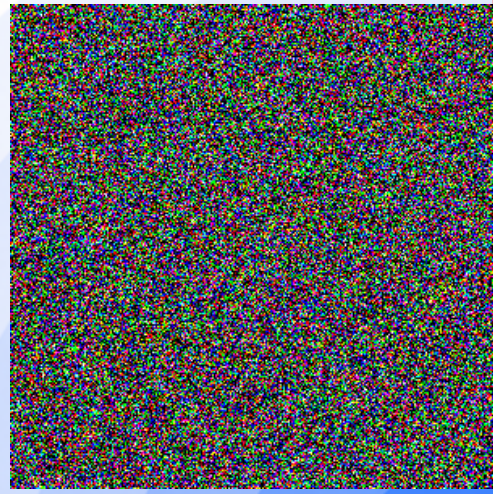
Geometry-Aware Sampling Error of Diffusion Models

Samuel Hurault, Thomas Moreau, Gabriel Peyré



Diffusion models

Noising process

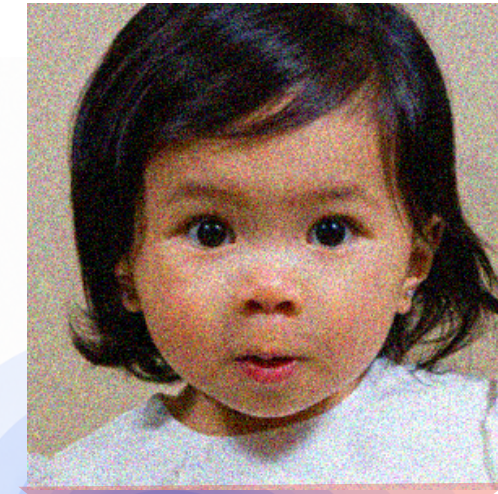
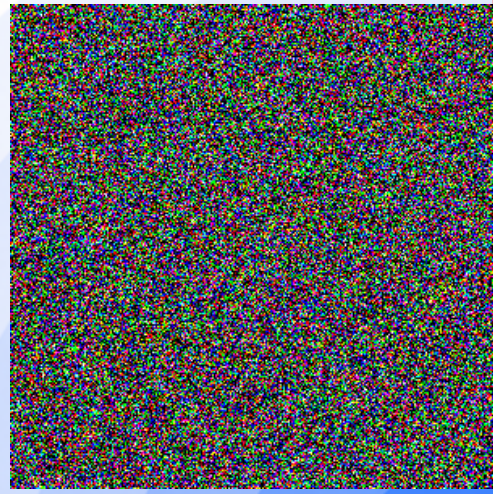


Generative process



Diffusion models

Noising process

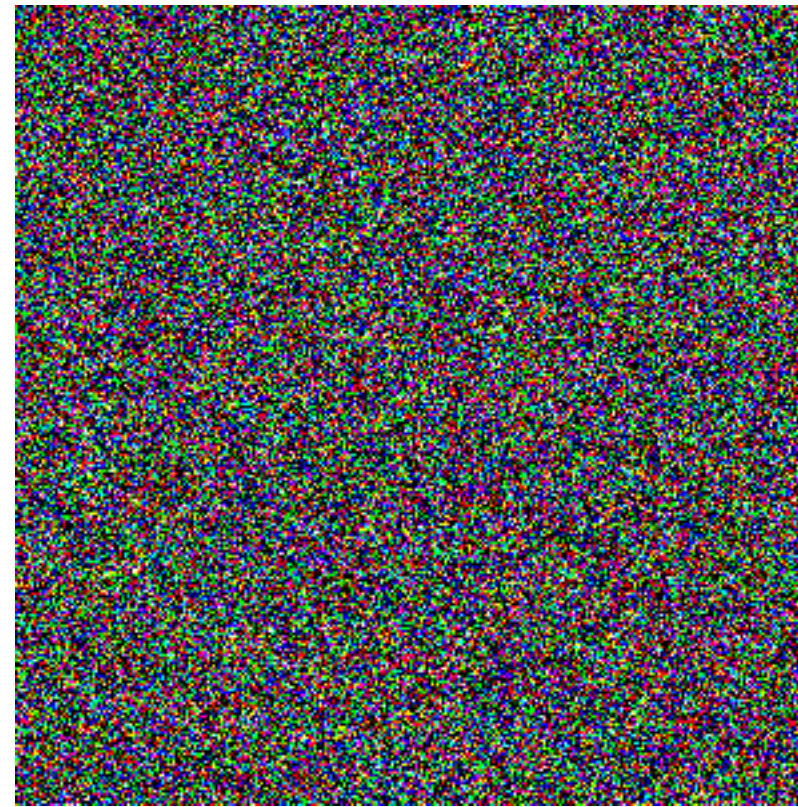


Forward noising process: from data to Gaussian



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

$+$ $\sigma_1 \cdot$

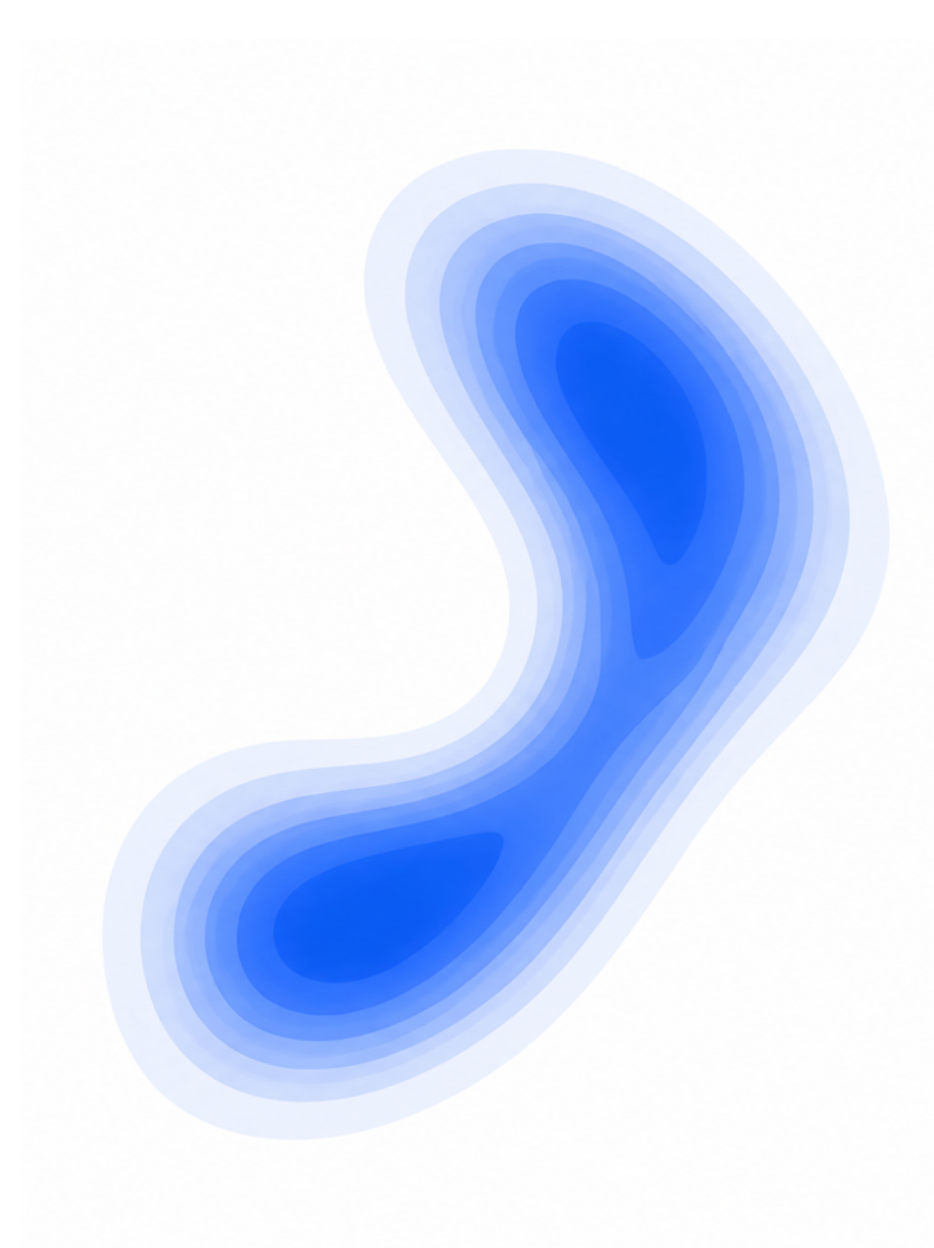


$$w \sim \mathcal{N}(0, \text{Id})$$

$=$

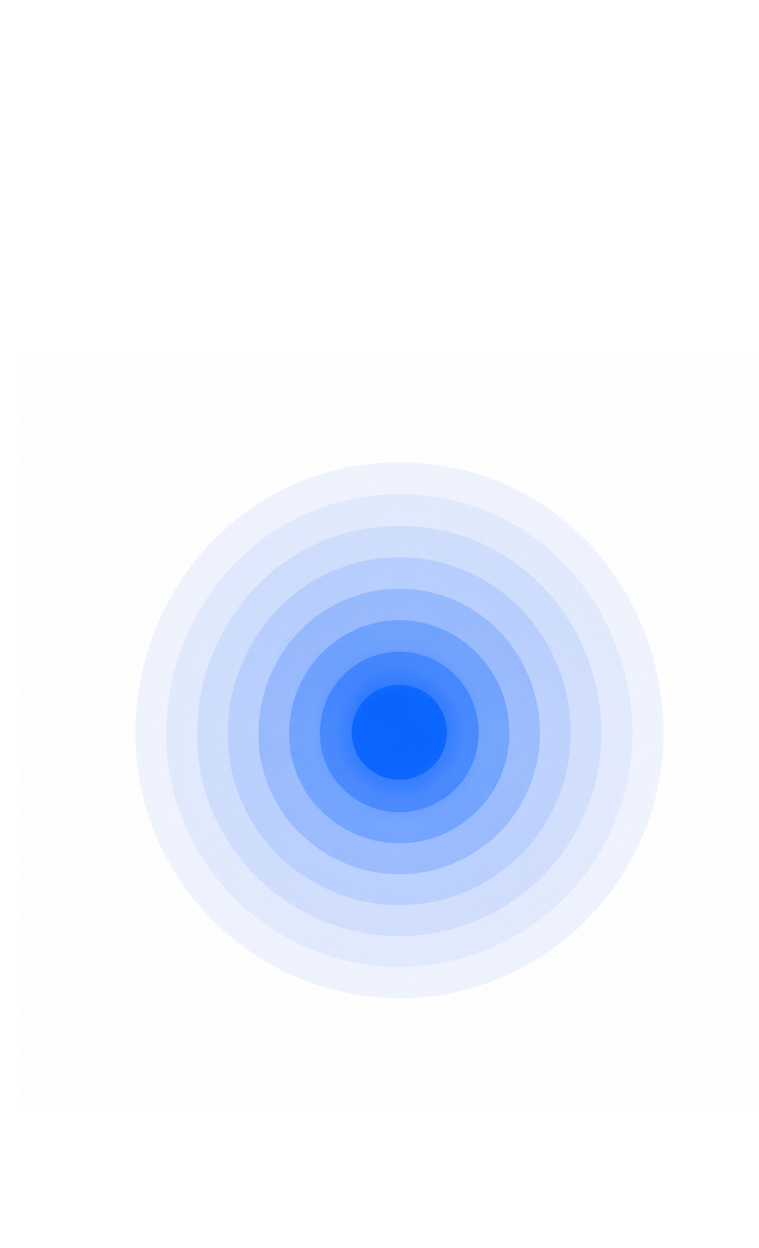


$$y = x + \sigma_1 w$$



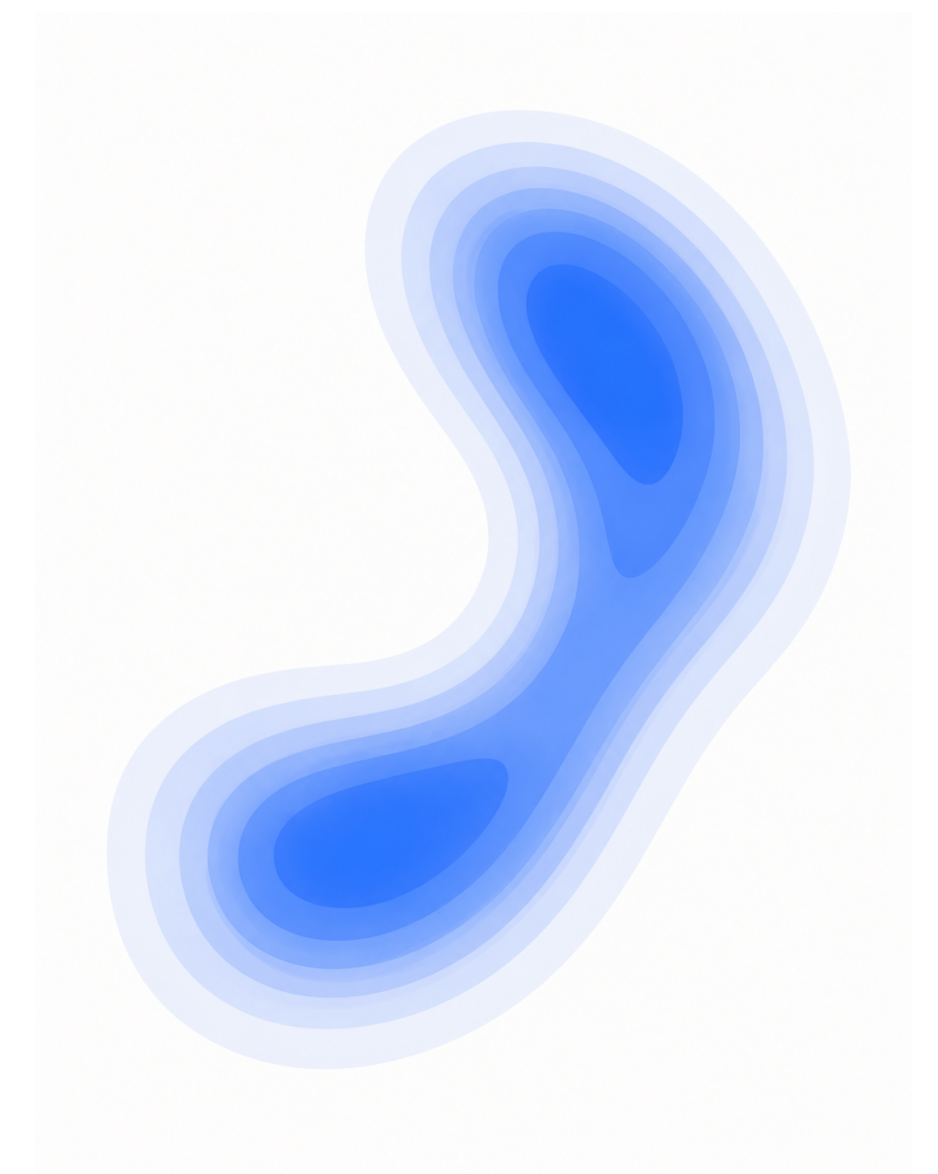
$$p_{\text{data}}$$

$\star$



$$\mathcal{N}(0, \sigma_1^2 \text{Id})$$

$=$



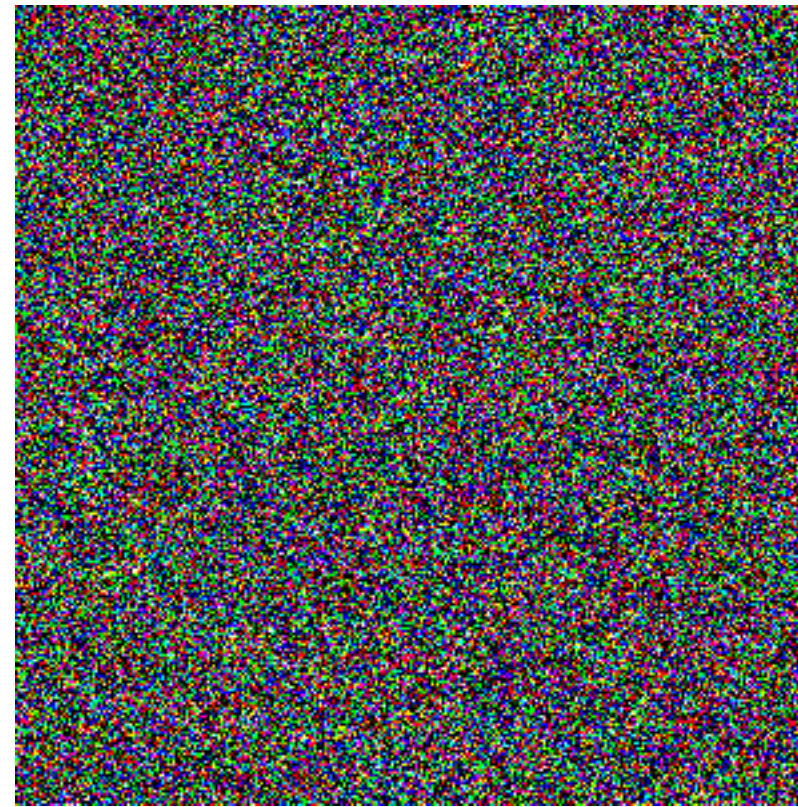
$$p_1 = p_{\text{data}} \star \mathcal{N}(0, \sigma_1^2 \text{Id})$$

Forward noising process: from data to Gaussian



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

$+$ $\sigma_2 \cdot$

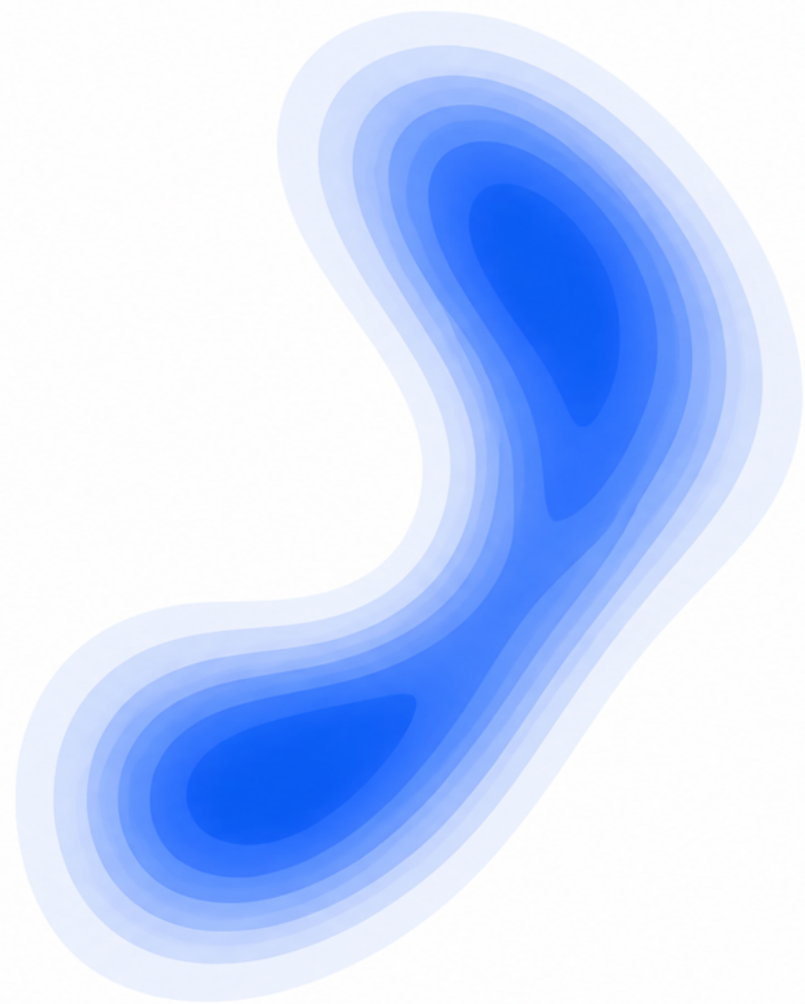


$$w \sim \mathcal{N}(0, \text{Id})$$

$=$

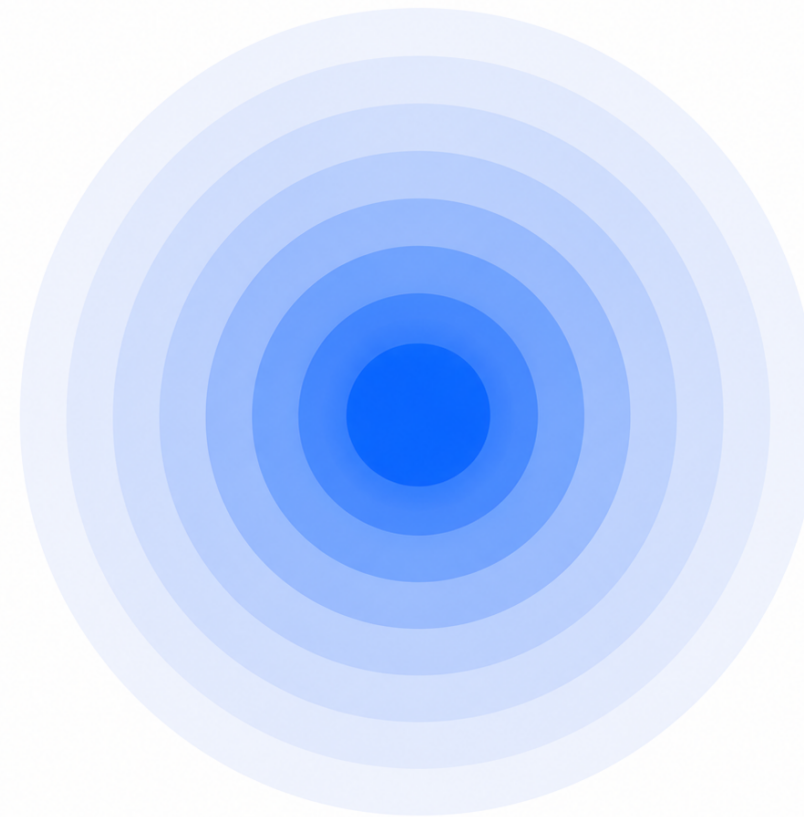


$$y = x + \sigma_2 w$$



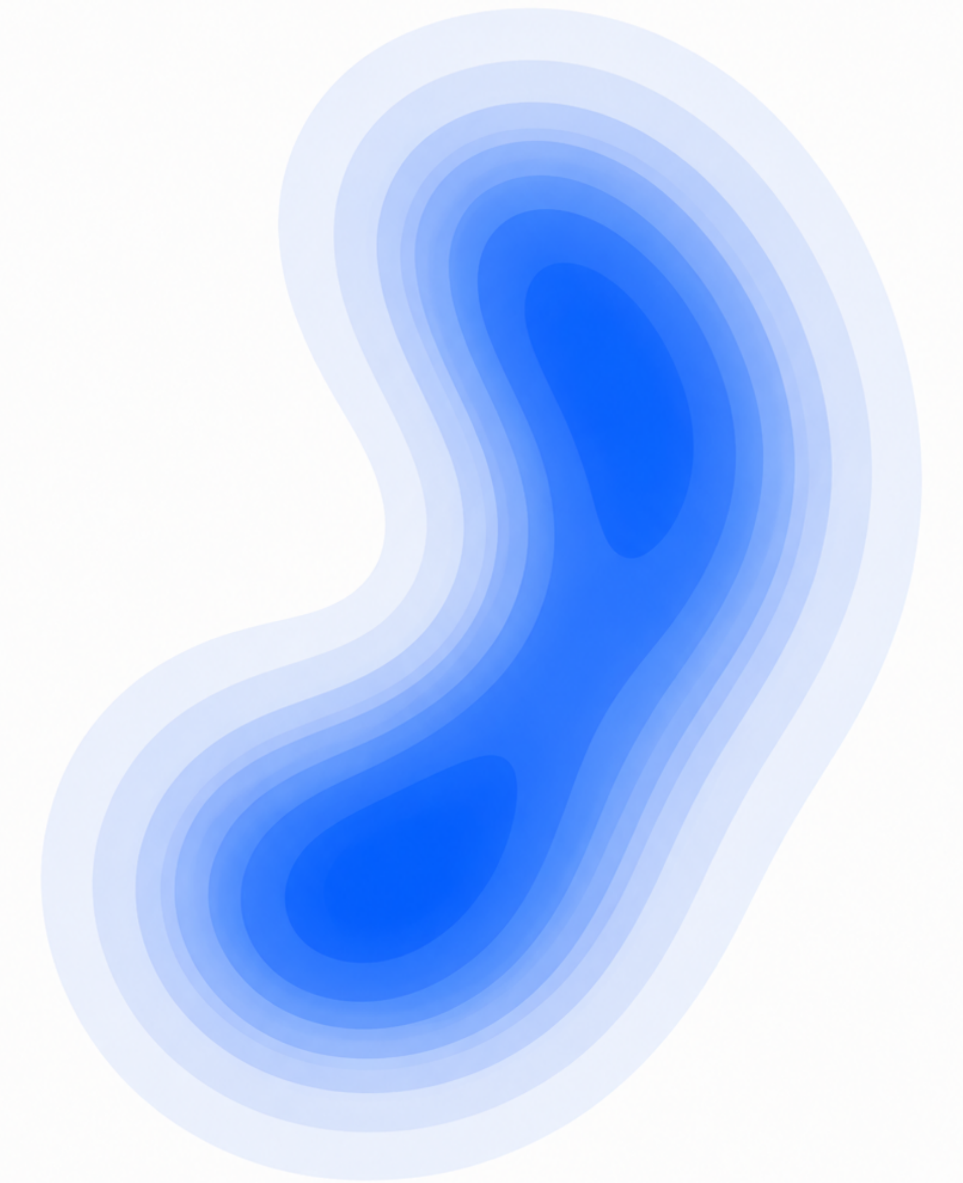
$$p_{\text{data}}$$

$\star$



$$\mathcal{N}(0, \sigma_2^2 \text{Id})$$

$=$



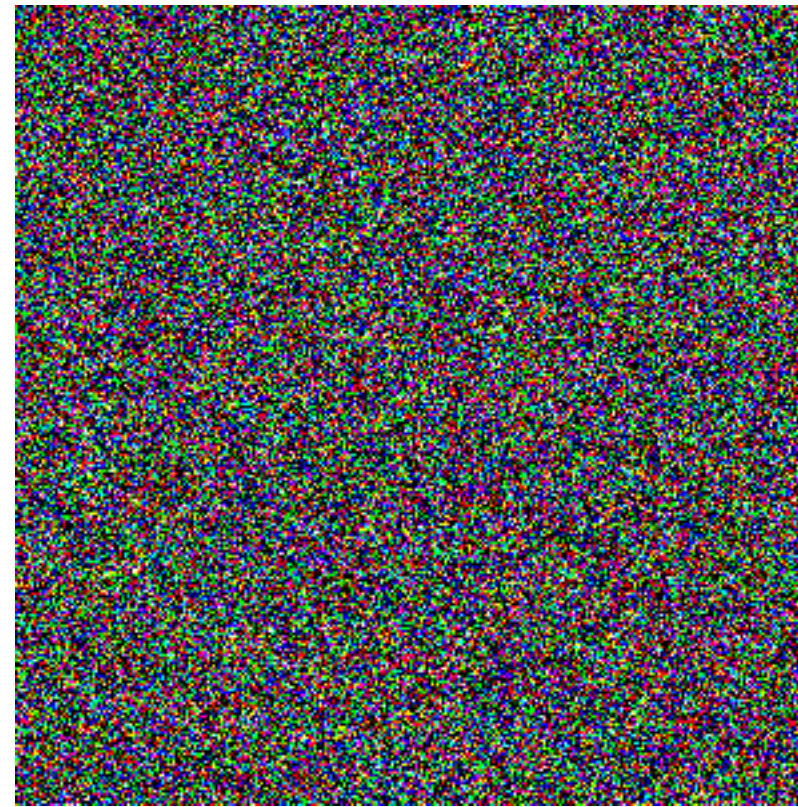
$$p_2 = p_{\text{data}} \star \mathcal{N}(0, \sigma_2^2 \text{Id})$$

Forward noising process: from data to Gaussian



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

$+$ $\sigma_3 \cdot$

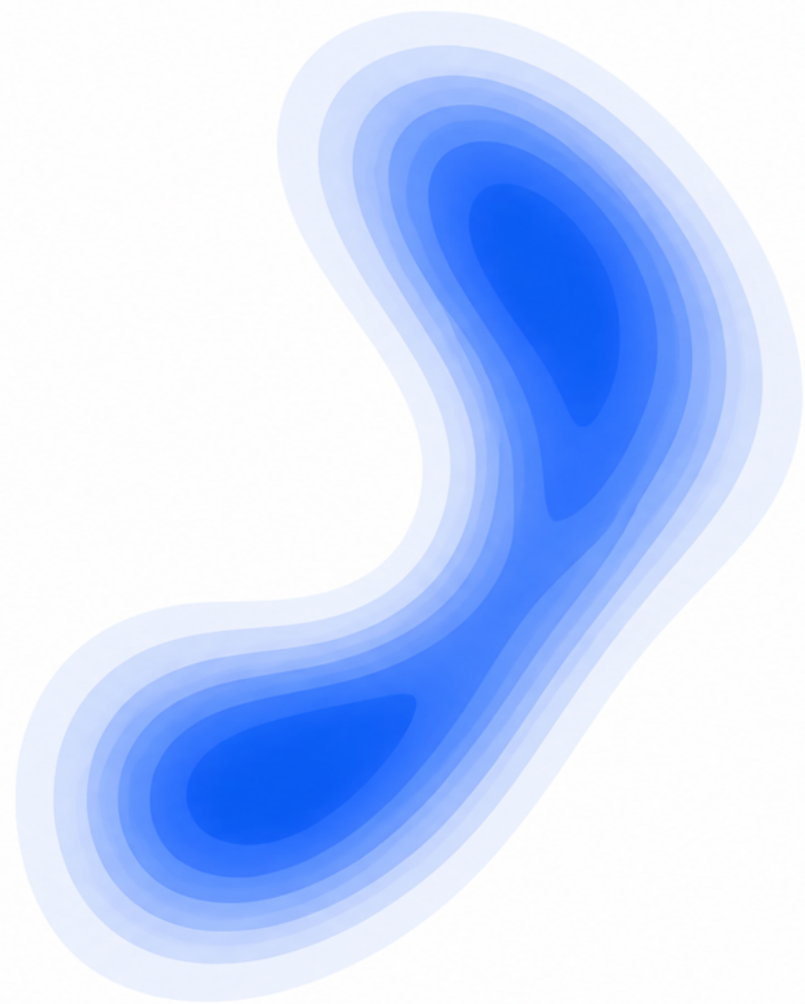


$$w \sim \mathcal{N}(0, \text{Id})$$

$=$

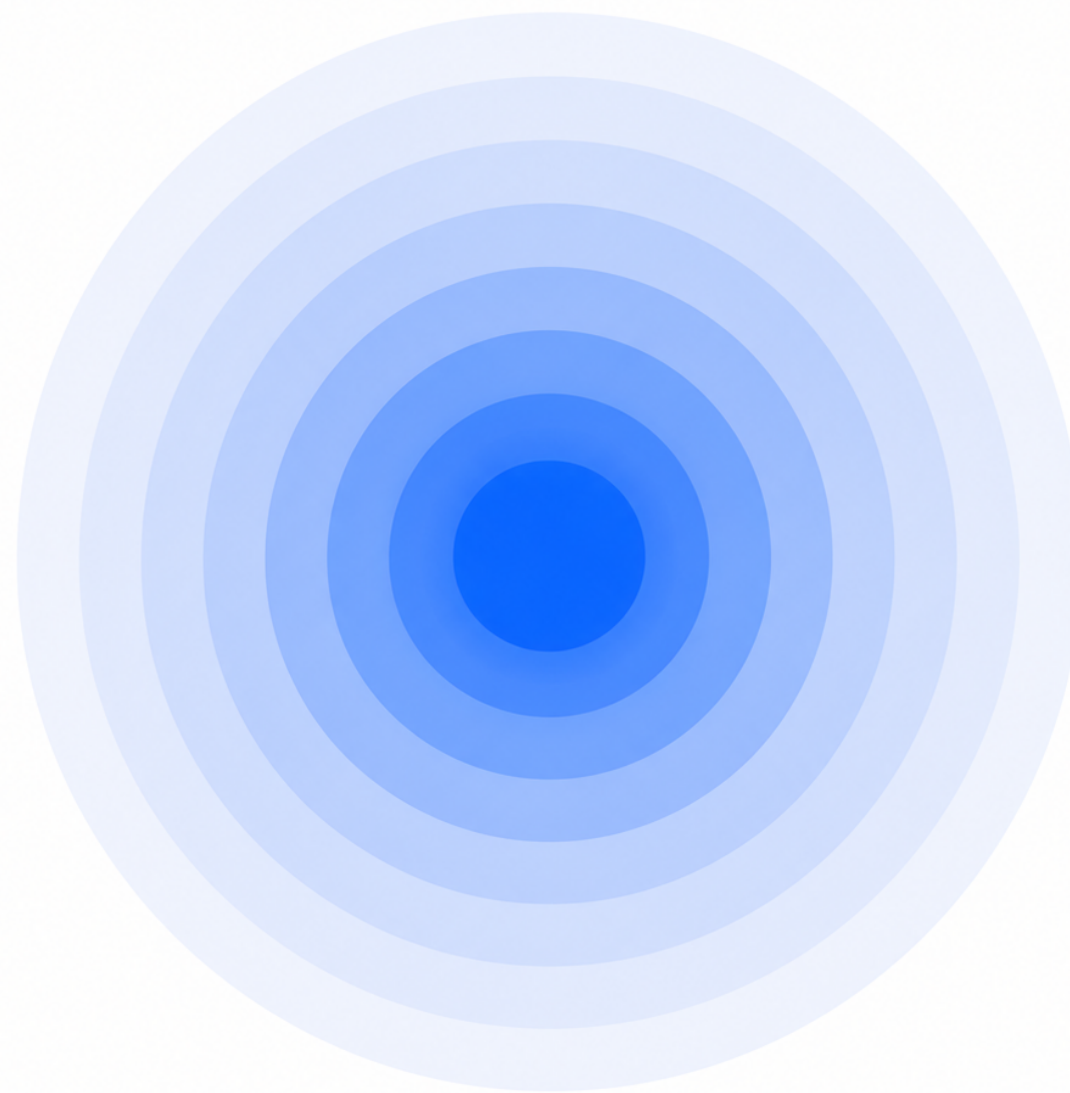


$$y = x + \sigma_3 w$$



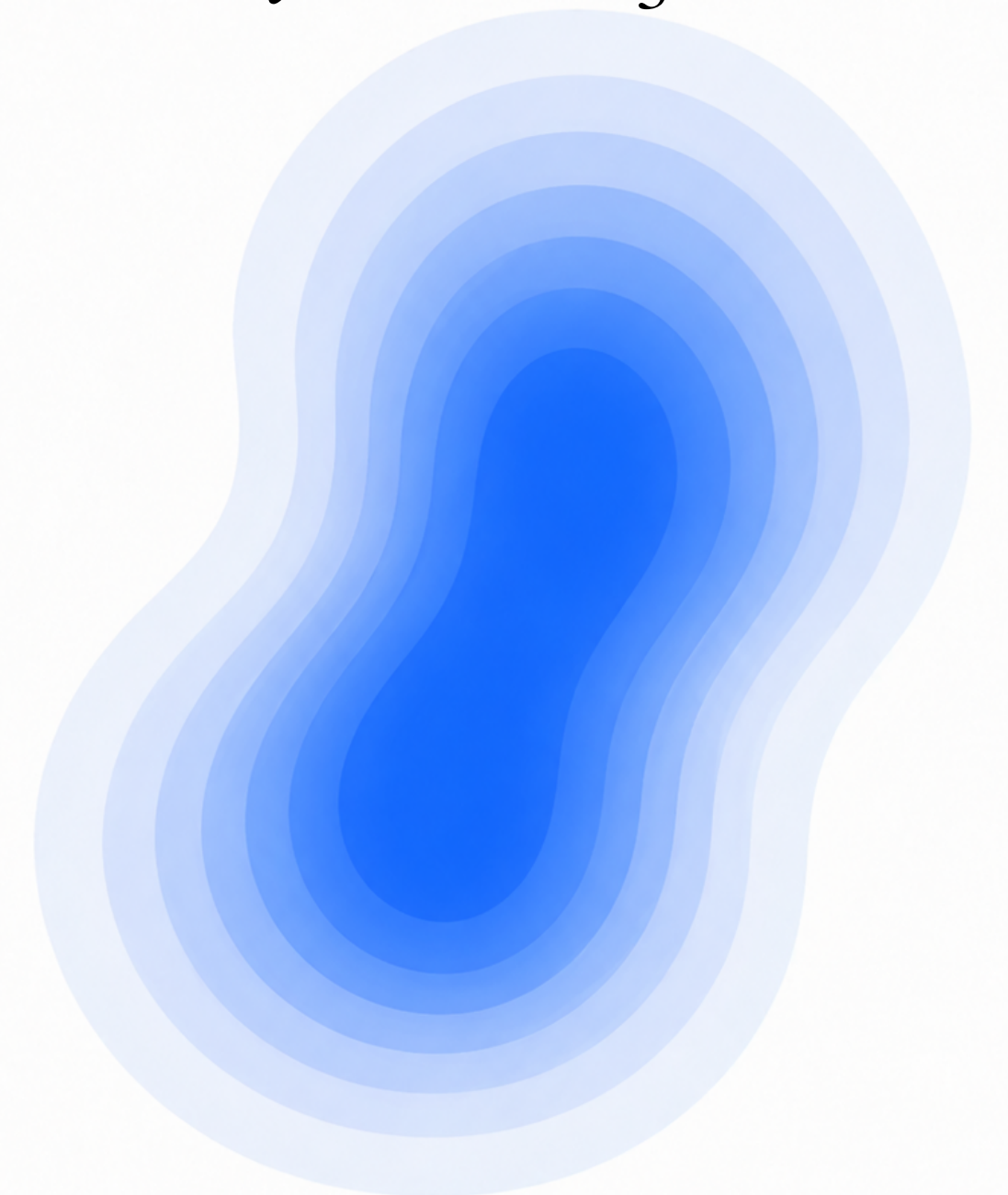
$$p_{\text{data}}$$

$\star$



$$\mathcal{N}(0, \sigma_3^2 \text{Id})$$

$=$



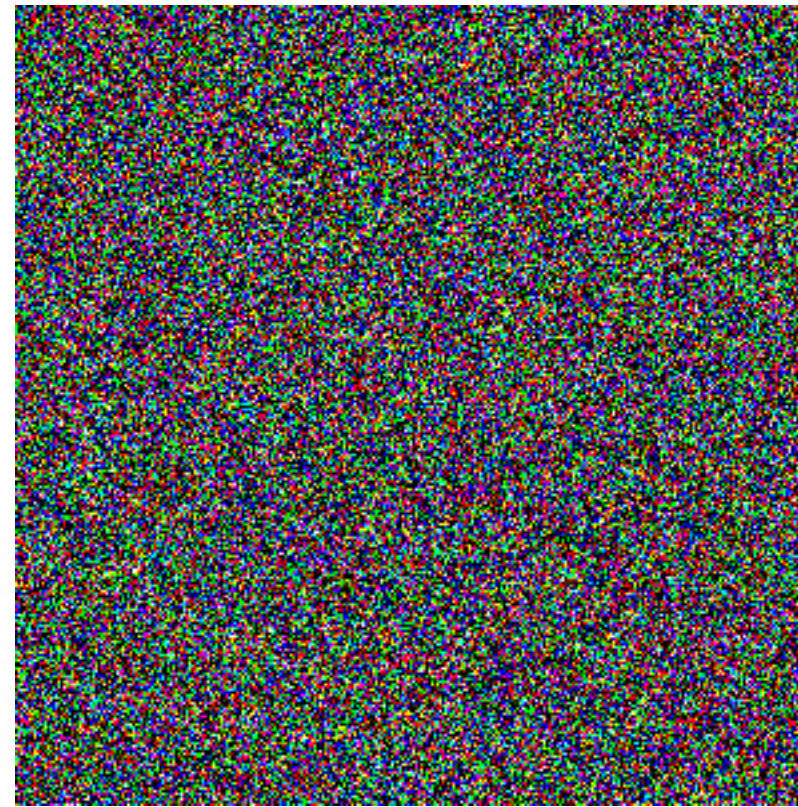
$$p_3 = p_{\text{data}} \star \mathcal{N}(0, \sigma_3^2 \text{Id})$$

Forward noising process: from data to Gaussian



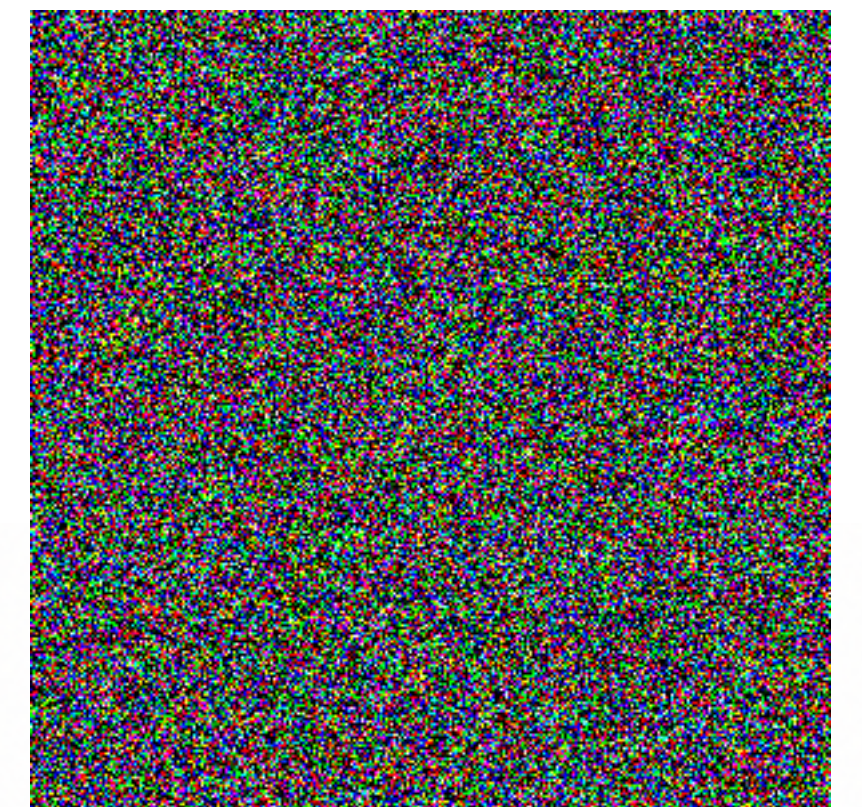
$$x \sim p_{\text{data}}$$

$+$ $\sigma_4 \cdot$

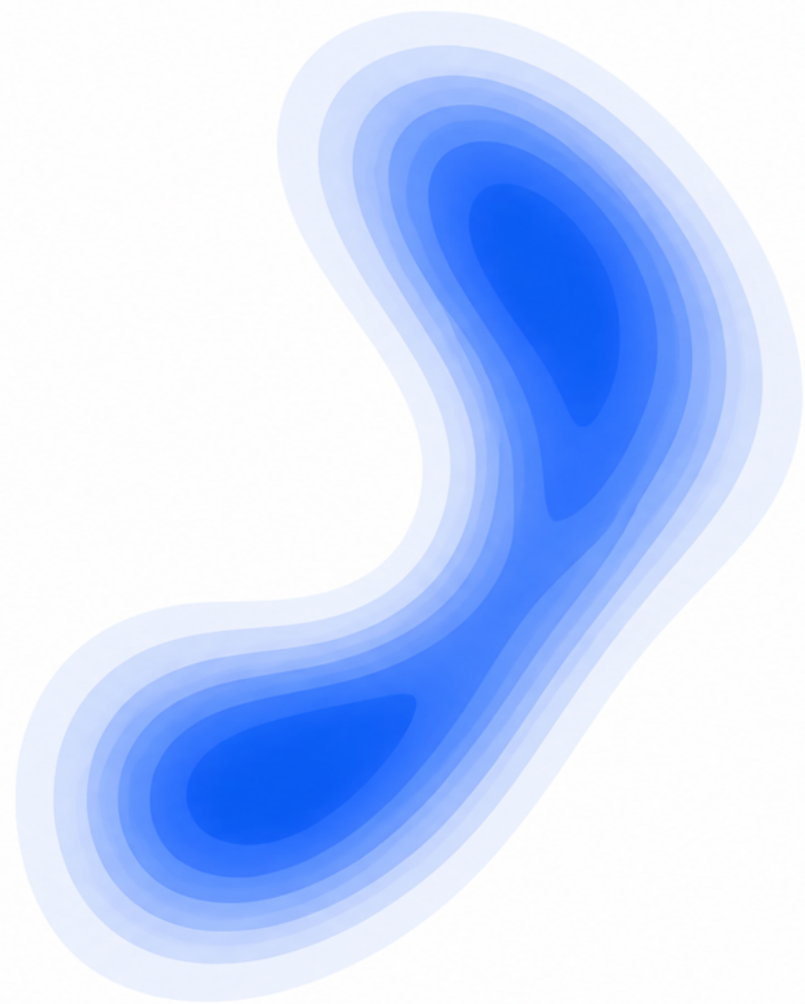


$$w \sim \mathcal{N}(0, \text{Id})$$

$=$

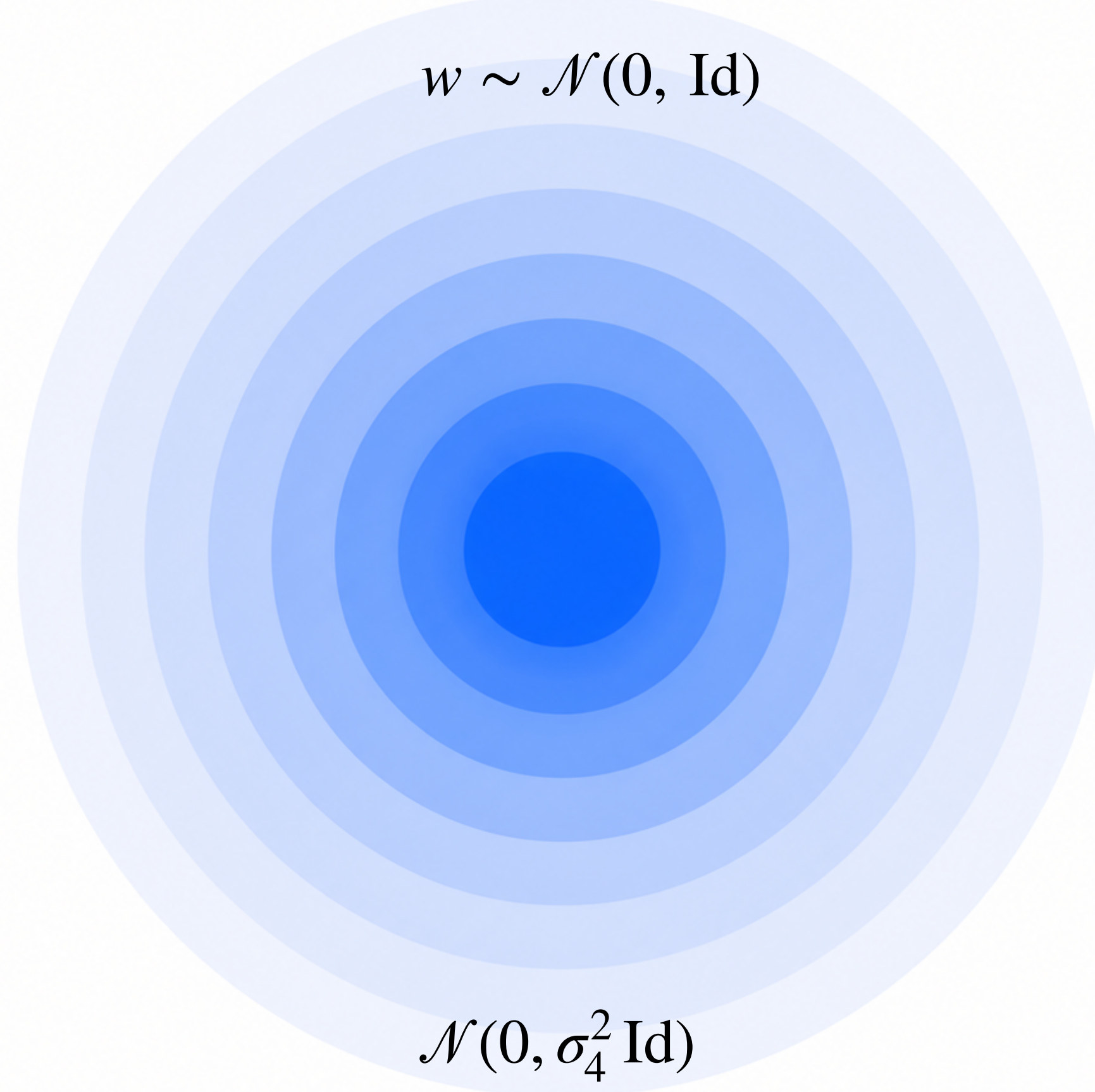


$$y = x + \sigma_4 w$$



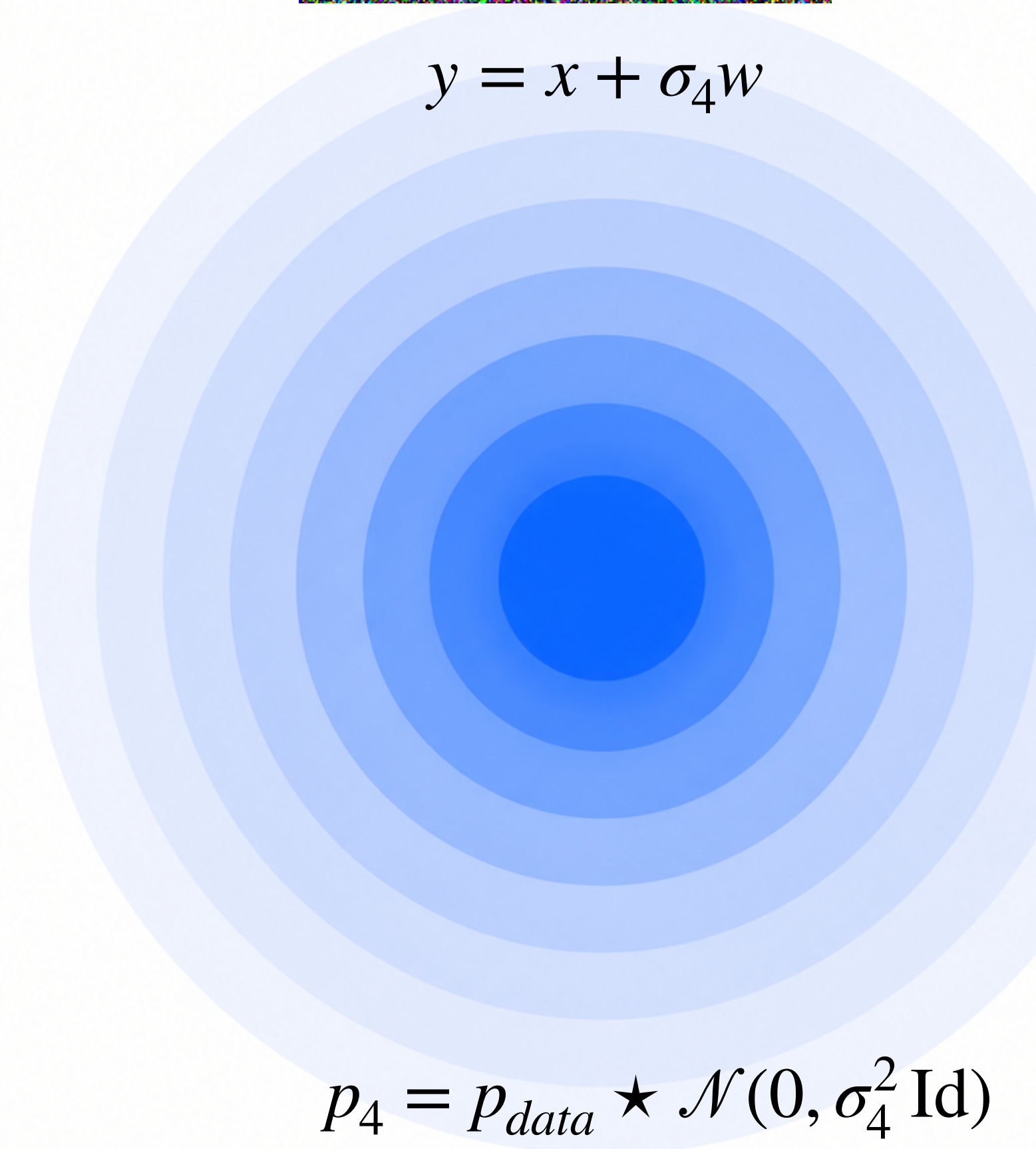
$$p_{\text{data}}$$

$\star$



$$\mathcal{N}(0, \sigma_4^2 \text{Id})$$

$=$

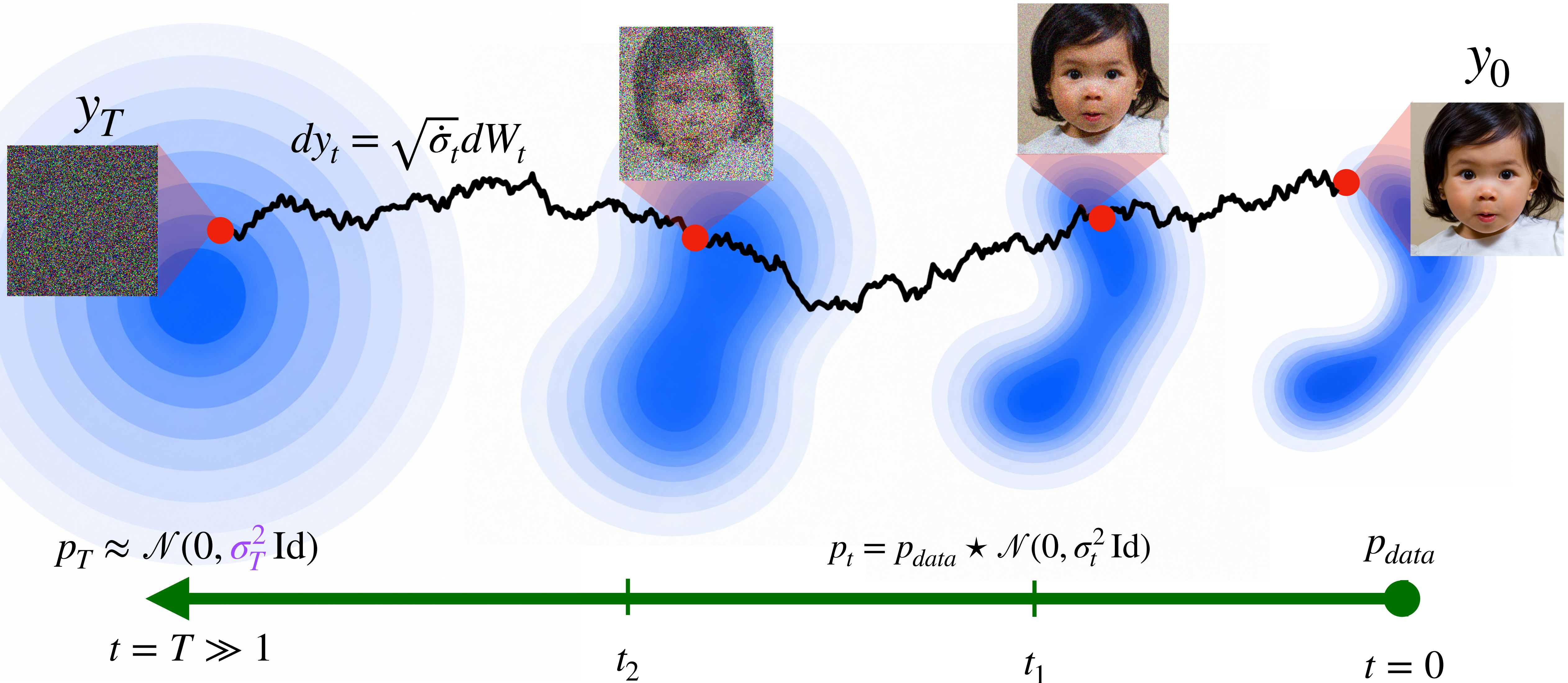


$$p_4 = p_{\text{data}} \star \mathcal{N}(0, \sigma_4^2 \text{Id})$$

Forward noising process: Variance Exploding

$$y_t \stackrel{d}{=} y_0 + \sigma_t w, \quad w \sim \mathcal{N}(0, \mathbf{I})$$

With increasing noise level σ_t e.g. $\sigma_t = t^\beta$

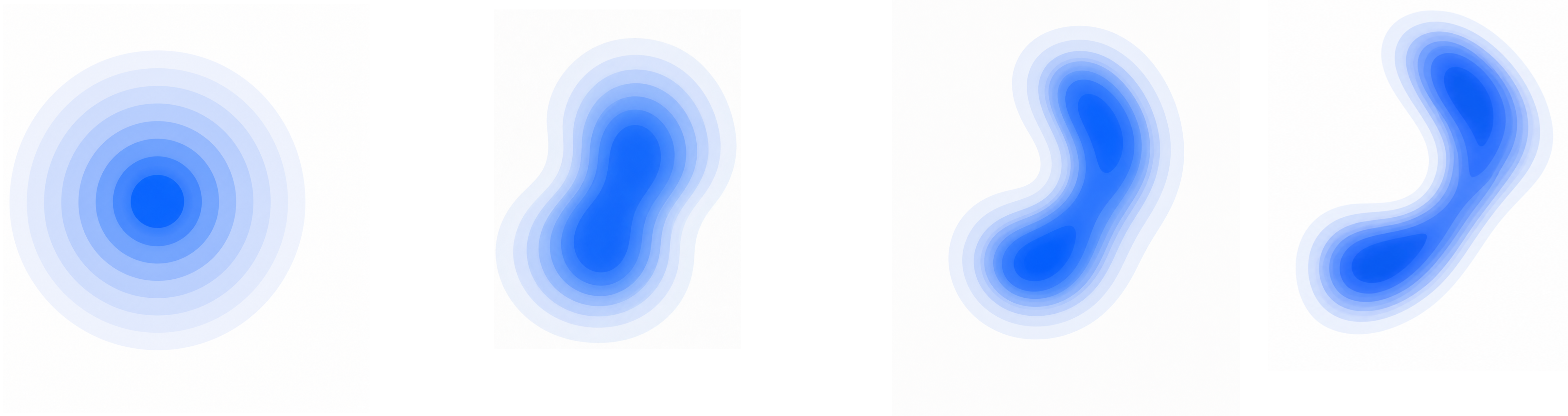


Forward noising process: Variance Preserving

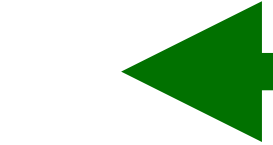
$$y_t \stackrel{d}{=} \eta_t(y_0 + \sigma_t w), \quad w \sim \mathcal{N}(0, I)$$

With increasing noise level σ_t

$$\eta_t = \sqrt{\frac{1}{1 + \sigma_t^2}}$$



$$p_T \approx \mathcal{N}(0, \text{Id})$$



$$t = T \gg 1$$

$$t_2$$

$$t_1$$

$$p_{data}$$

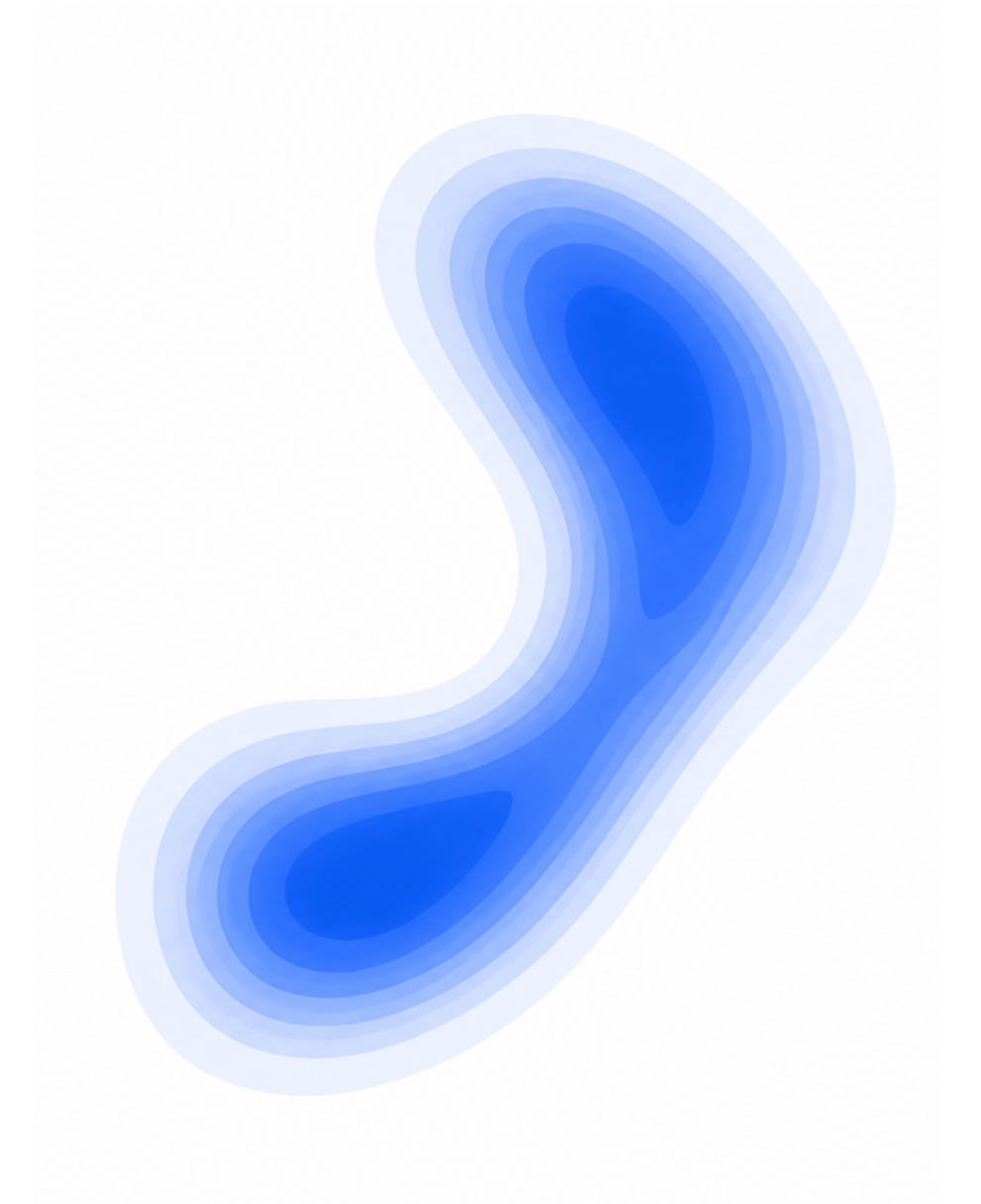
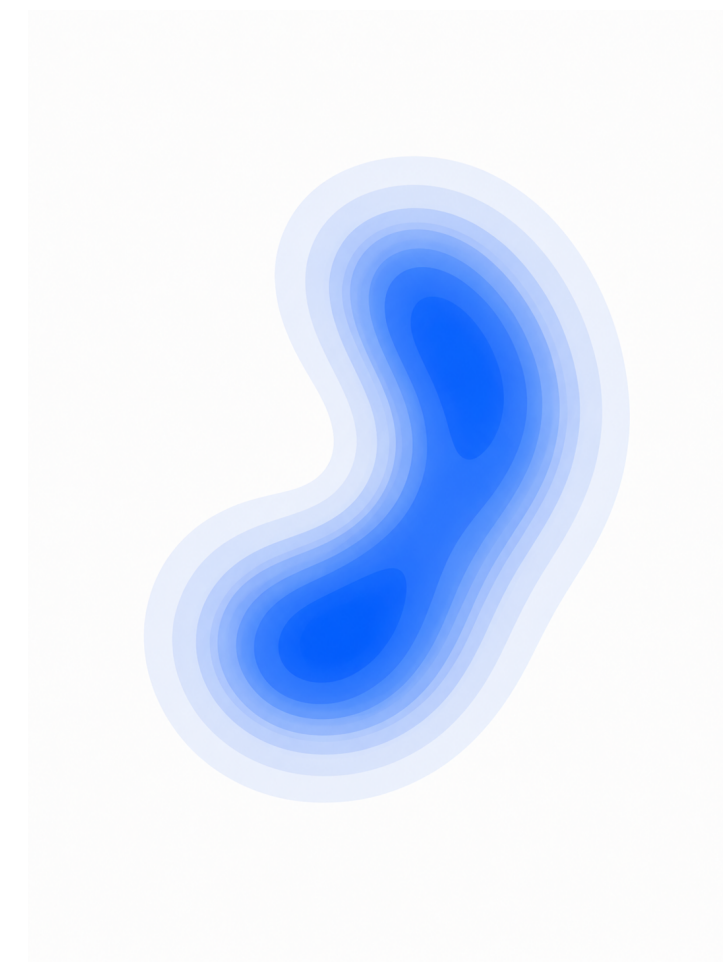
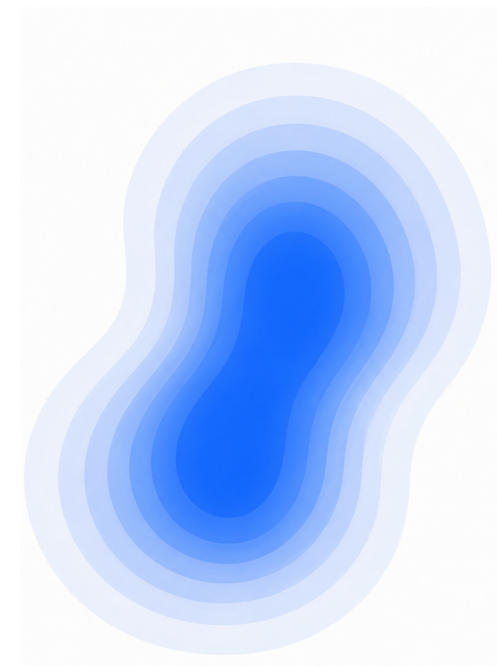
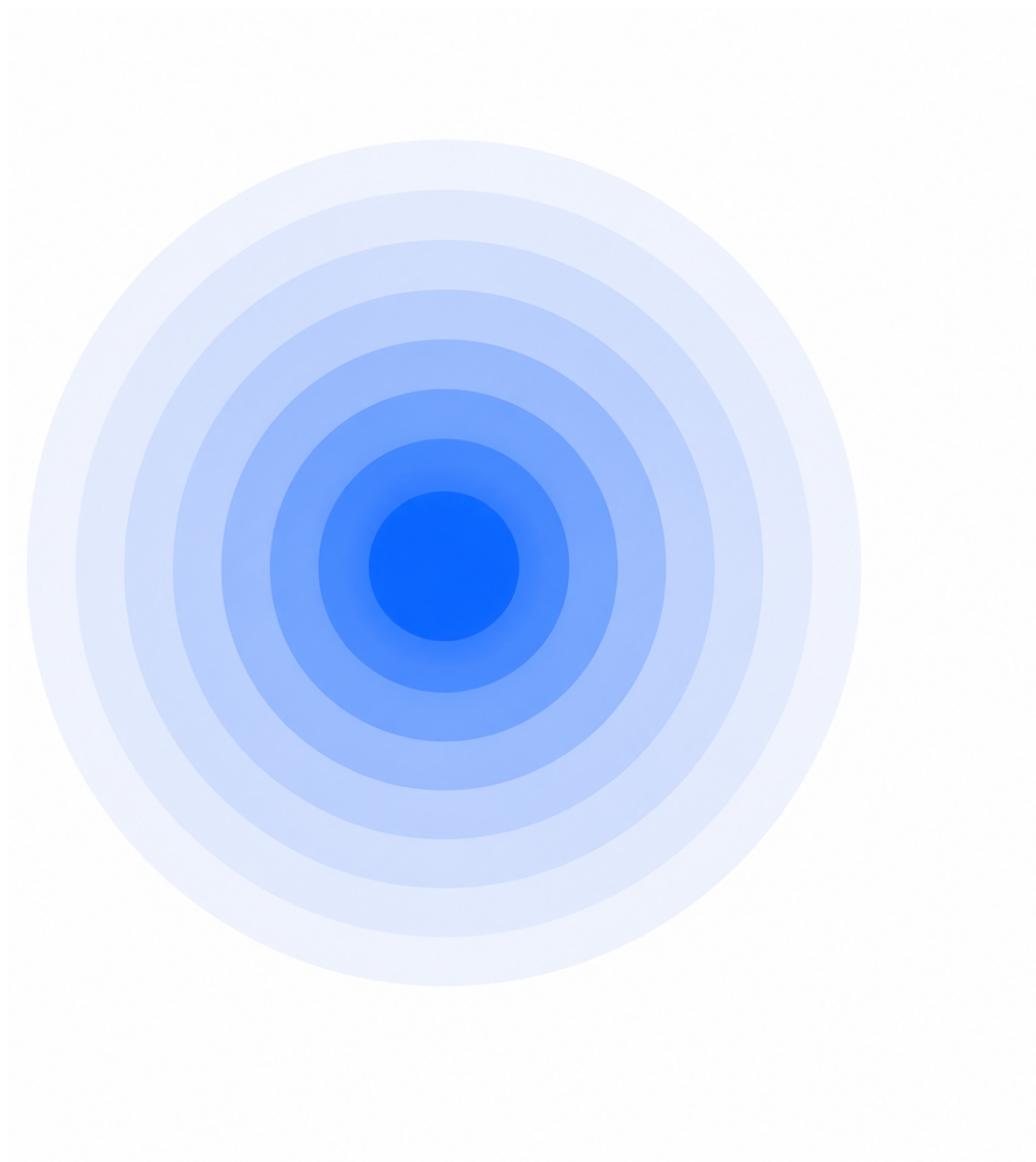
$$t = 0$$

Forward noising process: Flow Matching

$$y_t \stackrel{d}{=} \eta_t(y_0 + \sigma_t w), \quad w \sim \mathcal{N}(0, \mathbf{I})$$

With increasing noise level σ_t

$$\eta_t = \frac{1}{1 + \sigma_t}$$



$$p_T \approx \mathcal{N}(0, \text{Id})$$

$$t = T \gg 1$$

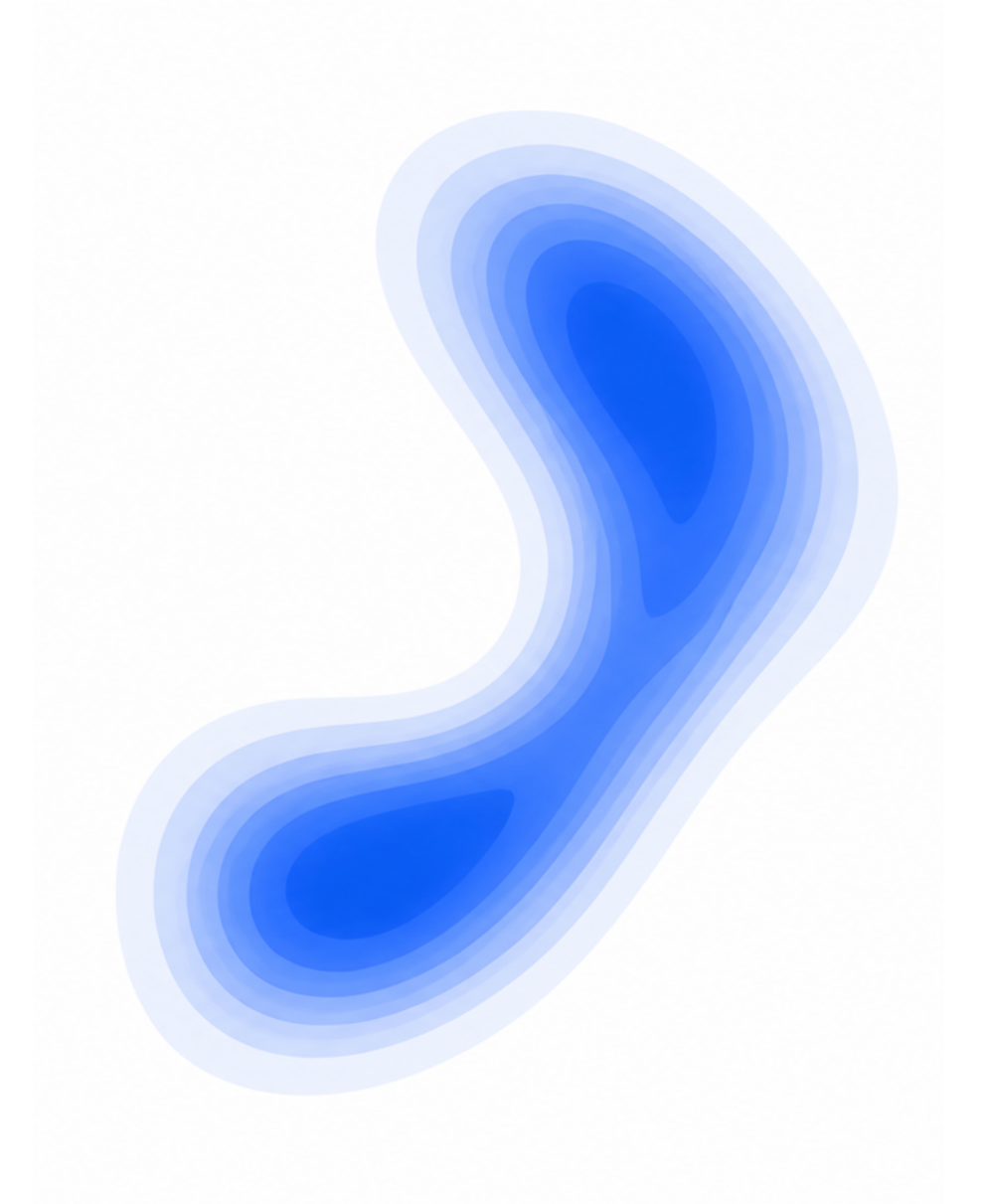
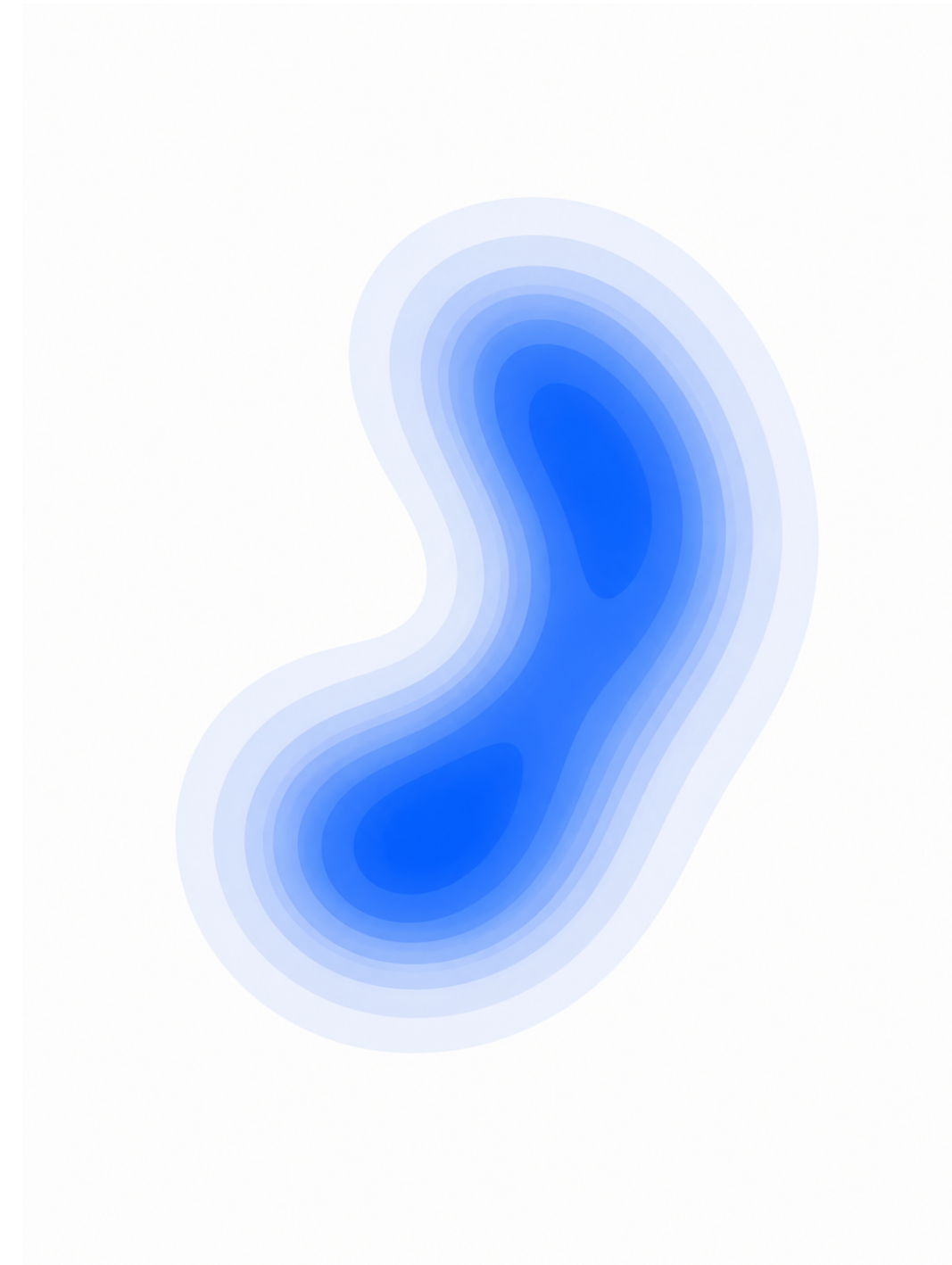
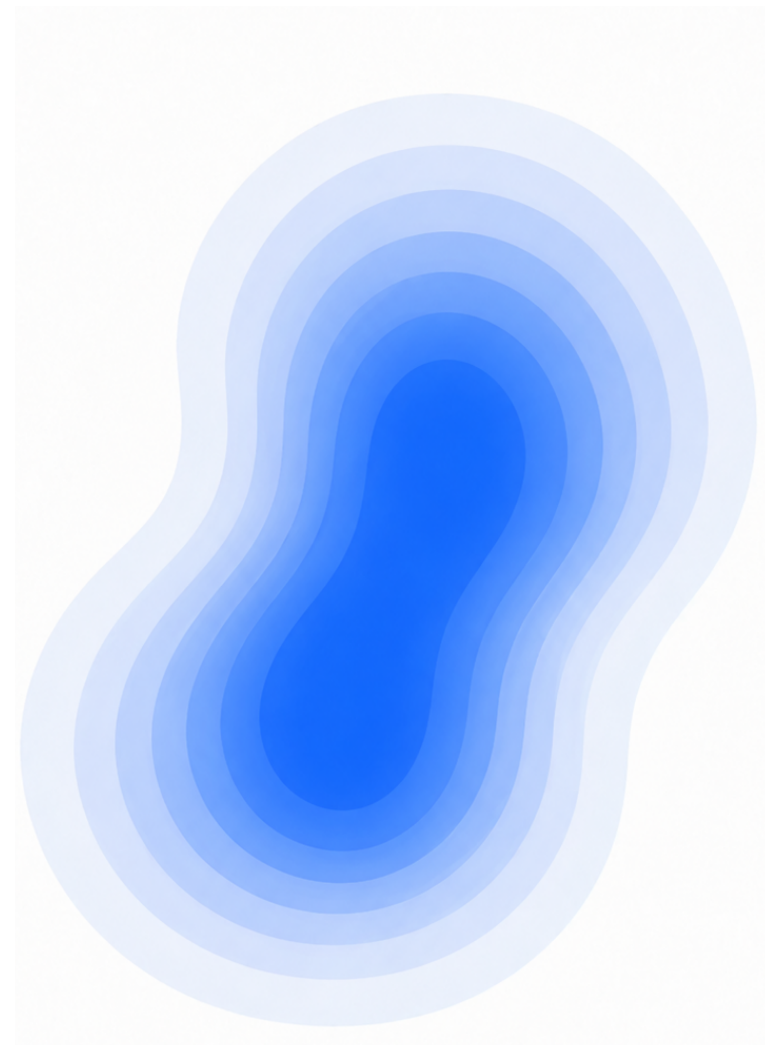
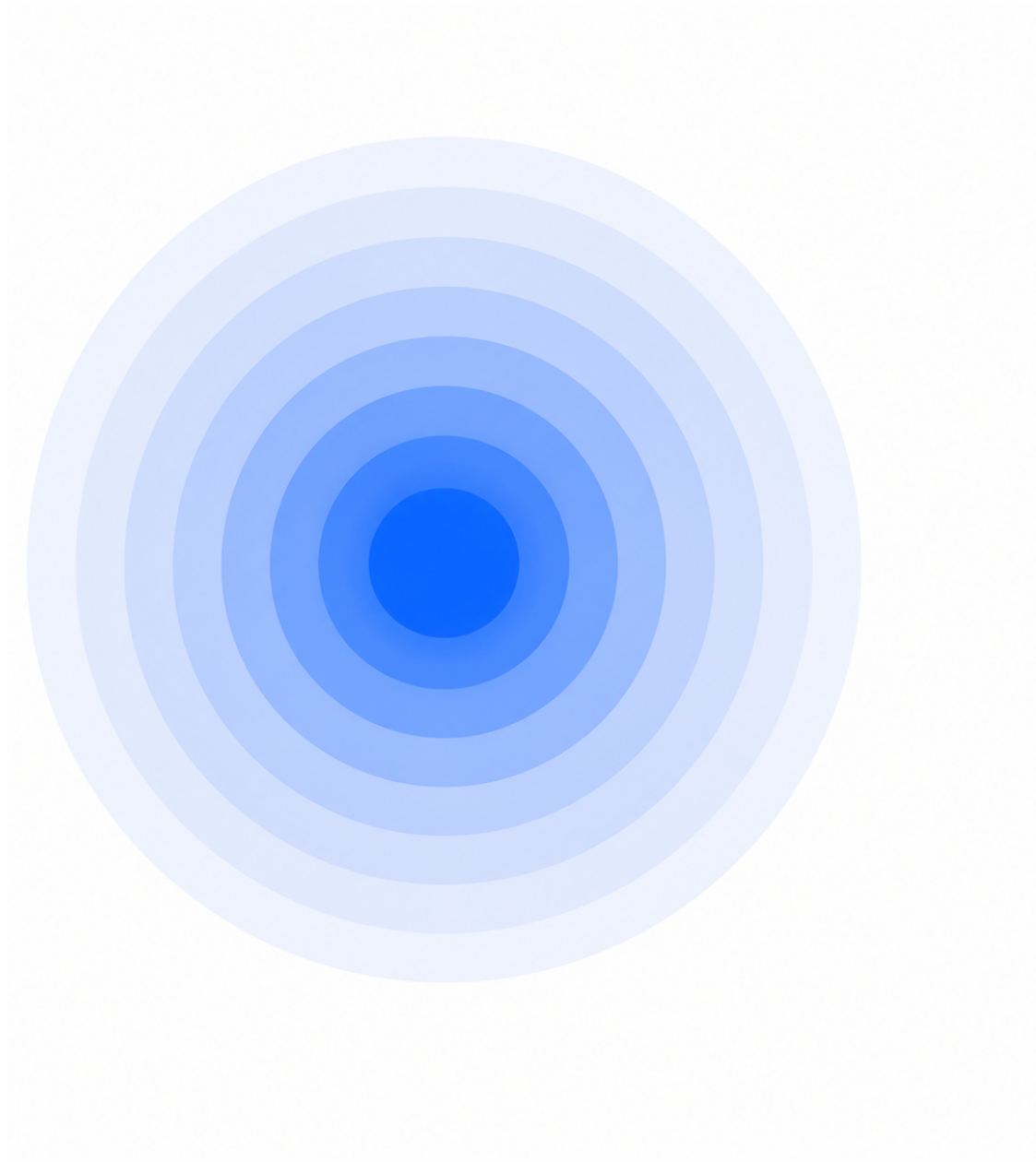
$$t_2$$

$$t_1$$

$$p_{data}$$

$$t = 0$$

Diffusion models



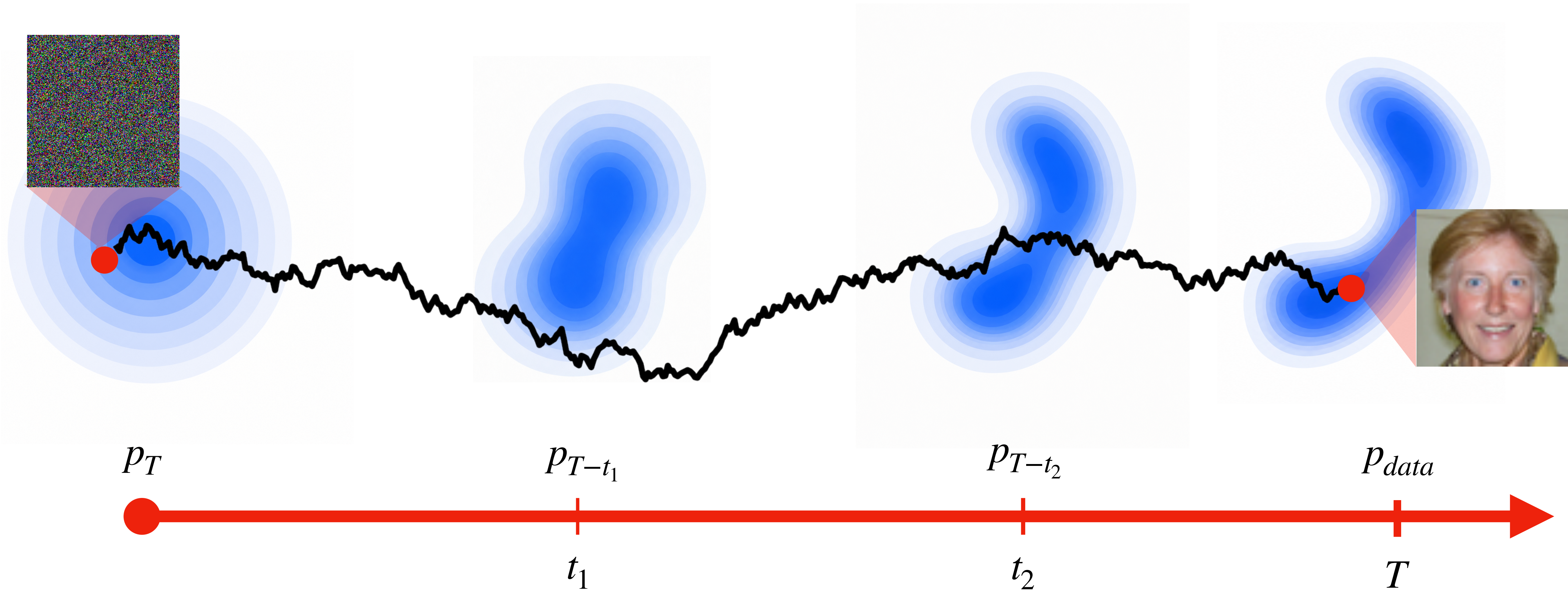
Generative process

Generative process: from noise to data

For any $\alpha \geq 0$

$$x_0 \sim p_T, \quad dx_t = v_t(x_t) dt + \sqrt{2\alpha a_t} dW_t \quad \Rightarrow x_t \sim p_{T-t}$$

$$v_{T-t}(x) = -\frac{\dot{\eta}_t}{\eta_t} x + (1+\alpha) \eta_t^2 \dot{\sigma}_t \sigma_t \nabla \log p_t(x) \quad a_{T-t} = \eta_t^2 \dot{\sigma}_t \sigma_t$$

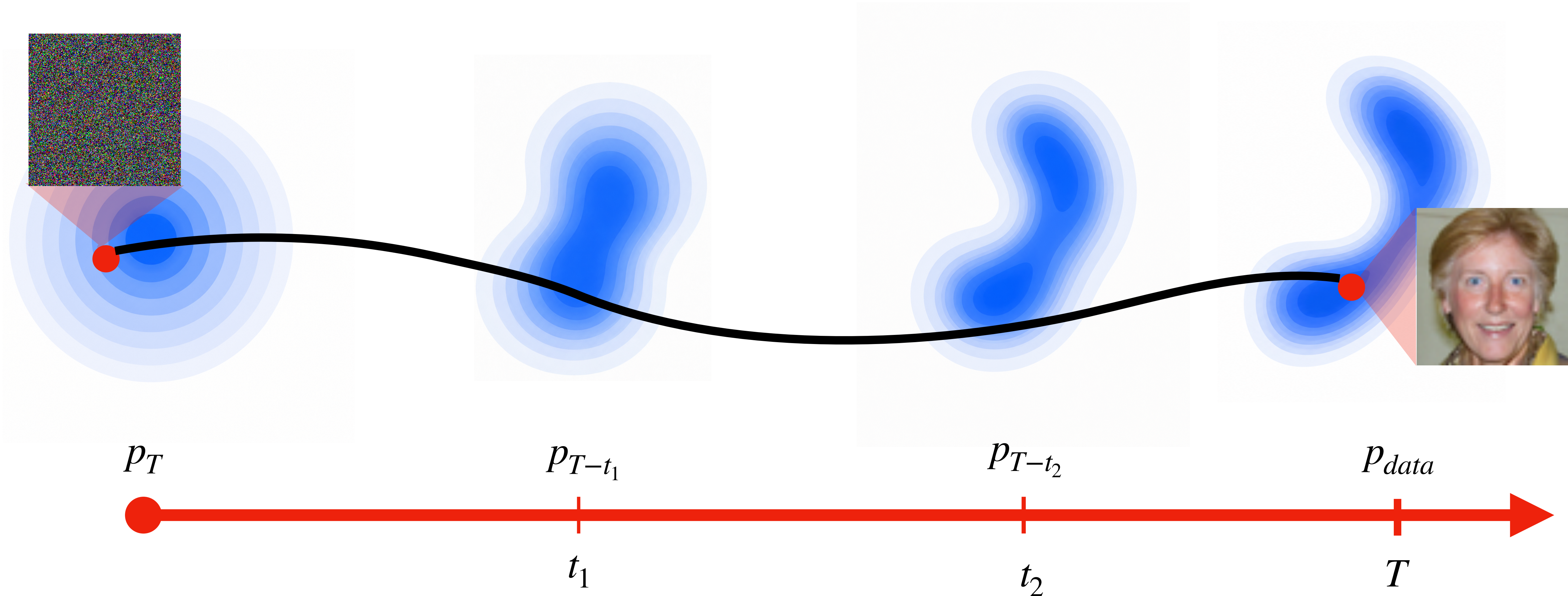


Generative process: from noise to data

$$x_0 \sim p_T, \quad dx_t = v_t(x_t) dt$$

$$v_{T-t}(x) = -\frac{\dot{\eta}_t}{\eta_t} x + \eta_t^2 \dot{\sigma}_t \sigma_t \nabla \log p_t(x)$$

$$a_{T-t} = \eta_t^2 \dot{\sigma}_t \sigma_t$$



Diffusion models parameters

$$x_0 \sim p_T, \quad dx_t = v_t(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_{T-t}(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

Parameters of the algorithm: diffusion coefficient α , time schedulers η_t, σ_t

[Hurault et al., '25 '26] Optimization of parameters to minimize sampling errors

From Score Matching to Diffusion:
A Fine-Grained Error Analysis in the Gaussian Setting

Samuel Hurault
ENS Paris, PSL, CNRS
samuel.hurault@ens.fr

Matthieu Terris
Univ. Paris-Saclay, Inria, CEA
matthieu.terris@inria.fr

Thomas Moreau
Univ. Paris-Saclay, Inria, CEA
thomas.moreau@inria.fr

Gabriel Peyré
ENS Paris, PSL, CNRS
gabriel.peyre@ens.fr

Geometry-Aware Discretization Error of
Diffusion Models

Samuel Hurault
Univ. Gustave Eiffel, CNRS, LIGM
samuel.hurault@univ-eiffel.fr

Thomas Moreau
Univ. Paris-Saclay, Inria, CEA
thomas.moreau@inria.fr

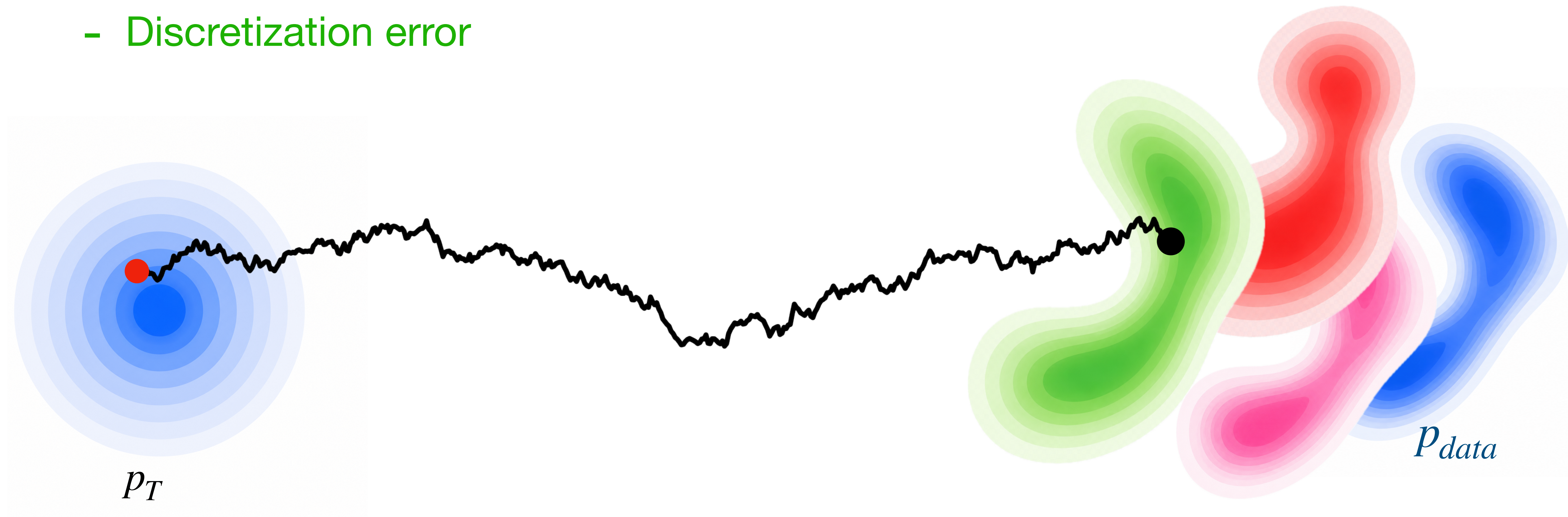
Gabriel Peyré
CNRS, ENS, PSL Université
gabriel.peyre@ens.fr

Diffusion models errors

$$x_0 \sim p_T, \quad dx_t = v_t(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_{T-t}(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

- initialization error
- Score training error
- Discretization error



Diffusion models errors

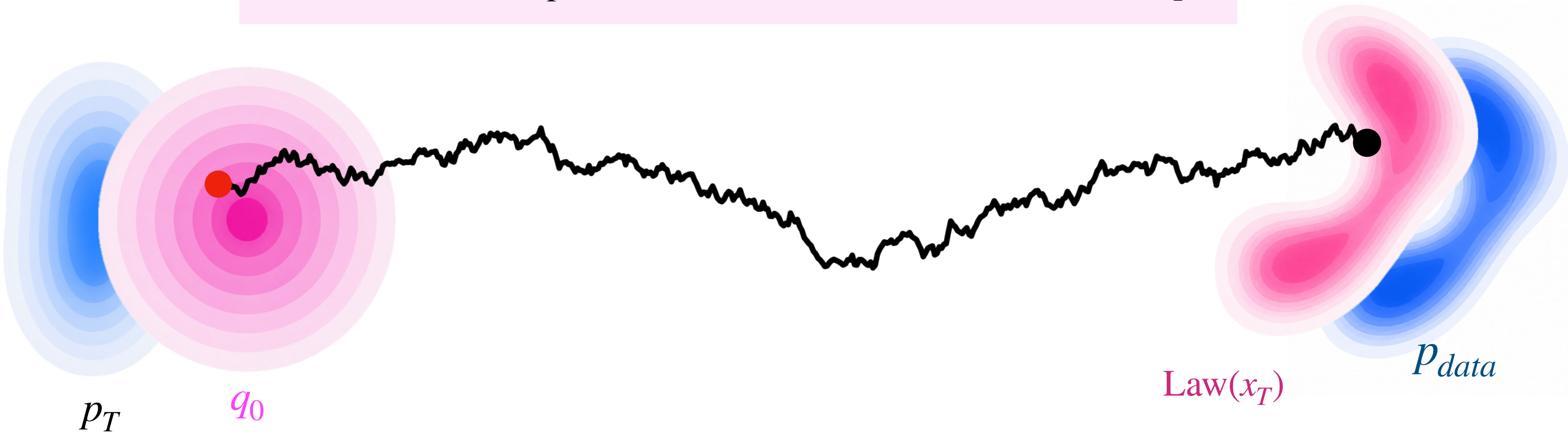
Initialization error

$$x_0 \sim q_0, \quad dx_t = v_t(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_{T-t}(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

The process is initialized at $q_0 \neq p_T = \text{Law}(\eta_T(x_0 + \sigma_T w))$ *Variance Exploding:* $q_0 = \mathcal{N}(0, \sigma_T^2 \mathbf{I})$
Variance Preserving: $q_0 = \mathcal{N}(0, \mathbf{I})$

$$W_2(p_T, q_0) = O(\sigma_T^{-1}) \quad \text{error negligible for large enough } \sigma_T$$



Diffusion models errors: **score training error**

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_{T-t}^\theta(x) = f(\alpha, \eta_t, \sigma_t, s_t^\theta(x))$$

Denoising Score Matching

s_t^θ trained to approximate $\nabla \log p_t$ by :

$$\min_{\theta} \int_t \mathbb{E}_{x \sim p_{data}} \mathbb{E}_{w \sim \mathcal{N}(0, I)} \left\| s_t^\theta(x + \sigma_t w) + \frac{w}{\sigma_t} \right\|^2 dt + cst$$

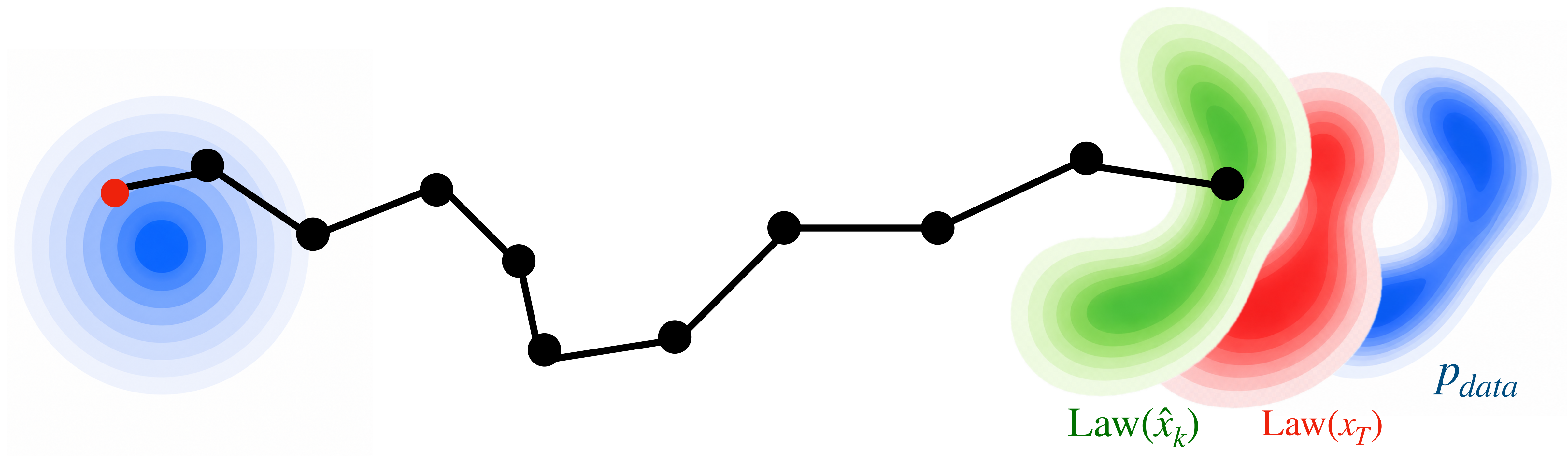
- Generalization error ($N_{data} < \infty$)
- Optimization error
- Approximation error ($|\theta| < \infty$)



Diffusion models errors: discretization error

Euler discretization
with stepsize $\gamma > 0$

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}^\theta(\hat{x}_k) + \sqrt{2\gamma\alpha_{t_k}} \hat{W}_k$$

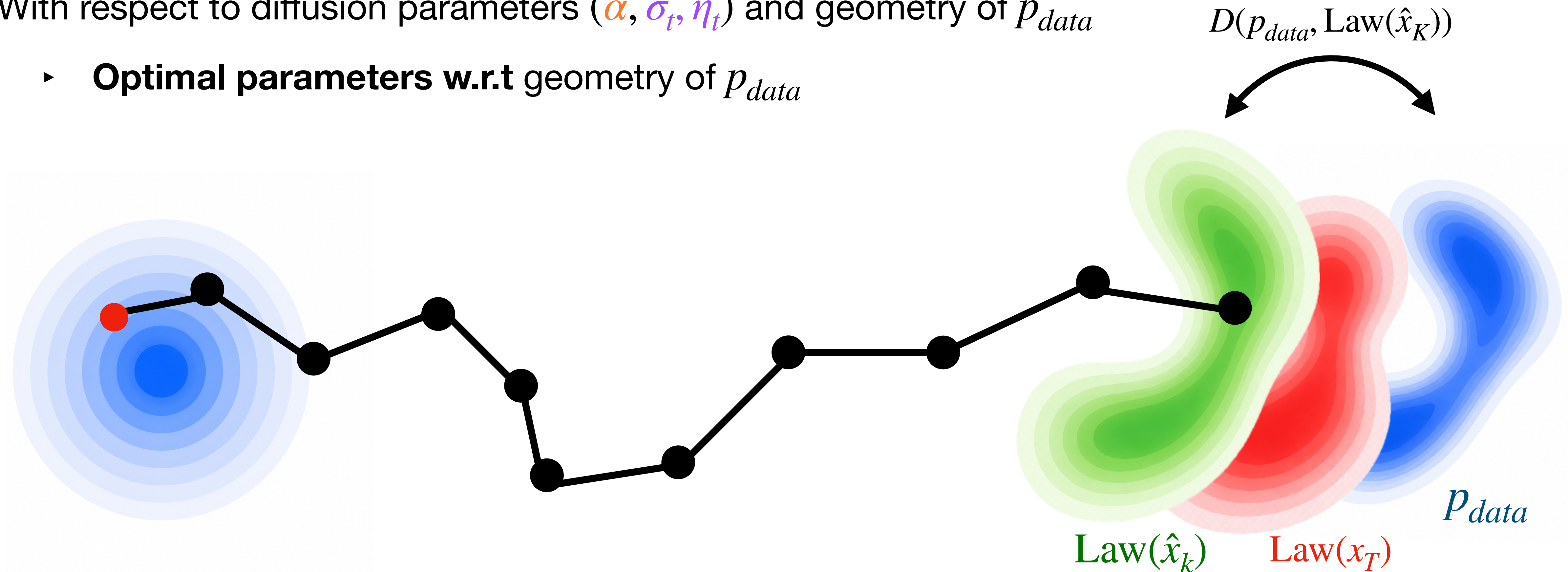


Diffusion models errors

Euler discretization
with stepsize $\gamma > 0$

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}^\theta(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

- We calculate explicit and exact sampling error due to **training** and **discretization**
- With respect to diffusion parameters (α, σ_t, η_t) and geometry of p_{data}
 - **Optimal parameters w.r.t** geometry of p_{data}



Diffusion models errors

Euler discretization
with stepsize $\gamma > 0$

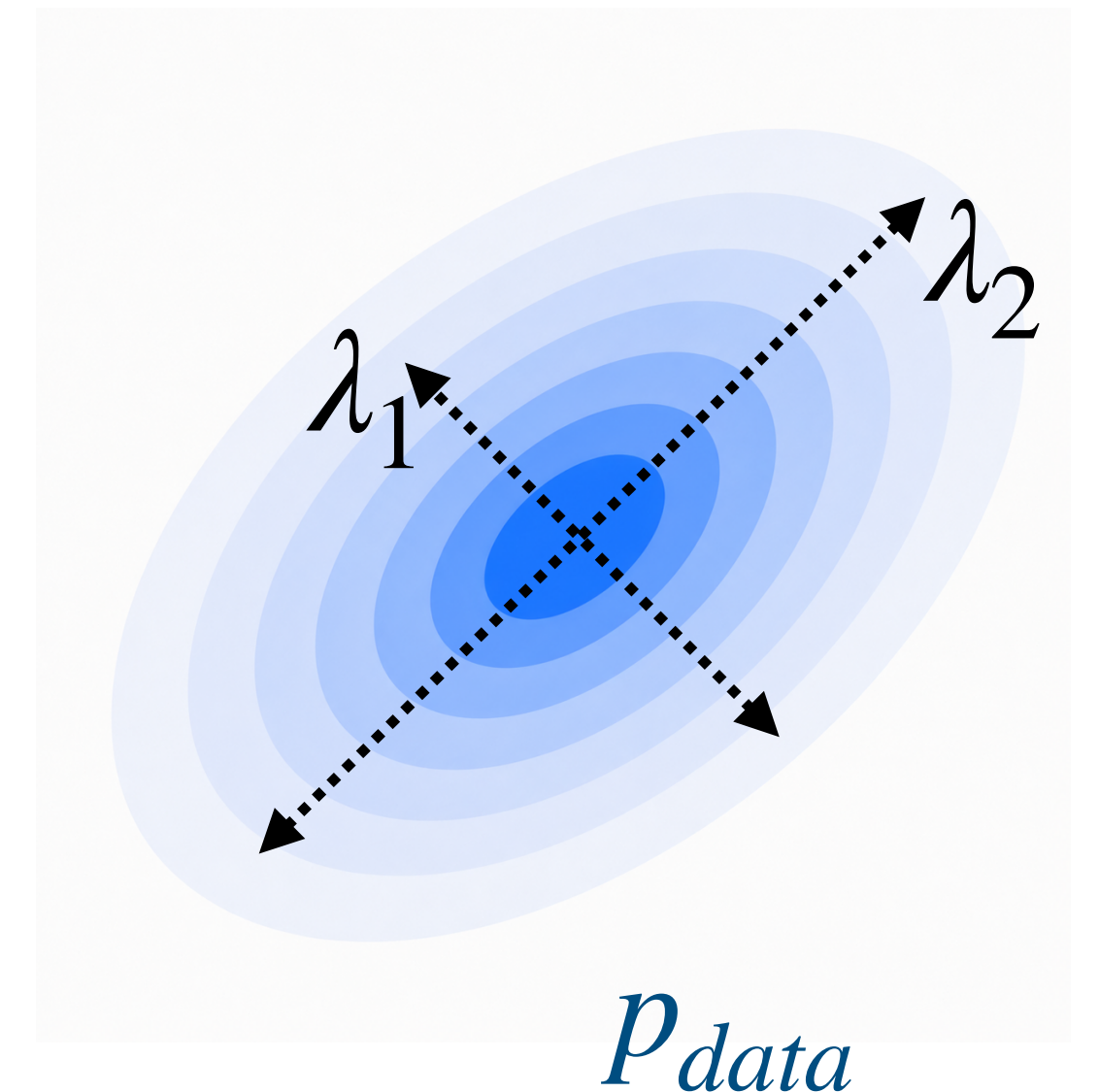
$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha_{t_k}} \hat{W}_k$$

- We calculate explicit and exact sampling error due to **training** and **discretization**
- With respect to diffusion parameters (α, σ_t, η_t) and geometry of p_{data}
 - **Optimal parameters w.r.t** geometry of p_{data}

Assuming Gaussian data

$$p_{data} \sim \mathcal{N}(\mu_{data}, \Sigma_{data} = U \text{Diag}(\lambda_i) U^\top)$$

- Verify if predictions stand for **real image (non-Gaussian)** data



Existing works: For **general** p_{data} , upper bounds

$$D(p_{data}, \text{Law}(\hat{Y}_K)) \lesssim \mathcal{E}_{disc} + \mathcal{E}_{train}$$

- Not tight
- No influence of **geometry of** p_{data}

Diffusion models errors

Euler discretization
with stepsize $\gamma > 0$

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha_t} \hat{W}_k$$

- We calculate explicit and exact sampling error due to **training** and **discretization**
- With respect to diffusion parameters (α, σ_t, η_t) and geometry of p_{data}

- **Optimal parameters w.r.t** geometry of p_{data}

Assuming Gaussian data

$$p_{data} \sim \mathcal{N}(\mu_{data}, \Sigma_{data} = U \text{Diag}(\lambda_i) U^\top)$$

- Verify if predictions stand for **real image (non-Gaussian)** data

In this talk:

1. **Discretization** error assuming exact score [H., Terris, Moreau, Peyré '25]
2. **Score training** error assuming *no discretization* [H., Moreau, Peyré '26]

Discretization error for Gaussian data

*Euler discretization
with stepsize $\gamma > 0$*

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

$$v_{T-t}(x) = -\frac{\dot{\eta}_t}{\eta_t} x + (1+\alpha) \eta_t^2 \dot{\sigma}_t \sigma_t \nabla \log p_t(x)$$

Discretization error for Gaussian data

Euler discretization
with stepsize $\gamma > 0$

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

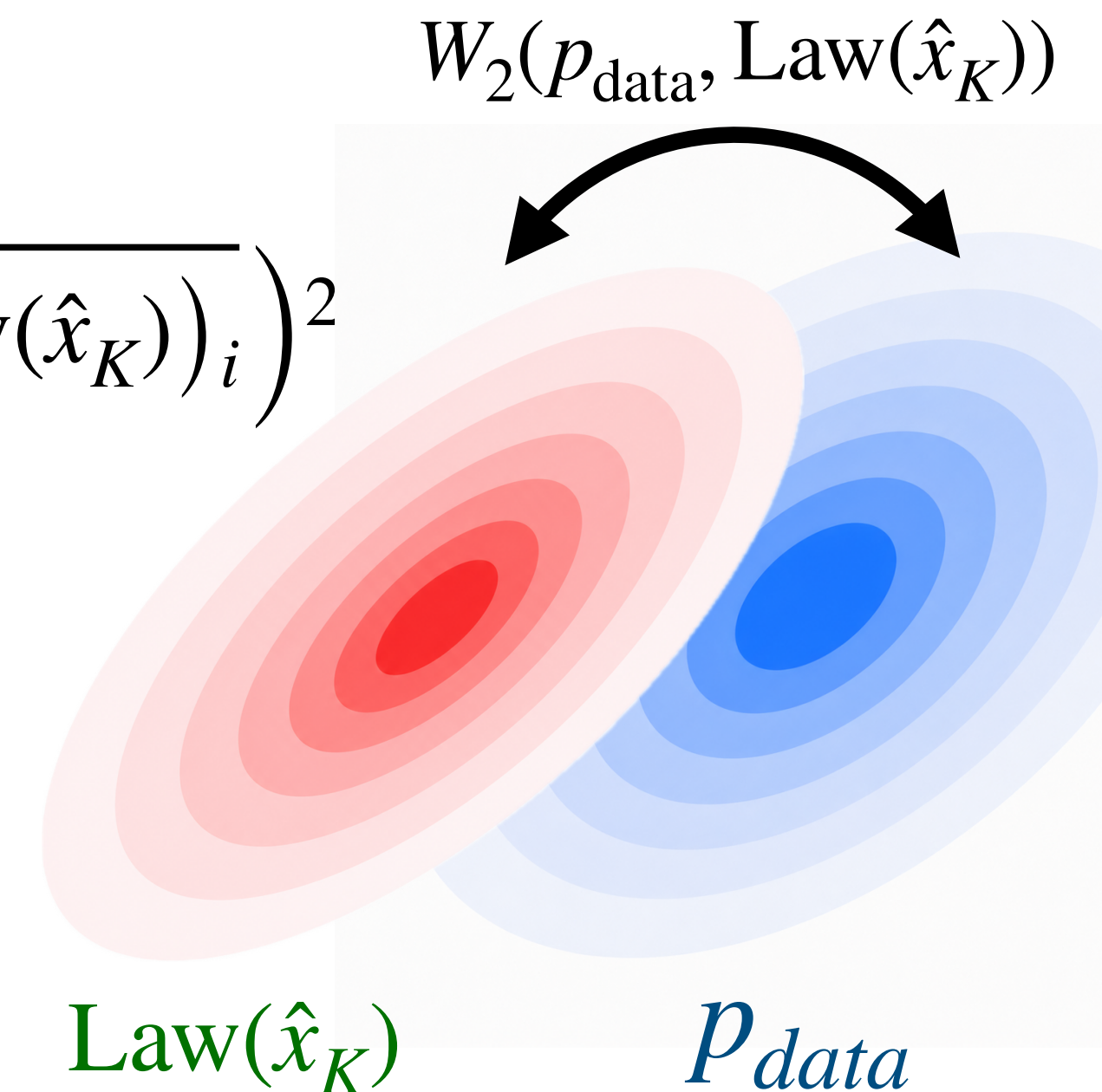
$$v_{T-t}(x) = -\frac{\dot{\eta}_t}{\eta_t} x + (1+\alpha) \eta_t^2 \dot{\sigma}_t \sigma_t \nabla \log p_t(x)$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$ $\nabla \log p_t(x) = -(\eta_t^2(\Sigma_{\text{data}} + \sigma_t^2 \text{Id}))^{-1} x$

- Linear velocity $\rightarrow$ Gaussianity preserved along the process $\text{Law}(\hat{x}_k) = \mathcal{N}(0, \text{Cov}(\hat{x}_k))$
- A_t **diagonal** in eigenbasis of $\Sigma_{\text{data}} \rightarrow \text{Cov}(\hat{x}_k)$ **too**

The final W_2 discretization error:

$$W_2(p_{\text{data}}, \text{Law}(\hat{x}_K))^2 = B(\Sigma_{\text{data}}, \text{Cov}(\hat{x}_K))^2 = \sum_{i=1}^d \left(\sqrt{\lambda_i} - \sqrt{\lambda(\text{Cov}(\hat{x}_K))_i} \right)^2$$



Discretization error for Gaussian data

Euler discretization
with stepsize $\gamma > 0$

$$\hat{x}_0 \sim q_0, \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

$$v_{T-t}(x) = -\frac{\dot{\eta}_t}{\eta_t} x + (1+\alpha) \eta_t^2 \dot{\sigma}_t \sigma_t \nabla \log p_t(x)$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

$$\nabla \log p_t(x) = -(\eta_t^2(\Sigma_{\text{data}} + \sigma_t^2 \text{Id}))^{-1} x$$

- Linear velocity $\rightarrow$ Gaussianity preserved along the process $\text{Law}(\hat{x}_k) = \mathcal{N}(0, \text{Cov}(\hat{x}_k))$
- A_t diagonal in eigenbasis of $\Sigma_{\text{data}} \rightarrow \text{Cov}(\hat{x}_k)$ too

The final W_2 discretization error:

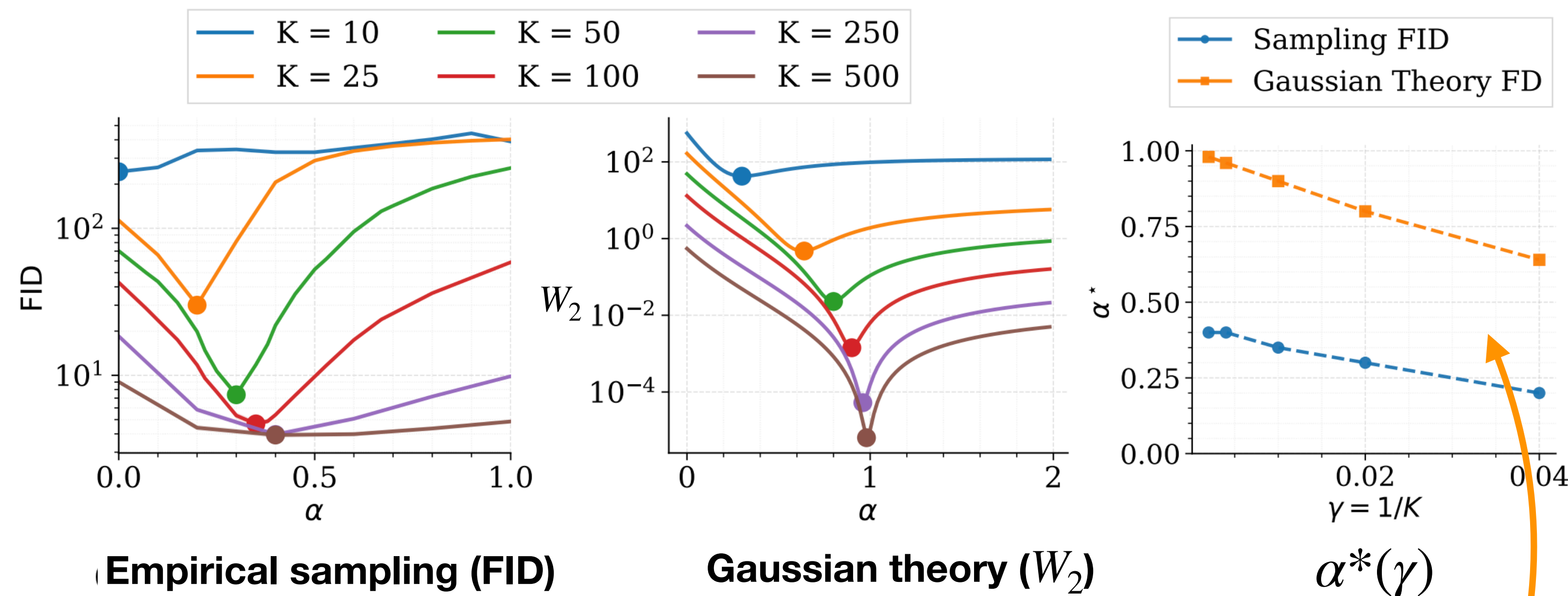
$$\begin{aligned} W_2(p_{\text{data}}, \text{Law}(\hat{x}_K))^2 &= B(\Sigma_{\text{data}}, \text{Cov}(\hat{x}_K))^2 = \sum_{i=1}^d \left(\sqrt{\lambda_i} - \sqrt{\lambda(\text{Cov}(\hat{x}_K))_i} \right)^2 \\ &= \underbrace{\gamma^2}_{\text{stepsize}} \sum_{i=1}^d \int_0^T \underbrace{f(\lambda_i, \alpha, \eta_t, \sigma_t)^2}_{\substack{\text{diffusion coefficient} \\ \text{\textit{i}^{th} eigenvalue of } \Sigma_{\text{data}}}} dt + o(\gamma^2) \end{aligned}$$

schedulers

✓ Explicit objective for optimizing parameters α, η_t, σ_t w.r.t data geometry $(\lambda_i)_{i \leq d}$

Optimization of the diffusion parameter α

Image dataset sampling (FFHQ) Sampling error w.r.t number of discretization steps K (stepsize $\gamma = 1/K$) :



Using the EDM schedule
[Karras et al., '23] $\eta_t = 1, \sigma_t = \sigma_{\max} t^\beta$

$\alpha^\star$ decreases ($\approx$ linearly)
with the stepsize γ

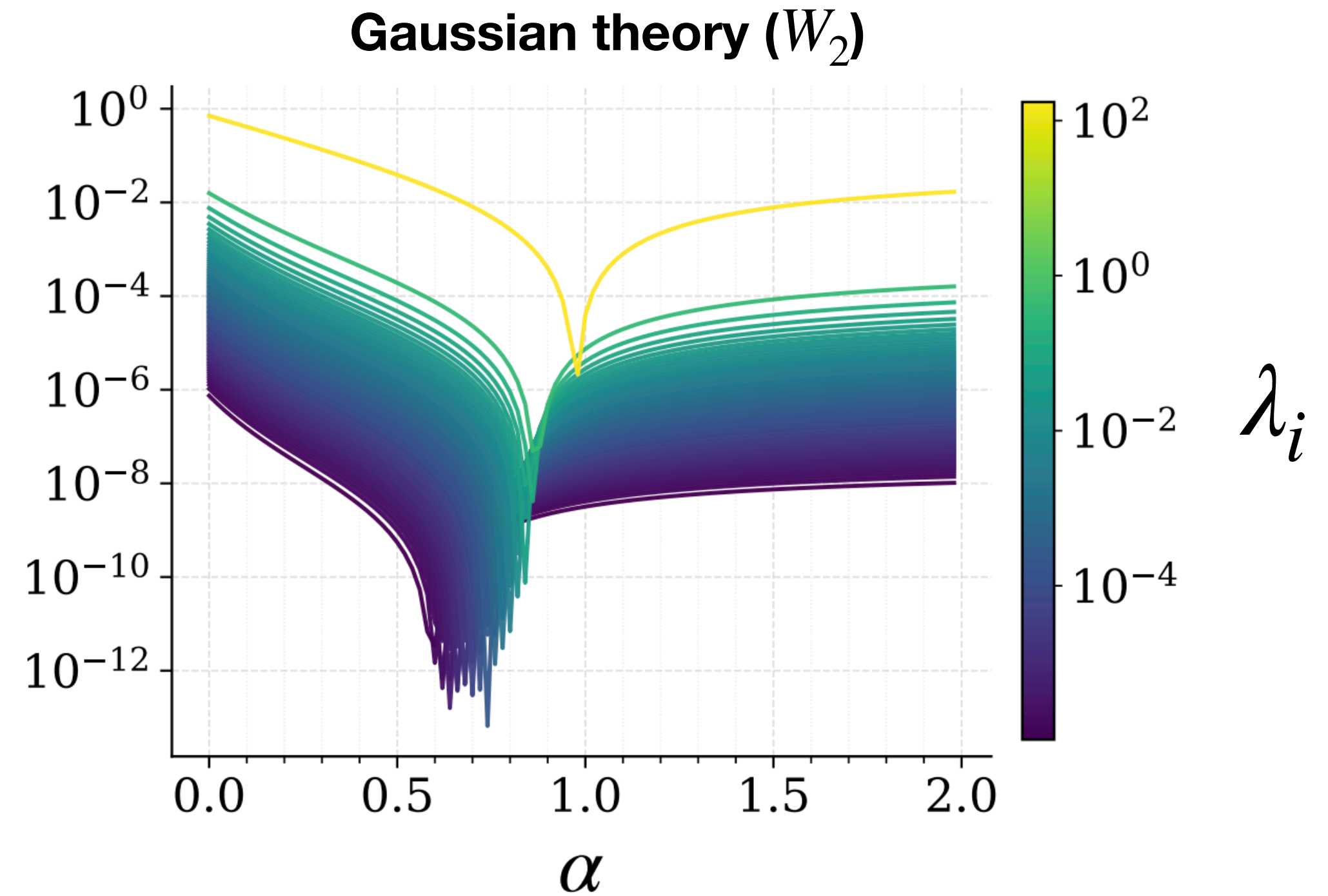
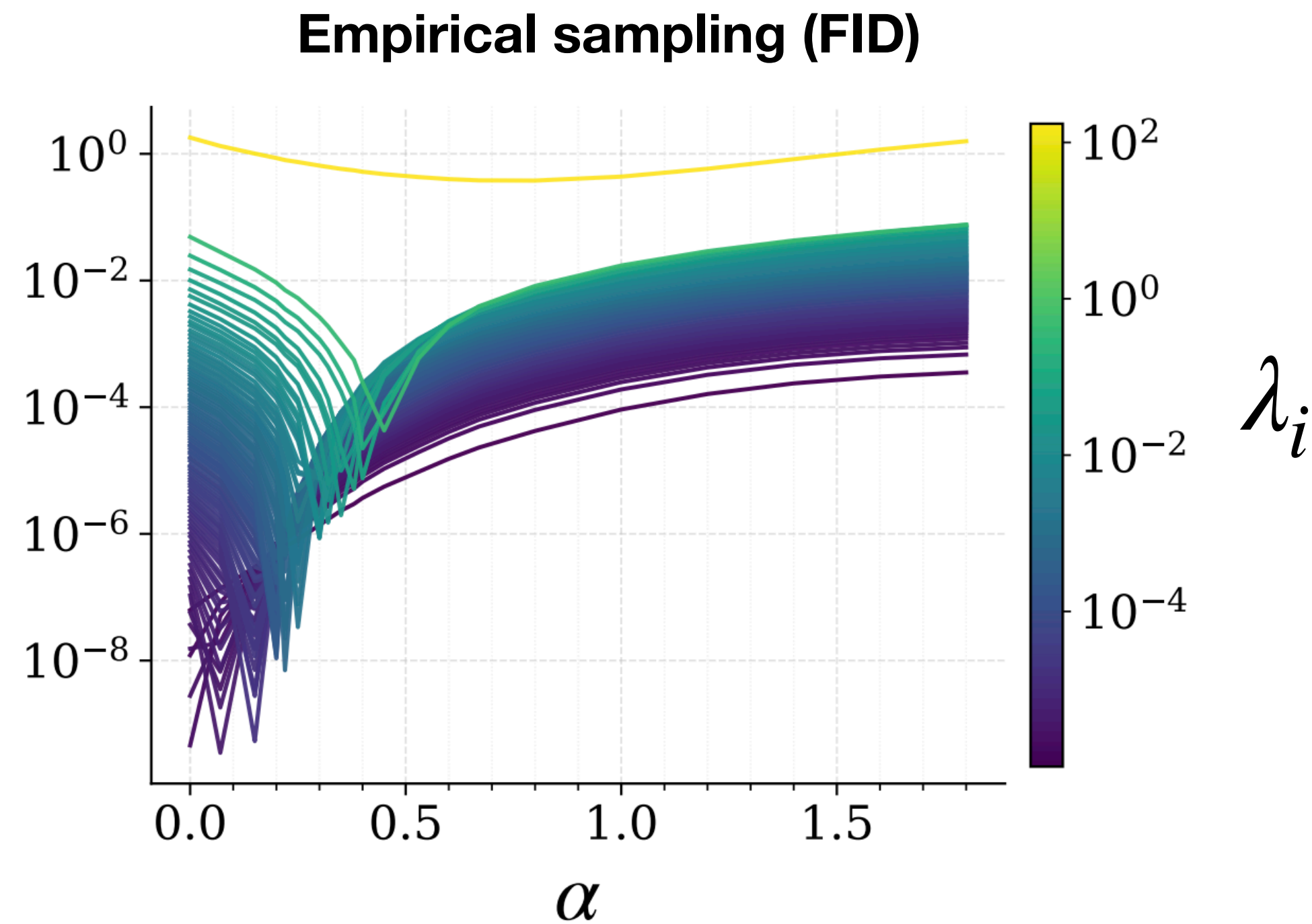
For Gaussian data,

$$\alpha^\star(\gamma) = \operatorname{argmin}_\alpha W_2(p_{\text{data}}, \text{Law}(\hat{Y}_K)) = 1 - \gamma \underbrace{\sum_{i=1}^d w_i \left(\frac{1}{\lambda_i} \right)^{\frac{1}{2\beta}}}_{\text{slope}} + o(\gamma)$$

The decrease slope depends on $(\lambda_i)_{i \leq d}$:
smaller variance $\lambda_i \rightarrow$ smaller $\alpha^\star$

Optimization of the diffusion parameter α

Image dataset sampling (FFHQ) W_2 error decomposed per eigendirection of Σ_{data} :



For *Gaussian data*,

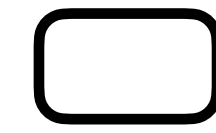
$$\alpha^\star(\gamma) = \operatorname{argmin}_{\alpha} W_2(p_{\text{data}}, \text{Law}(\hat{x}_K)) = 1 - \gamma \sum_{i=1}^d w_i \left(\frac{1}{\lambda_i} \right)^{\frac{1}{2\beta}} + o(\gamma)$$

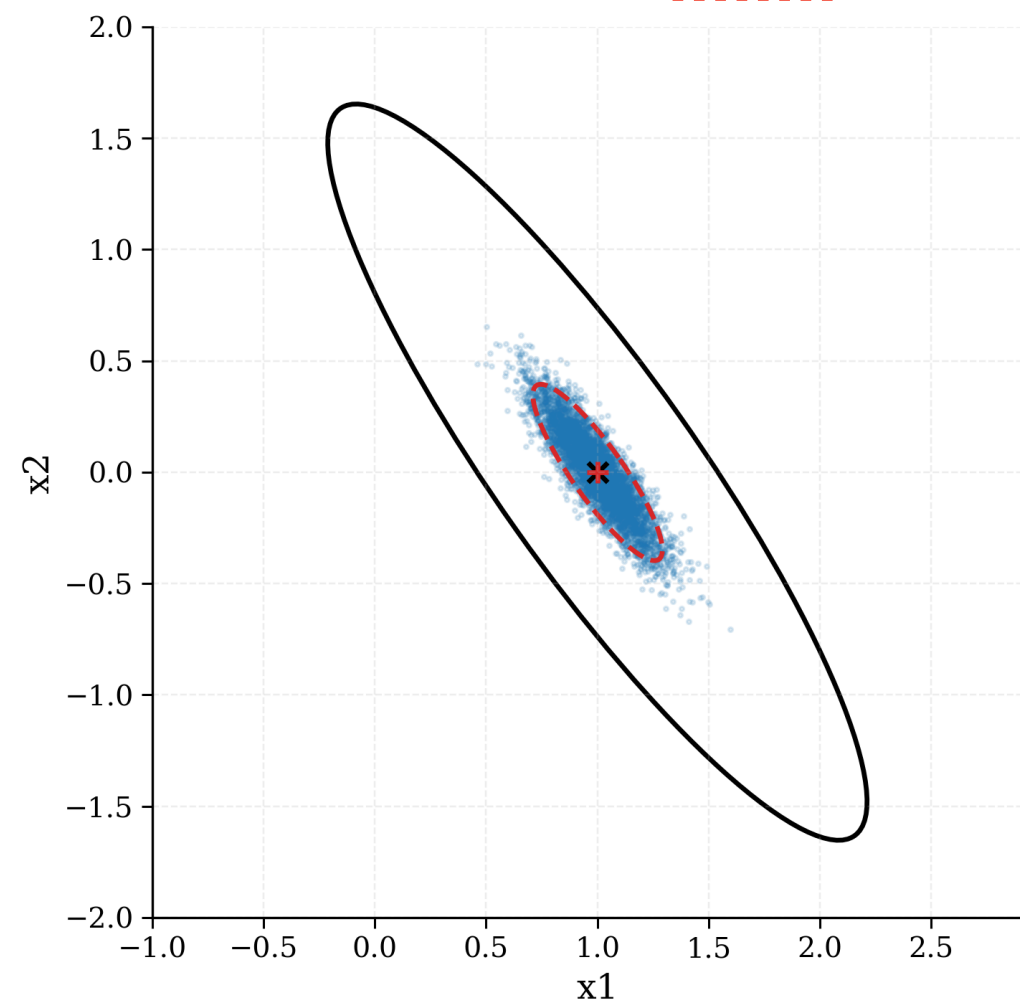
The decrease slope depends on $(\lambda_i)_{i \leq d}$:
smaller variance $\lambda_i \rightarrow$ smaller $\alpha^\star$

Optimization of the diffusion parameter α

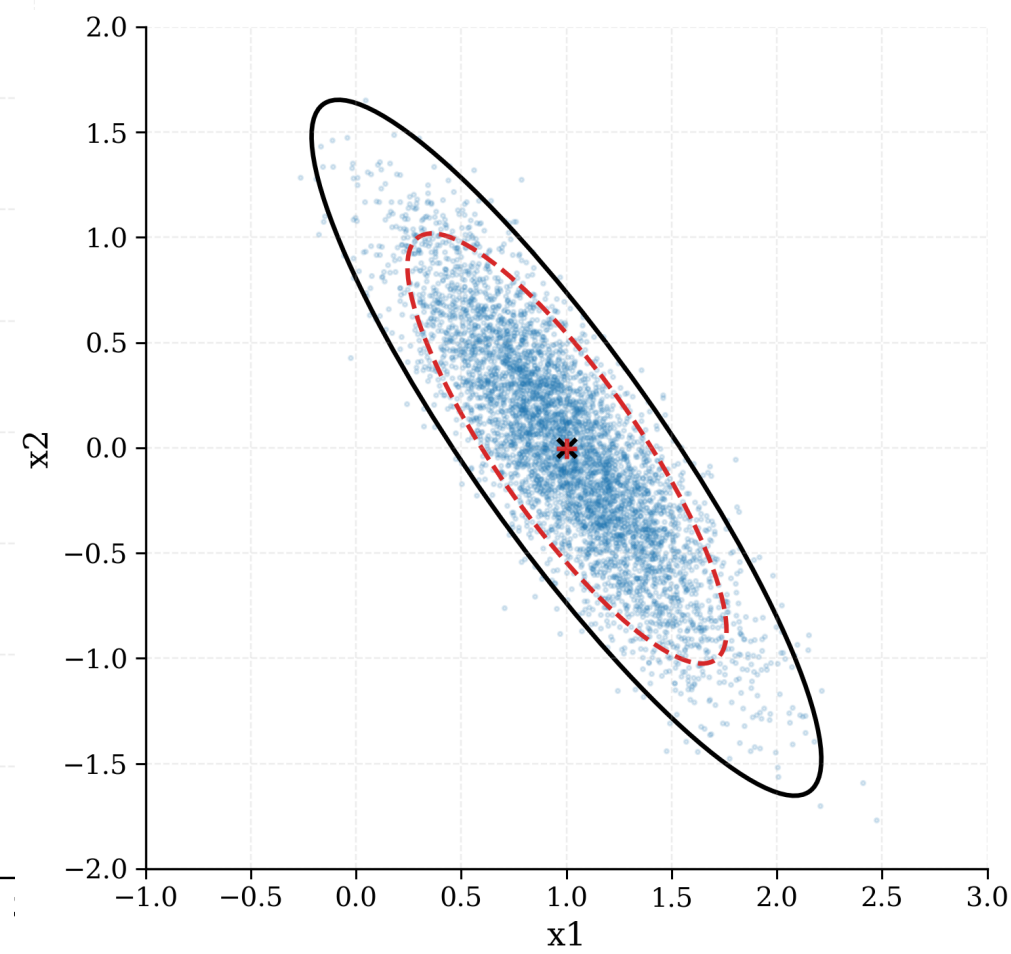
Gaussian sampling in 10 steps

 $\text{Law}(\hat{x}_k)$ covariance

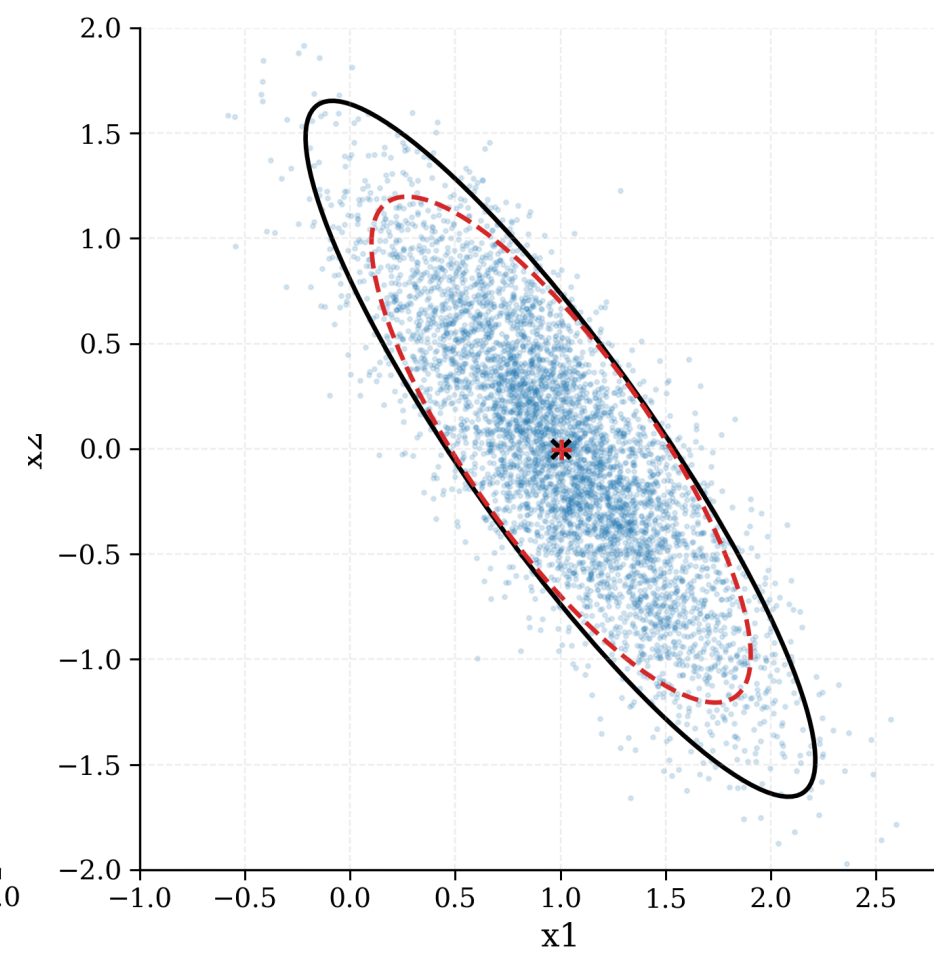
 p_{data} covariance



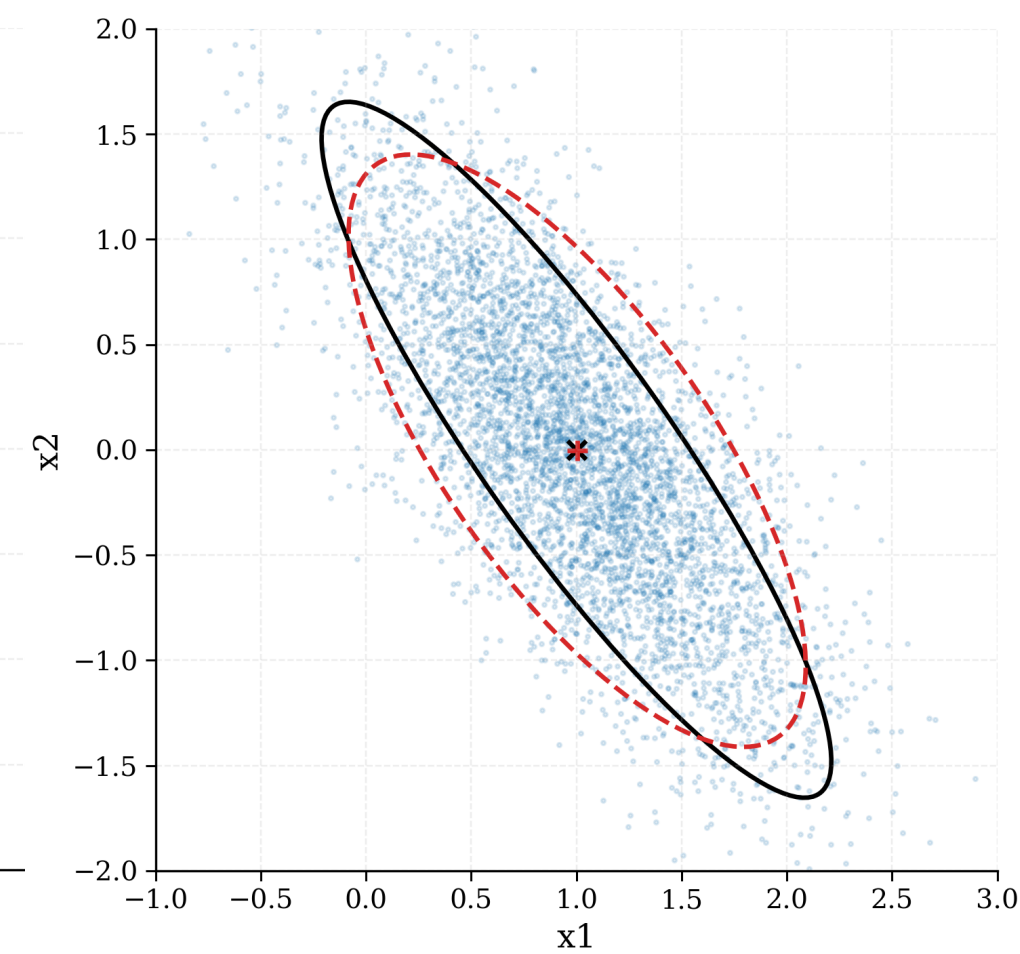
$\alpha = 0$



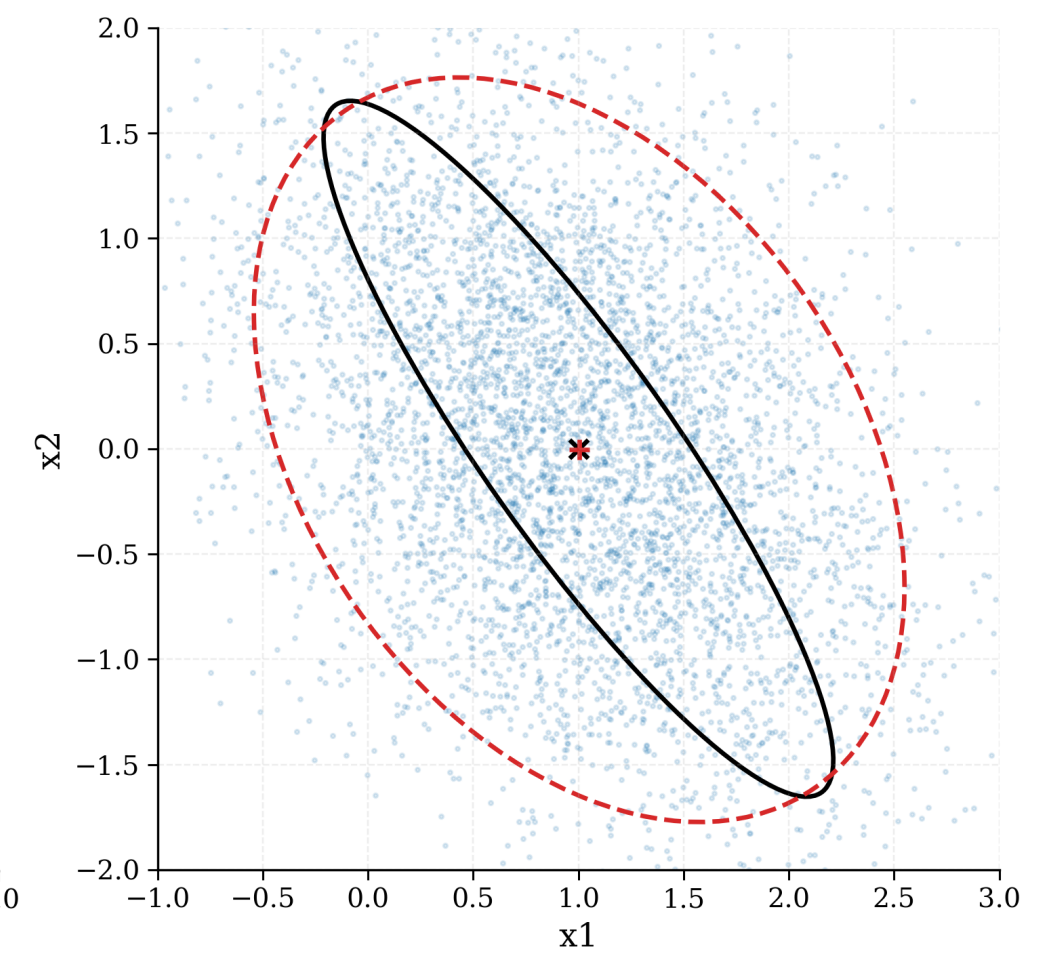
$\alpha = 0.25$



$\alpha = 0.5$



$\alpha = 0.75$



$\alpha = 1$

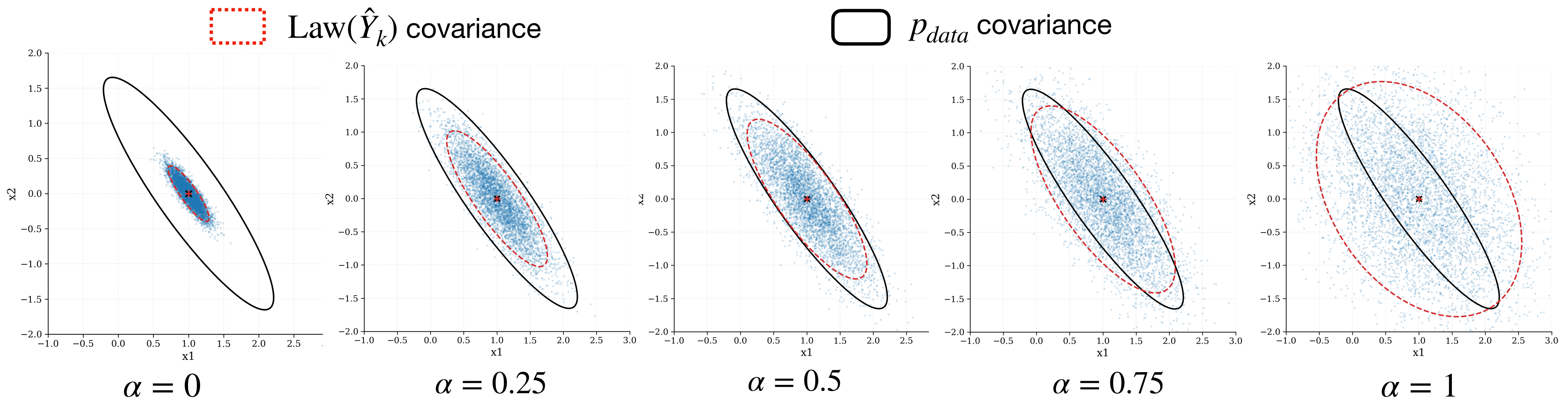
For *Gaussian data*,

$$\alpha^\star(\gamma) = \operatorname{argmin}_{\alpha} W_2(p_{\text{data}}, \text{Law}(\hat{x}_K)) = 1 - \gamma \sum_{i=1}^d w_i \left(\frac{1}{\lambda_i} \right)^{\frac{1}{2\beta}} + o(\gamma)$$

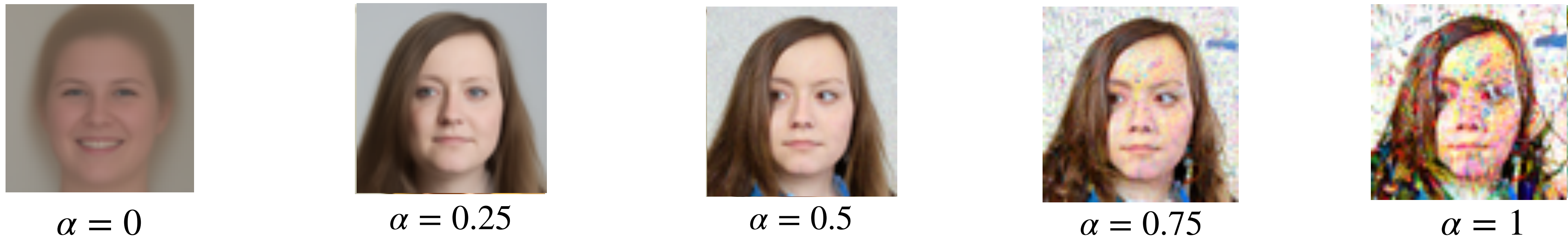
The decrease slope depends on $(\lambda_i)_{i \leq d}$:
smaller variance $\lambda_i \rightarrow$ smaller $\alpha^\star$

Optimization of the diffusion parameter α

Gaussian sampling in 10 steps



FFHQ sampling in 50 steps:



► For small number of steps, ODE sampling ($\alpha = 0$) samples close to the data mean

Optimization of the diffusion parameter α

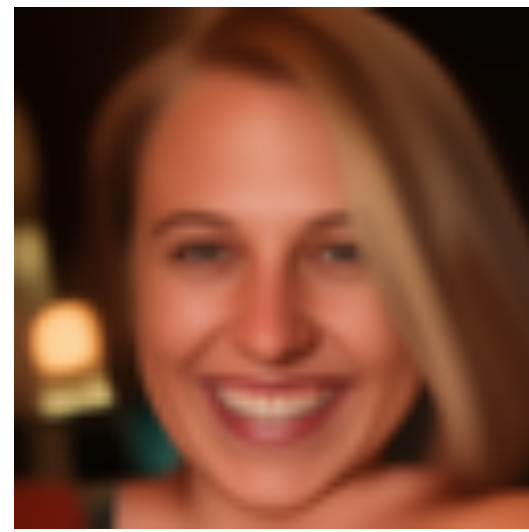
Posterior sampling: sample the posterior $p(x | y)$



x



$y = k \star x + w$



$\alpha = 0$



$\alpha = 0.25$



$\alpha = 0.5$

Discretization error allows
MMSE estimation using
diffusion sampling

▸ For small number of steps, choosing $\alpha = 0$ samples close to the **MMSE**

FFHQ sampling in 50 steps:



$\alpha = 0$



$\alpha = 0.25$



$\alpha = 0.5$



$\alpha = 0.75$



$\alpha = 1$

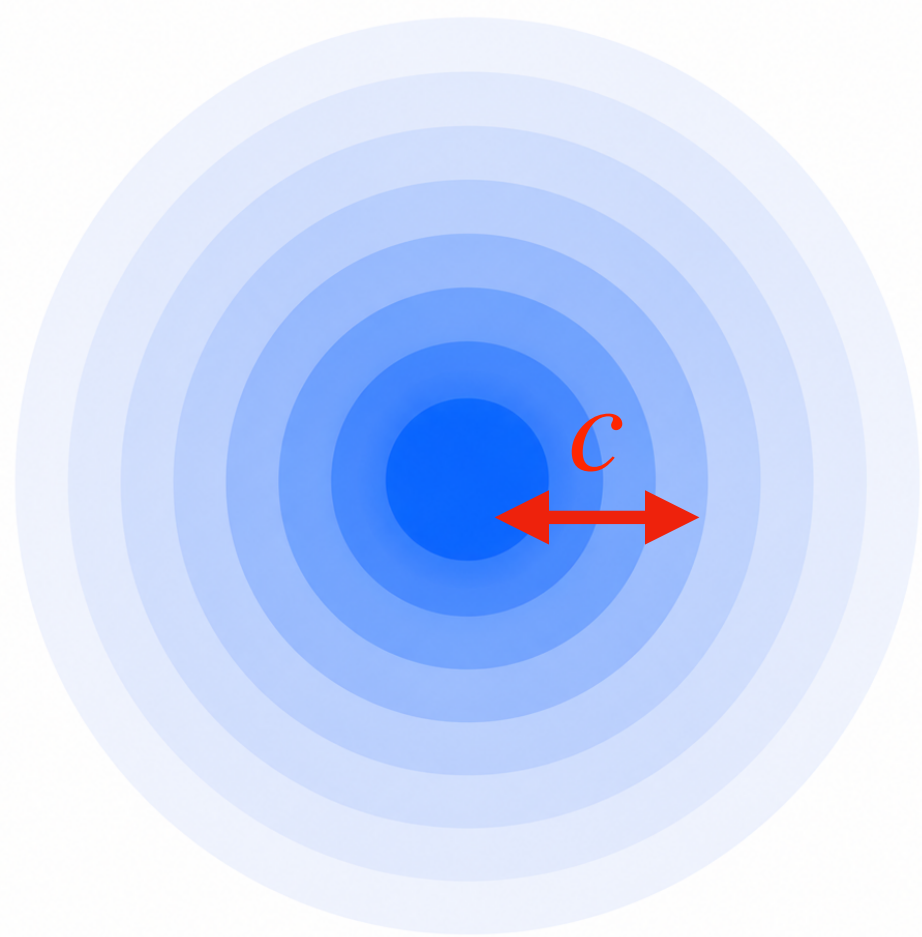
▸ For small number of steps, ODE sampling ($\alpha = 0$) samples close to the data mean

Optimization of the rescaling schedule η_t

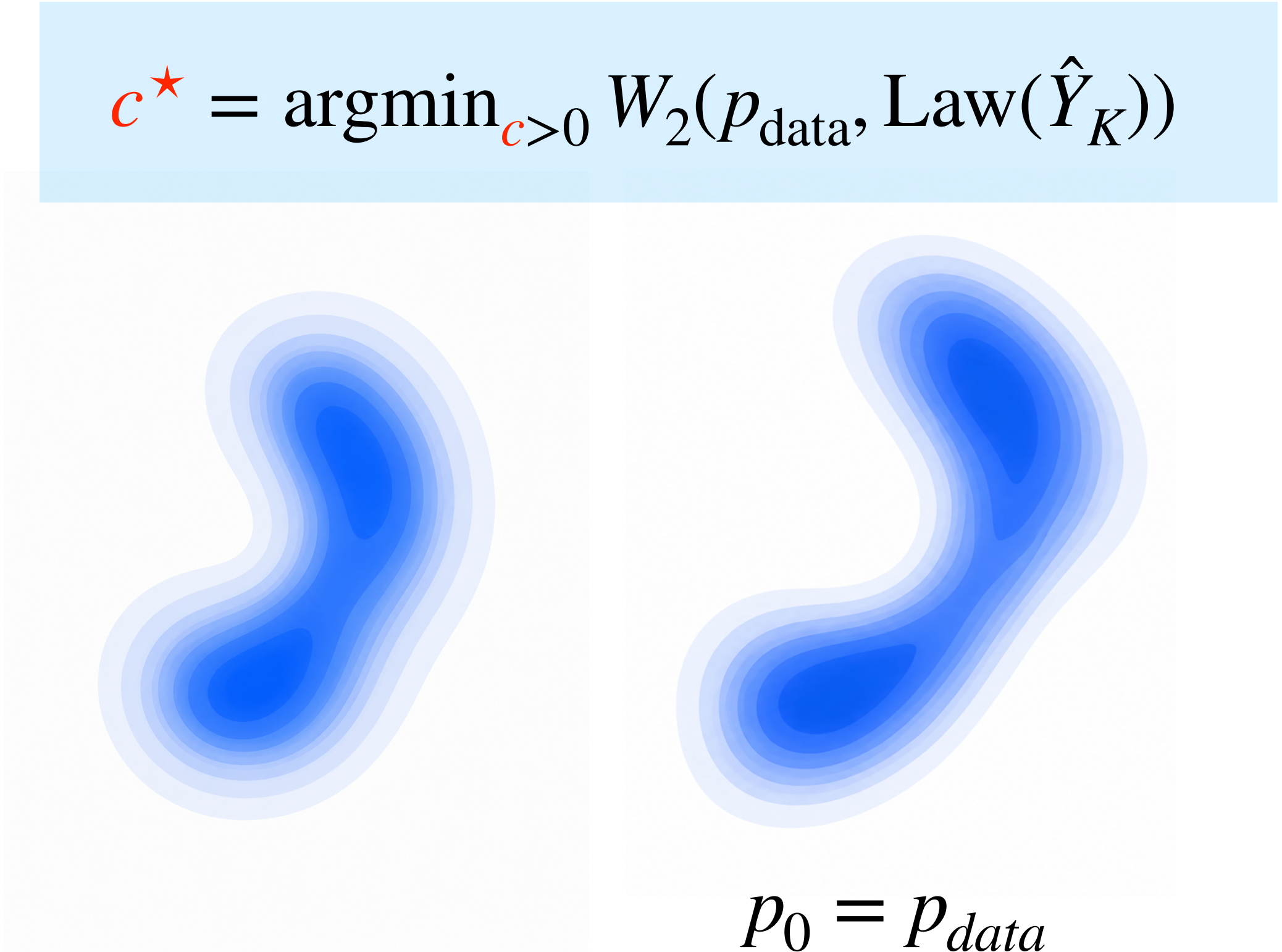
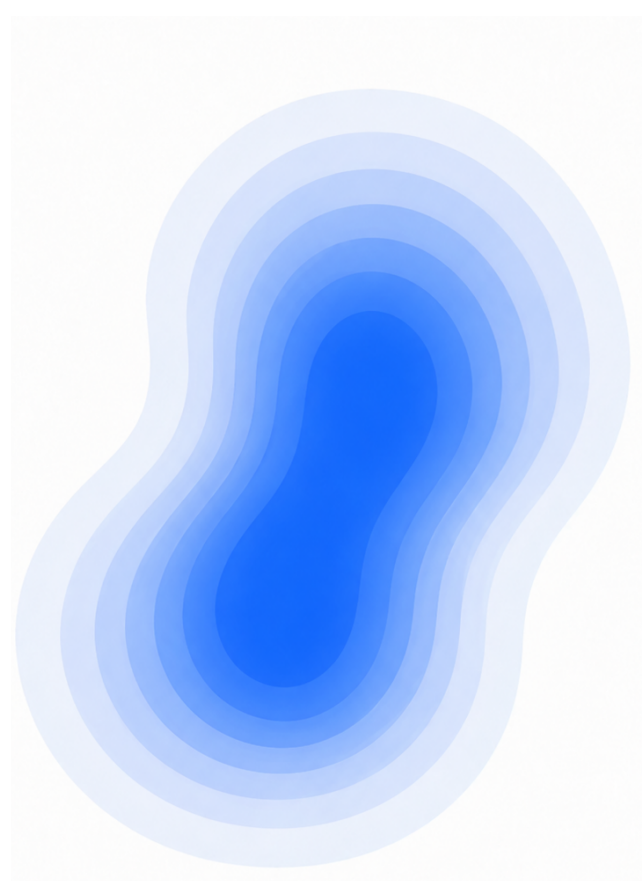
We choose the parametrization

$$\eta_t(c) = \sqrt{\frac{c}{c + \sigma_t^2}}$$

- Variance Preserving (VP) : $c = 1$
- Variance Exploding (VE) : $c = \infty$
- The **forward noising process** converges to $\mathcal{N}(0, c \text{I})$



$$p_T \approx \mathcal{N}(0, c \text{Id})$$



$$p_0 = p_{data}$$



Classical choices of rescaling schedules:

$$\eta_t^{VE} = 1, \quad \eta_t^{VP} = \sqrt{\frac{1}{1 + \sigma_t^2}}$$

$$c^* = \operatorname{argmin}_{c>0} W_2(p_{\text{data}}, \text{Law}(\hat{Y}_K))$$

Optimization of the rescaling schedule η_t

$$\eta_t(c) = \sqrt{\frac{c}{c + \sigma_t^2}}$$

For a power spectrum $\lambda_i = \lambda_{\max} i^{-p}$

$$c^\star = \lambda_{\max} \tau(p) \text{ with } \tau(p) > 0 \text{ decreasing for } p \lesssim 0.9$$

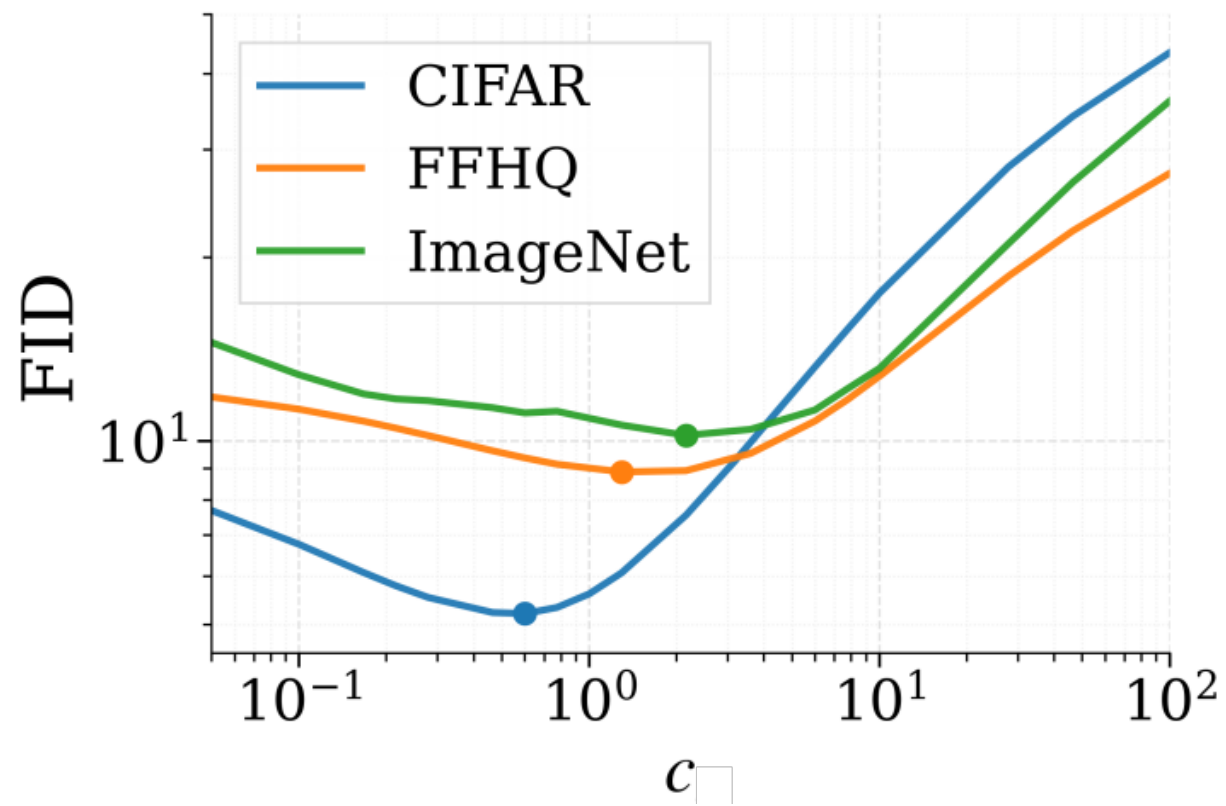
Full image datasets sampling

Small relative change of p
across datasets

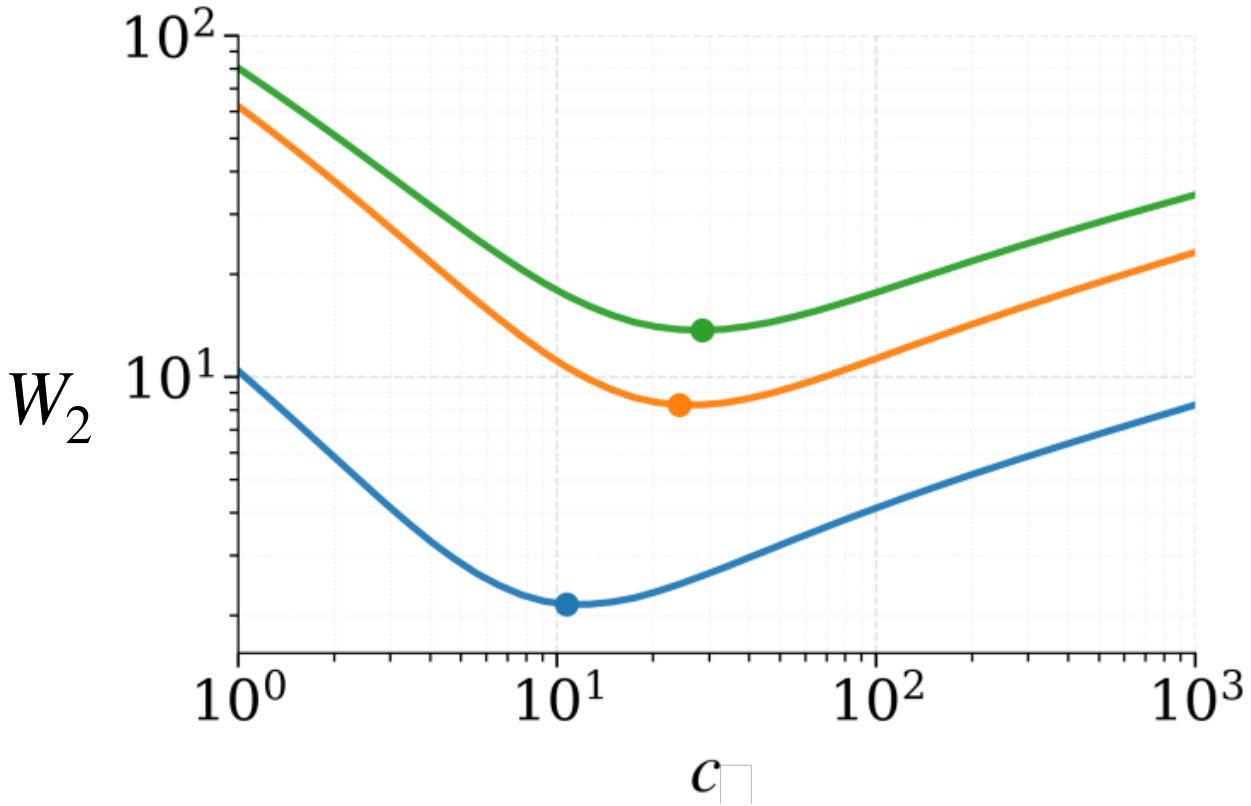
Influence of $\lambda_{\max}$

Dataset	$\lambda_{\max}$	p
CIFAR	55.3	1.43
FFHQ	173	1.47
ImageNet	225	1.35

Empirical sampling (FID)



Gaussian theory (W_2)



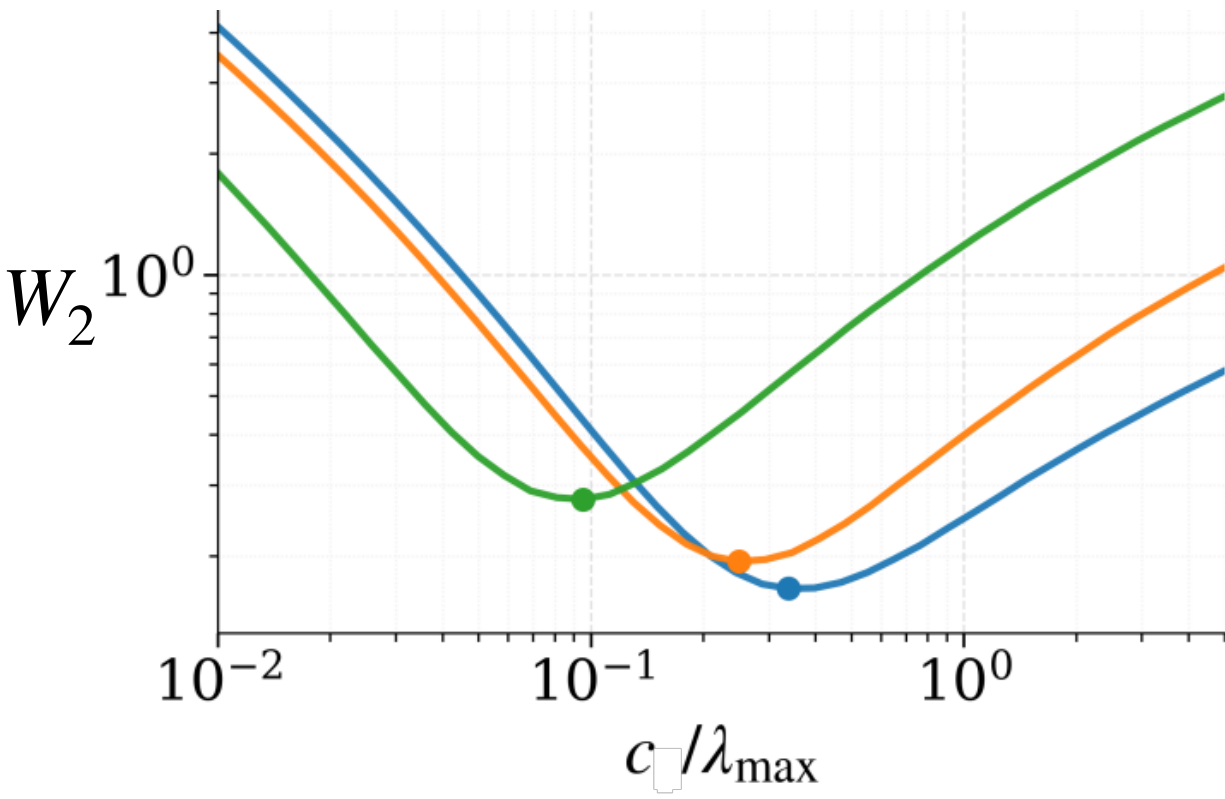
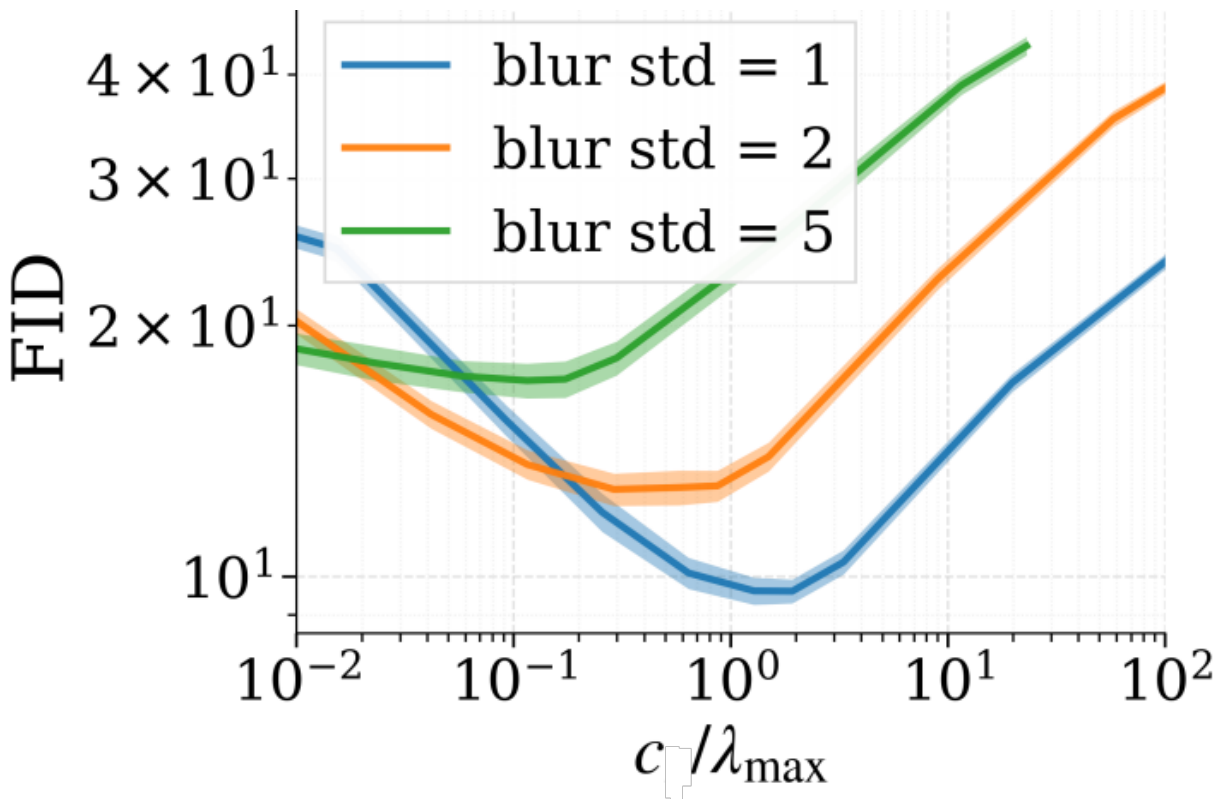
Deblurring Posterior sampling

Sampling of $p(x | y = k \star x + w)$

c -axis rescaled by $\lambda_{\max}$

Influence of p

blur std	$\lambda_{\max}$	p
1	0.39	0.57
2	0.86	0.61
5	0.78	0.83



Score Training error for Gaussian data

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

Exact linear score: $\nabla \log p_t(x) = A_t^* x$

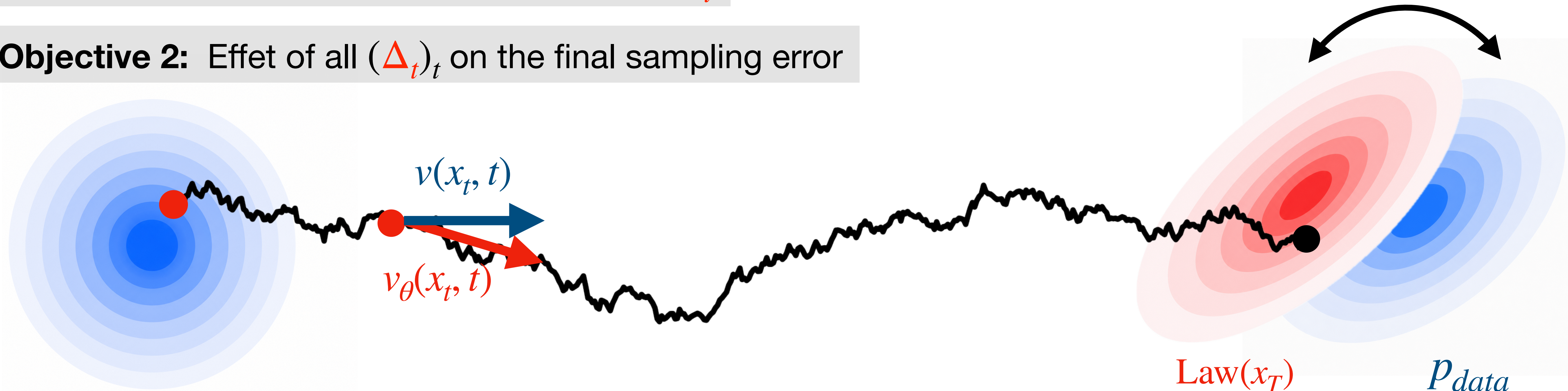
Trained linear score: $s_t^\theta(x) = A_t x = (A_t^* + \Delta_t) x$

Perturbation Δ_t due to inexact training:

- generalization error
- optimization error

Objective 1: At fixed t explicit training error Δ_t

Objective 2: Effect of all $(\Delta_t)_t$ on the final sampling error



Score Training error for Gaussian data

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

Exact linear score: $\nabla \log p_t(x) = A_t^* x$

Trained linear score: $s_t^\theta(x) = A_t x = (A_t^* + \Delta_t) x$

Perturbation Δ_t due to inexact training:

- generalization error
- optimization error

Objective 1: At fixed t explicit training error Δ_t

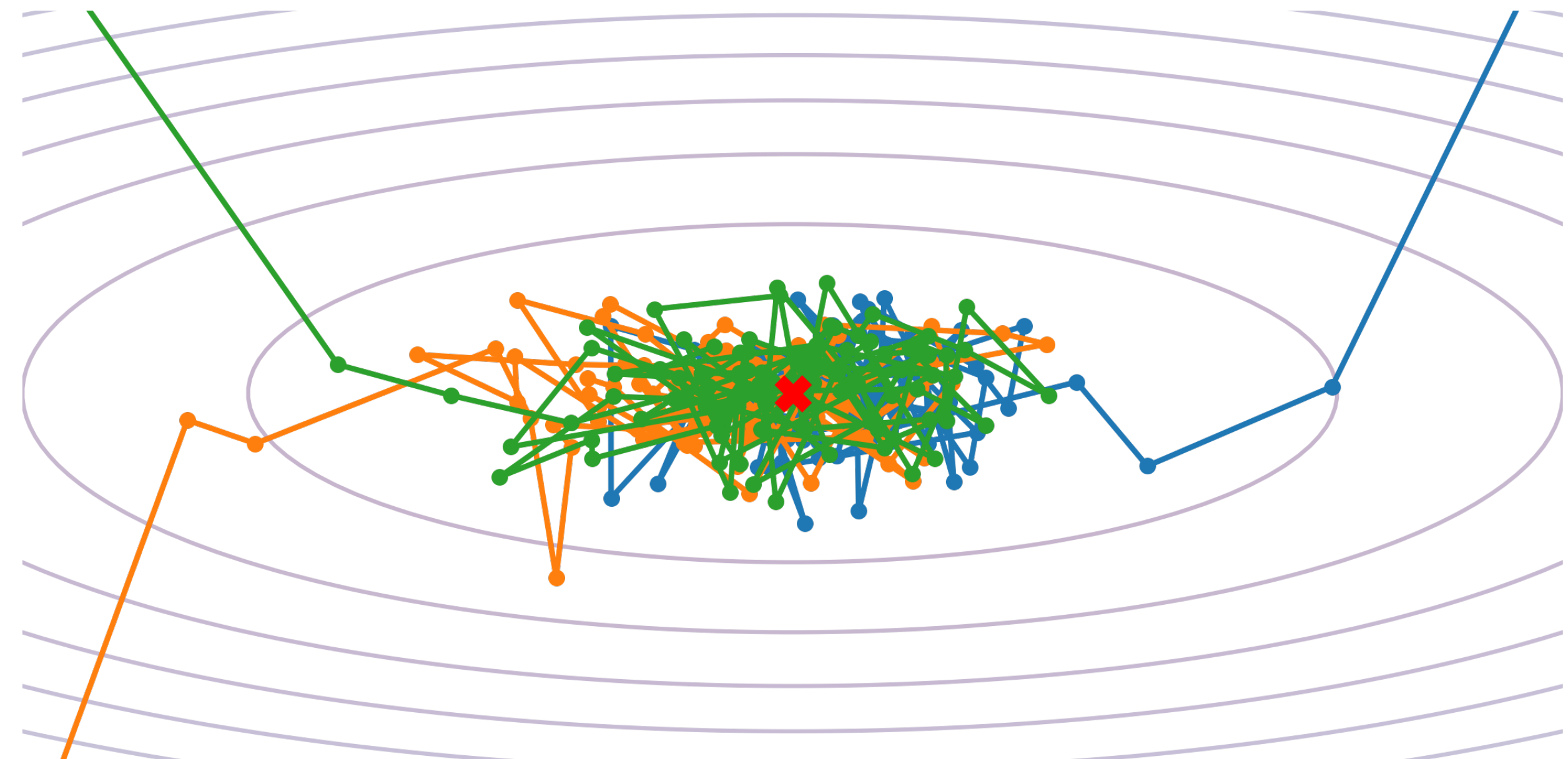
Minimize $F(\theta) = \mathbb{E}_{x \sim p_{\text{data}}, w \sim \mathcal{N}(0, I)} [f(\theta, x, w)]$
on a finite dataset $X_{N_{\text{data}}} = \{x_i \sim p_{\text{data}}\}_{0 \leq i < N_{\text{data}}}$

SGD with fixed learning rate τ :

For $k \geq 0$, $x_k \in X_{N_{\text{data}}}$, $w_k \sim \mathcal{N}(0, \text{Id})$,

$$\theta_{k+1} = \theta_k - \tau \nabla_\theta f(\theta_k, x_k, w_k)$$

[Dieuleveut et al., '17]



Score Training error for Gaussian data

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_t(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

Exact linear score: $\nabla \log p_t(x) = A_t^* x$

Trained linear score: $s_t^\theta(x) = A_t x = (A_t^* + \Delta_t) x$

Perturbation Δ_t due to inexact training:

- generalization error
- optimization error

Objective 1: At fixed t explicit training error Δ_t

Minimize $F(\theta) = \mathbb{E}_{x \sim p_{\text{data}}, w \sim \mathcal{N}(0, I)} [f(\theta, x, w)]$
on a finite dataset $X_{N_{\text{data}}} = \{x_i \sim p_{\text{data}}\}_{0 \leq i < N_{\text{data}}}$

SGD with fixed learning rate τ :

For $k \geq 0$, $x_k \in X_{N_{\text{data}}}$, $w_k \sim \mathcal{N}(0, \text{Id})$,
 $\theta_{k+1} = \theta_k - \tau \nabla_\theta f(\theta_k, x_k, w_k)$

[Dieuleveut et al., '17]

With a **linear score** $s_t^\theta(x) = A_t x$: $A_t = A_t^* + \Delta_t$

$$\mathbb{E}[\Delta_t] = o\left(\tau, \frac{1}{N_{\text{data}}}\right)$$

$$\text{Cov}(\Delta_t) = \tau \Sigma_t^\tau + \frac{1}{N_{\text{data}}} \Sigma_t^N + o\left(\tau, \frac{1}{N_{\text{data}}}\right)$$

Σ_t^τ and Σ_t^N explicit w.r.t $\Sigma_{\text{data}}, \alpha, \eta_t, \sigma_t$

Score Training error for Gaussian data

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_t(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

Exact linear score: $\nabla \log p_t(x) = A_t^* x$

Trained linear score: $s_t^\theta(x) = A_t x = (A_t^* + \Delta_t) x$

Perturbation Δ_t due to inexact training:

- generalization error
- optimization error

Objective 1: At fixed t explicit training error Δ_t

Objective 2: Effect of all $(\Delta_t)_t$ on the final sampling error

$$\text{Cov}(\Delta_t) = \tau \Sigma_t^\tau + \frac{1}{N_{\text{data}}} \Sigma_t^N + o\left(\tau, \frac{1}{N_{\text{data}}}\right)$$

Score Training error for Gaussian data

$$x_0 \sim q_0, \quad dx_t = v_t^\theta(x_t) dt + \sqrt{2\alpha a_t} dW_t$$

$$v_t(x) = f(\alpha, \eta_t, \sigma_t \nabla \log p_t(x))$$

Assuming Gaussian data $p_{\text{data}} = \mathcal{N}(0, \Sigma_{\text{data}} = U \text{Diag}(\lambda_i) U^\top)$

Exact linear score: $\nabla \log p_t(x) = A_t^* x$

Trained linear score: $s_t^\theta(x) = A_t x = (A_t^* + \Delta_t) x$

Perturbation Δ_t due to inexact training:

- generalization error
- optimization error

Objective 1: At fixed t explicit training error Δ_t

Objective 2: Effect of all $(\Delta_t)_t$ on the final sampling error

$$\text{Cov}(\Delta_t) = \tau \Sigma_t^\tau + \frac{1}{N_{\text{data}}} \Sigma_t^N + o\left(\tau, \frac{1}{N_{\text{data}}}\right)$$

- Linear drift $v_t = A_t x \rightarrow$ Gaussianity preserved along the process
- A_t **does not** co-diagonalizes with $\Sigma_{\text{data}} \rightarrow \text{Cov}(x_t)$ neither

where $\mathcal{E}^{\text{optim}}$ and $\mathcal{E}^{\text{gen}}$ take the form

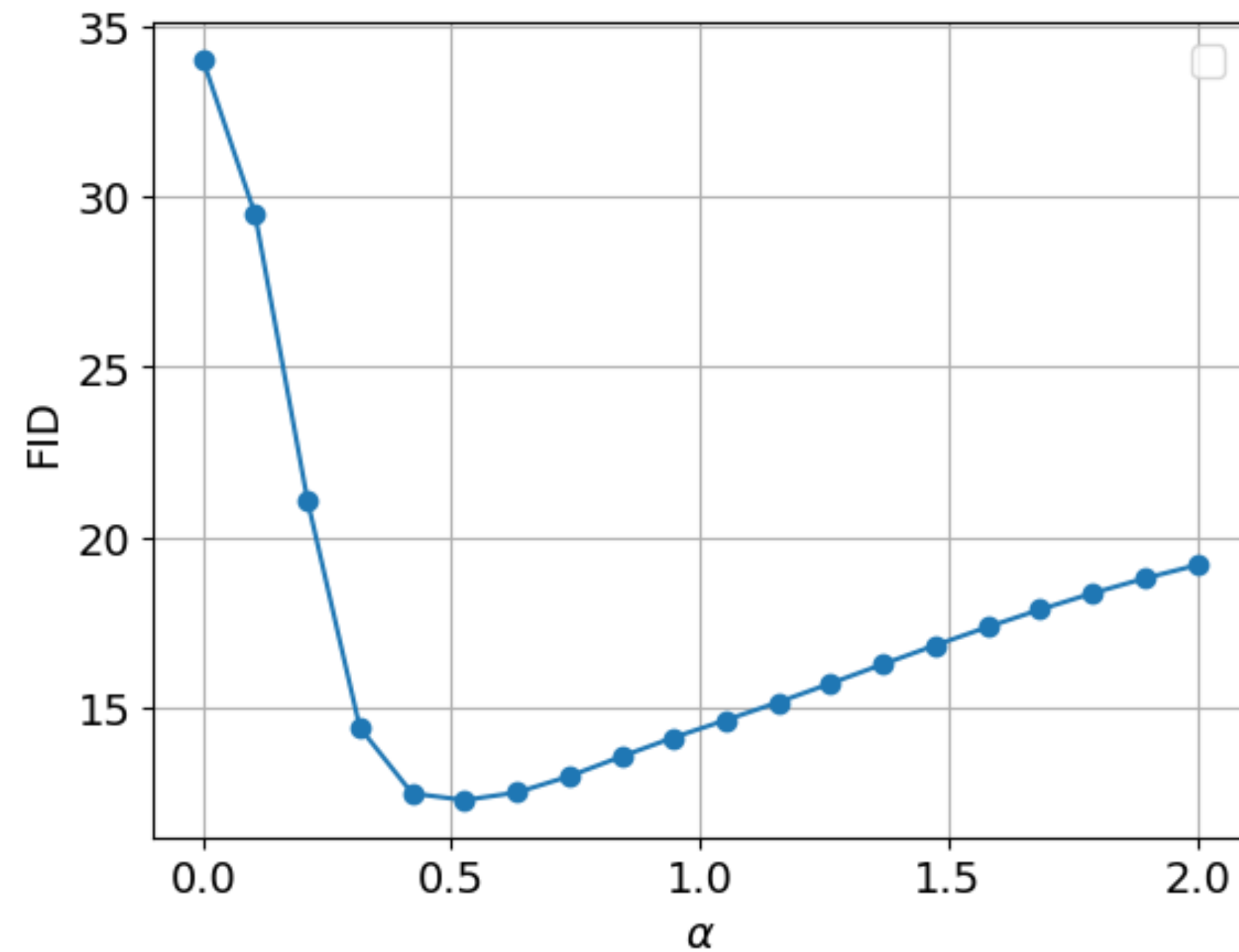
$$\mathbb{E}_\Delta \left[W_2(p_{\text{data}}, \text{Law}(x_T))^2 \right] = \left(\tau \mathcal{E}^{\text{optim}} + \frac{1}{N_{\text{data}}} \mathcal{E}^{\text{gen}} \right) + o\left(\tau, \frac{1}{N_{\text{data}}}\right)$$

$$\sum_{i=1}^d \sum_{j=1}^d \int_0^T f(\lambda_i, \lambda_j, \alpha, \sigma_s) ds$$

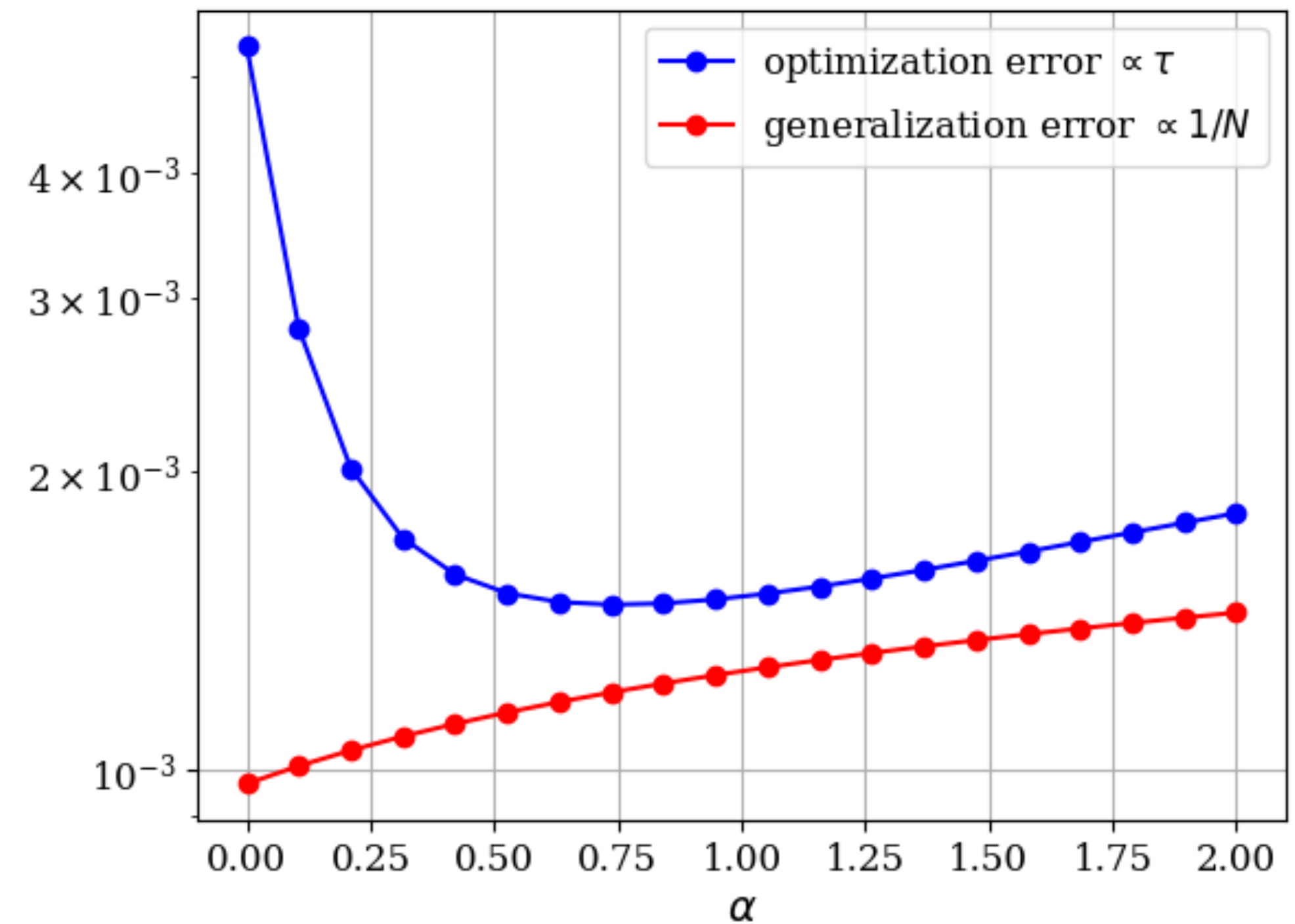
Optimization of the diffusion parameter α

$$\mathbb{E}_{\Delta} \left[W_2(p_{data}, \text{Law}(x_T))^2 \right] = \left(\tau \mathcal{E}^{optim} + \frac{1}{N_{data}} \mathcal{E}^{gen} \right) + o \left(\tau, \frac{1}{N_{data}} \right)$$

For $\sigma_t = t$, $\alpha^*(\mathcal{E}_t^{optim}) \approx \sqrt{\frac{\lambda_{max}}{2 + \lambda_{max}}} \in (0,1)$ and $\alpha^*(\mathcal{E}_t^{gen}) = 0$



Empirical sampling (FID)



Gaussian theory (W_2)

Conclusion

Data geometry matters for optimizing diffusion parameters w.r.t **discretization error**:

- Derived explicit **discretization** and **score training** errors of diffusion sampling.
- In Gaussian case, gives a **tractable objective** for tuning parameters.
- Study how the optimal **diffusion parameter** α and schedulers η_t, σ_t change with the shape of the **data covariance spectrum** $(\lambda_i)_{i \leq d}$.
- Gaussian theory predicts trends observed for **real image sampling**.

Generative models

given samples of p_{data} , generate new samples of p_{data}

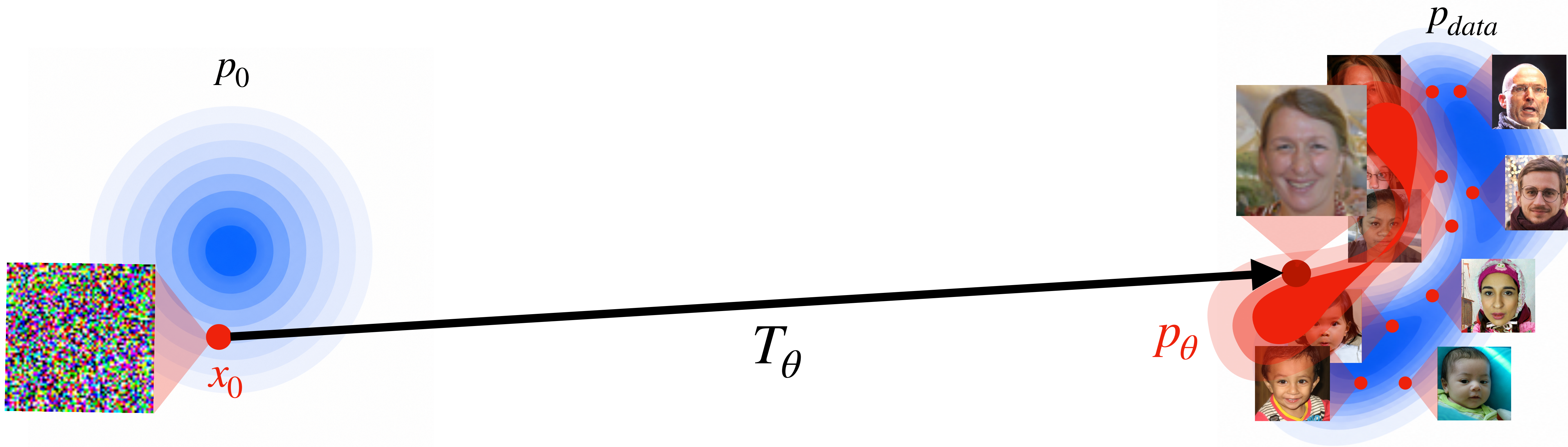
Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

Train a neural network T_θ to minimize

$$\min_{\theta} D(p_{data}, p_\theta)$$



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

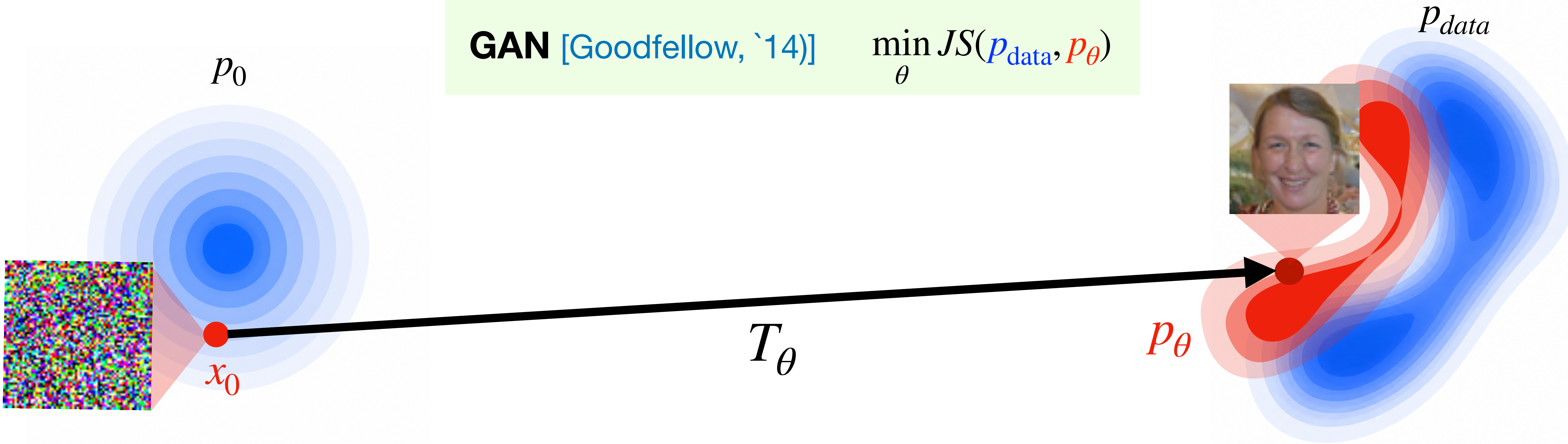
$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

Train a neural network T_θ to minimize

$$\min_{\theta} D(p_{data}, p_\theta)$$

GAN [Goodfellow, '14)]

$$\min_{\theta} JS(p_{data}, p_\theta)$$



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

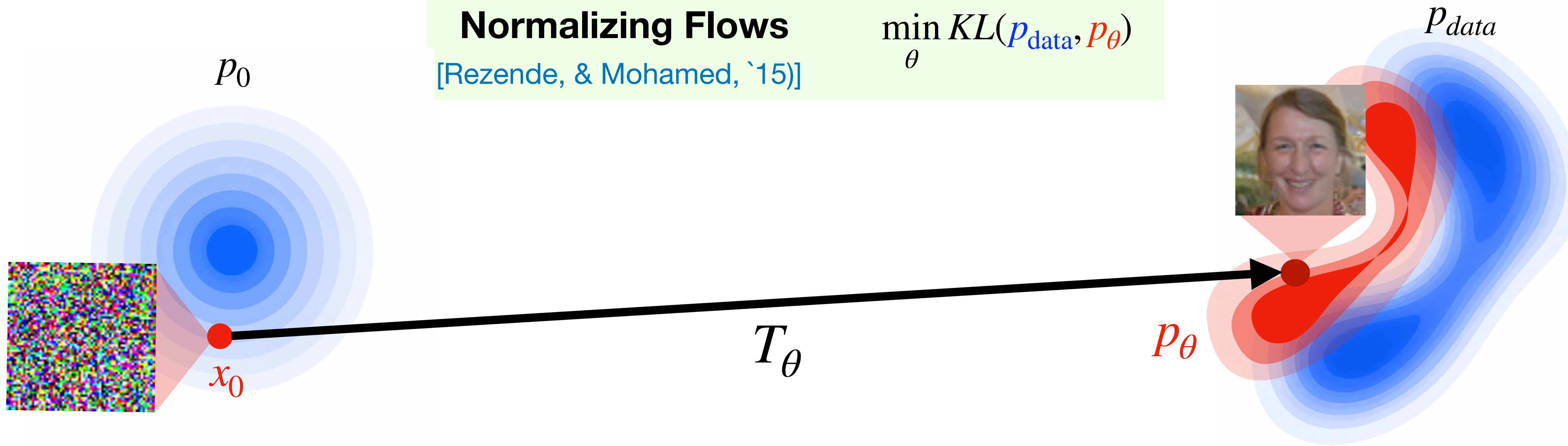
Train a neural network T_θ to minimize

$$\min_{\theta} D(p_{data}, p_\theta)$$

Normalizing Flows

[Rezende, & Mohamed, '15)]

$$\min_{\theta} KL(p_{data}, p_\theta)$$



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

Train a neural network T_θ to minimize

$$\min_{\theta} D(p_{data}, p_\theta)$$

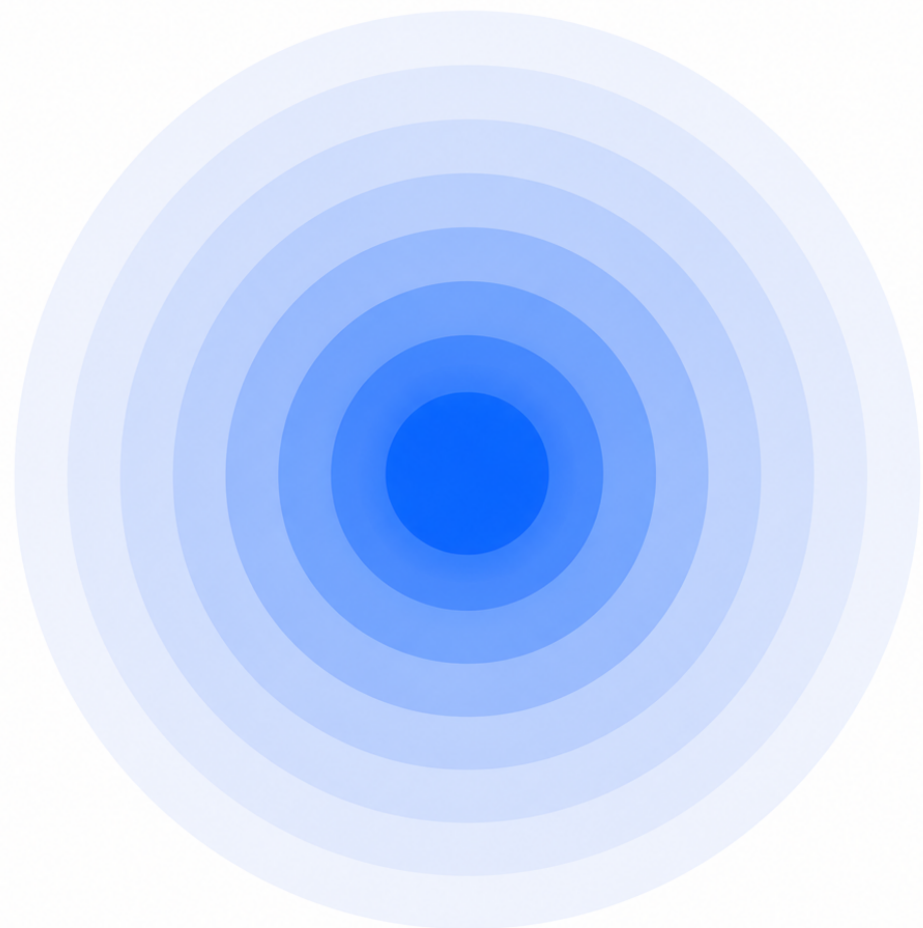
Normalizing Flows

[Rezende, & Mohamed, '15)]

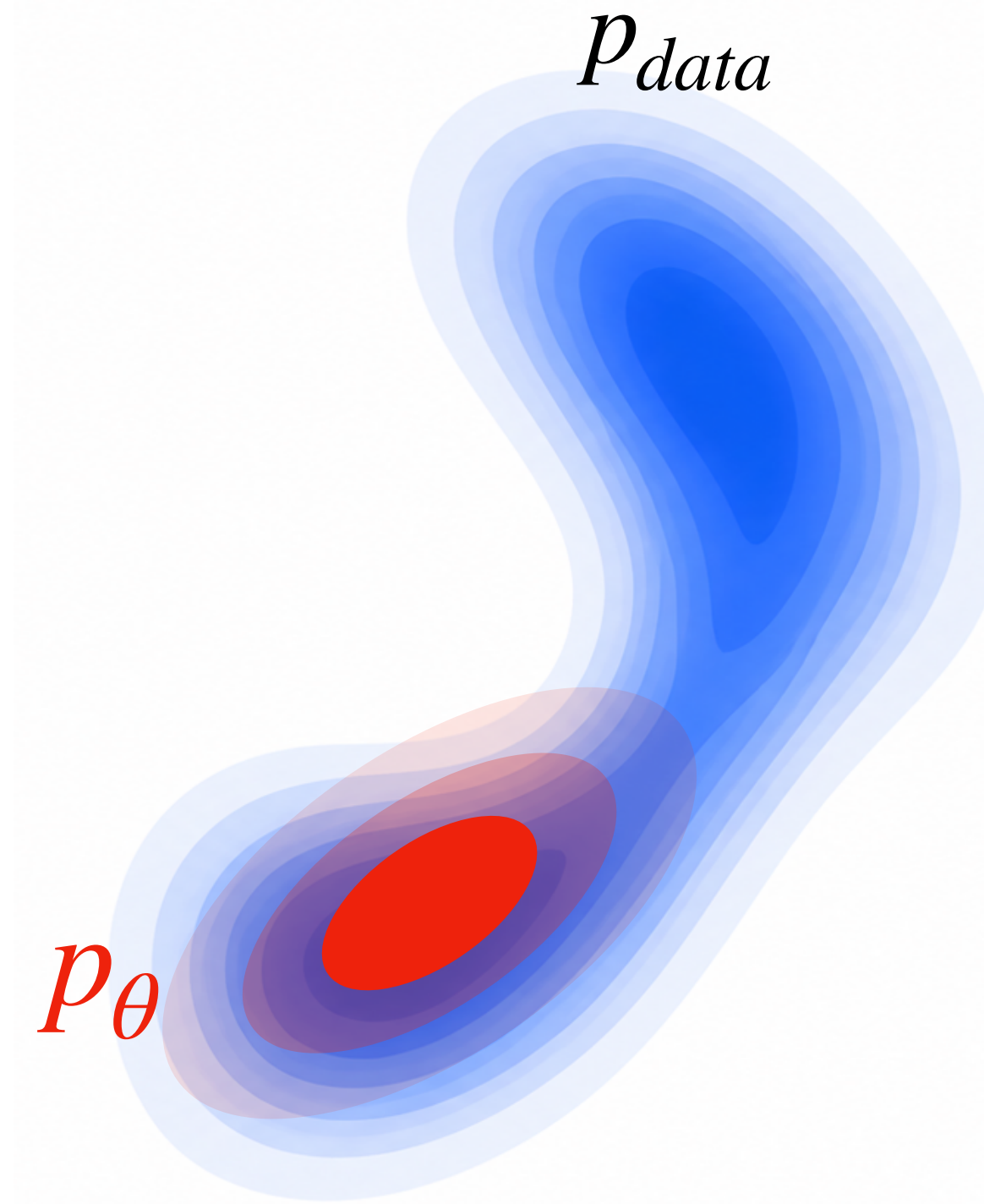
$$\min_{\theta} KL(p_{data}, p_\theta)$$

✗ Hard to train a single-step model:
unstable training, mode collapse...

p_0



p_{data}



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

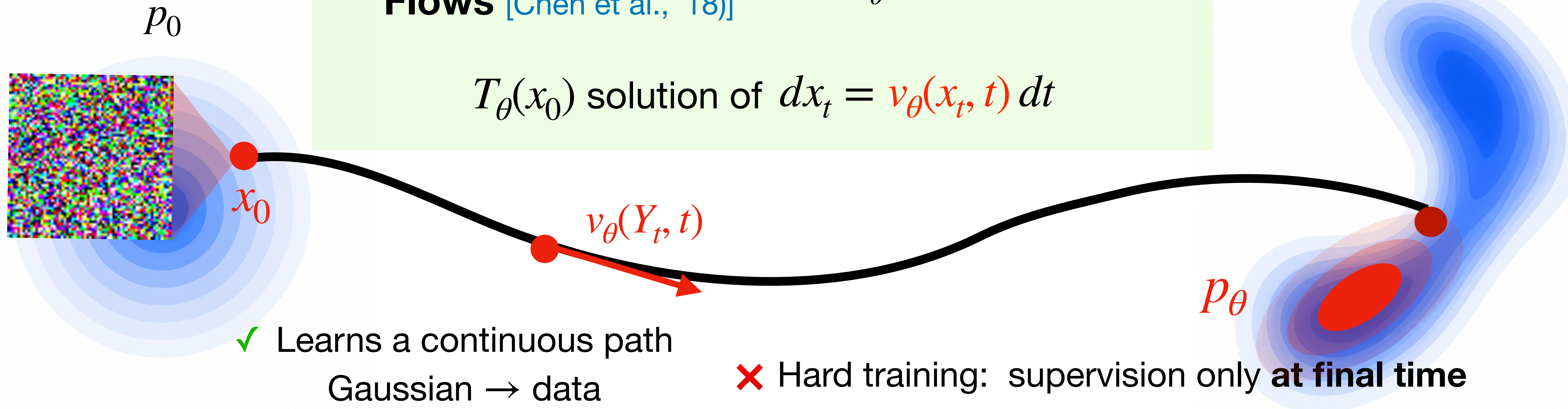
Train a neural network T_θ to minimize

$$\min_{\theta} D(p_{data}, p_\theta)$$

**Continuous Normalizing
Flows** [Chen et al., '18]

$$\min_{\theta} KL(p_{data}, p_\theta)$$

$$T_\theta(x_0) \text{ solution of } dx_t = v_\theta(x_t, t) dt$$



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

- Choose p_0 easy to sample
- Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that
 $x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

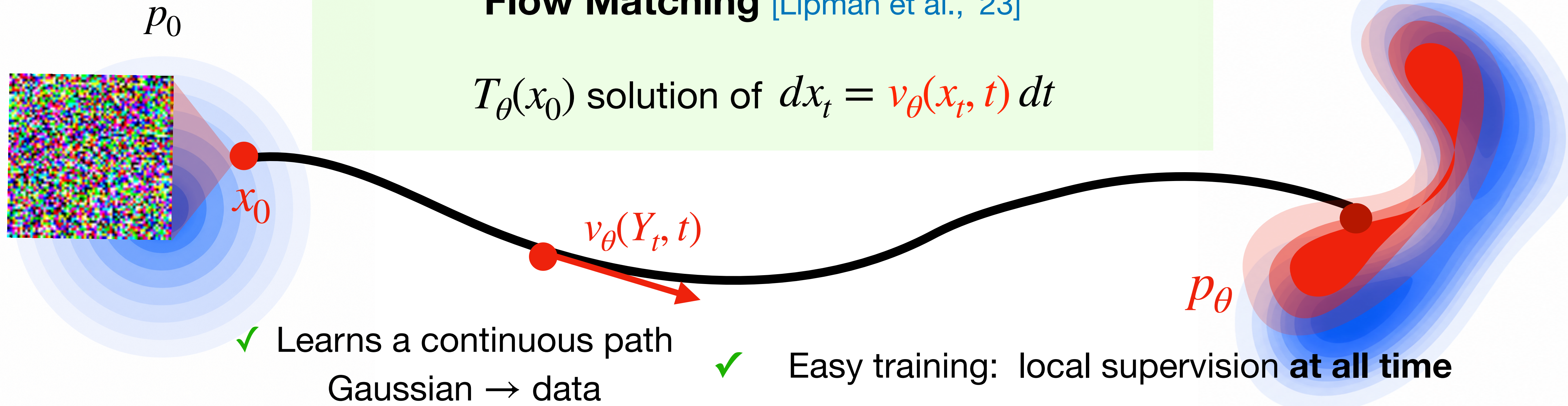
$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

Train a time-dependent NN v_θ to minimize

$$\min_{\theta} \int_t \|v_\theta(x, t) - v^*(x, t)\|$$

Diffusion Models [Song et al., '21] /
Flow Matching [Lipman et al., '23]

$$T_\theta(x_0) \text{ solution of } dx_t = v_\theta(x_t, t) dt$$



Generative models

given samples of p_{data} , generate new samples of p_{data}

Flow-based methods:

→ Choose p_0 easy to sample

→ Find a map $T_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that

$x_0 \sim p_0, x = T_\theta(x_0)$ samples p_{data}

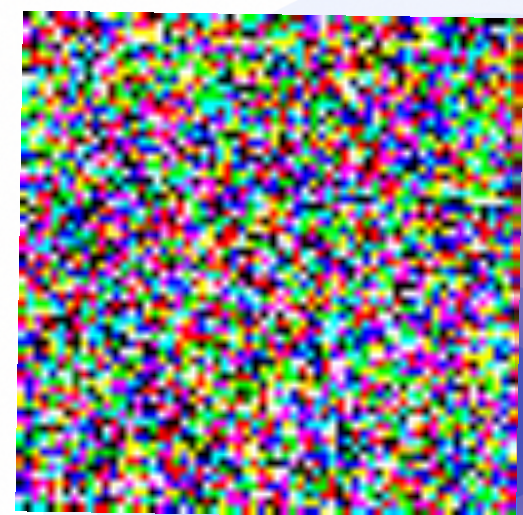
$$p_\theta := T_\theta \# p_0 \approx p_{data}$$

Train a time-dependent NN v_θ to minimize

$$\min_{\theta} \int_t \|v_\theta(x, t) - v^*(x, t)\|$$

Diffusion Models [Song et al., '21] /
Flow Matching [Lipman et al., '23]

$$T_\theta(x_0) \text{ solution of } dx_t = v_\theta(x_t, t) dt + \sqrt{2a_t} dW_t$$



p_0

x_0

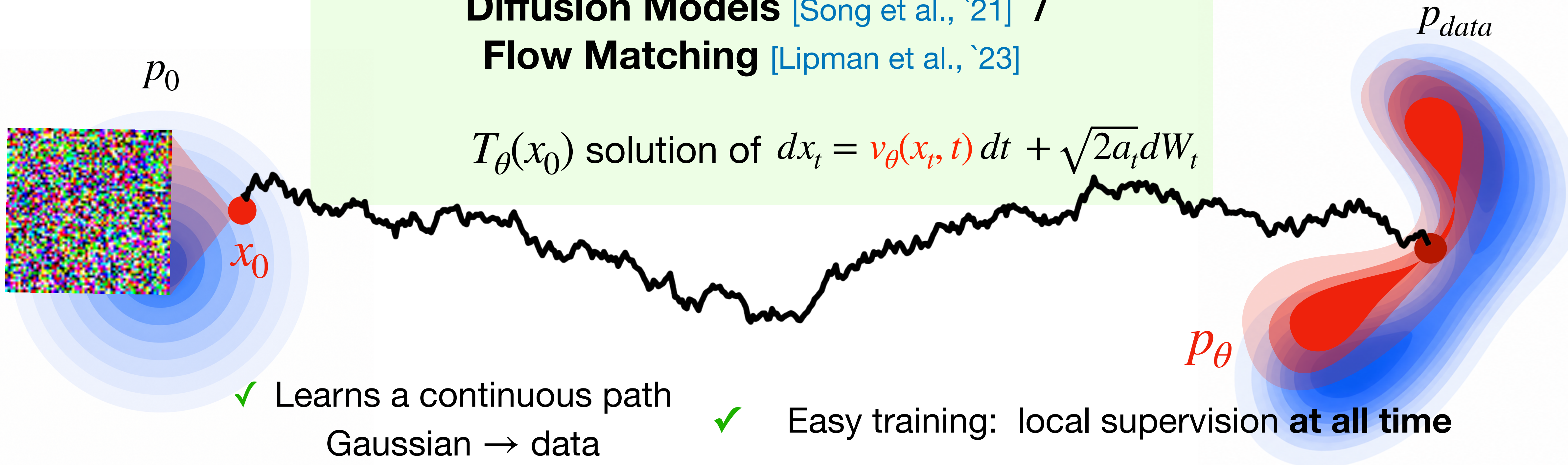
✓ Learns a continuous path
Gaussian $\rightarrow$ data



Easy training: local supervision **at all time**

p_θ

p_{data}



Diffusion models for posterior sampling $p(x | y = Ax + w)$

Apply the same algorithm with p_{data} replaced by $p(x | y)$

For $\alpha \geq 0$,

$$Y_0 \sim p_0, \quad dY_t = v_t(Y_t) dt + \sqrt{2a_t} dW_t$$

$$v_t(x) = -\frac{\dot{\eta}_{T-t}}{\eta_{T-t}} x + (1+\alpha) \eta_{T-t}^2 \dot{\sigma}_{T-t} \sigma_{T-t} \nabla \log p_{T-t}(x | y)$$

Using Baye's formula, for $x_t \sim p_t$ $\nabla \log p_t(x_t | y) = \nabla \log p_t(x_t) + \nabla \log p_t(y | x_t)$

✓ Unconditional score training

$$= \int p(y | x_0) p(x_0 | x_t) dx_0$$

✗ Untractable

DPS [Chung et. al, '23], PiGDM [Song et. al, '23], Moment Matching [Rozet et. al, '24] :

Use the Gaussian approximation $p(x_0 | x_t) \approx \mathcal{N}(\mathbb{E}[x_0 | x_t], \mathbb{V}[x_0 | x_t])$

Related to a pretrained denoiser by Tweedie's formulas

Diffusion models

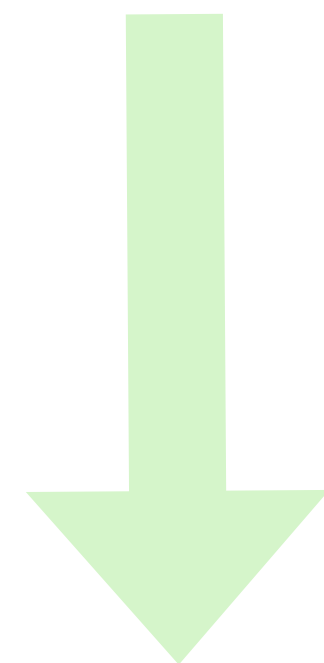
In the **continuous** setting, for fixed σ_t : VE $\eta_t = 1 \Leftrightarrow$ VP $\eta_t = \frac{1}{\sqrt{1 + \sigma_t^2}} \Leftrightarrow$ any rescaling η_t

VE schedule ($\sigma_t, \eta_t = 1$)

$$y_0 \sim p_T, \quad dy_t = v_t(y_t) dt + \sqrt{2\alpha\xi_t} dW_t$$

$$v_t = \xi_t \nabla \log p_{T-t}$$

$$\begin{cases} \beta_t = 0 \\ \xi_t = \dot{\sigma}_{T-t} \sigma_{T-t} \end{cases}$$



$$\hat{y}_t = \eta_t y_t$$

VP schedule (σ_t, η_t)

$$y_0 \sim \hat{p}_T, \quad d\hat{y}_t = v_t(\hat{y}_t) dt + \sqrt{2\alpha\xi_t} dW_t$$

$$v_t = \beta_t \text{Id} + \xi_t \nabla \log \hat{p}_{T-t} \left(\frac{\dot{}}{\eta_t} \right)$$

$$\begin{cases} \beta_t = -\frac{\dot{\eta}_{T-t}}{\eta_{T-t}} \\ \xi_t = \eta_{T-t}^2 \dot{\sigma}_{T-t} \sigma_{T-t} \end{cases}$$

Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process End continuous process

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

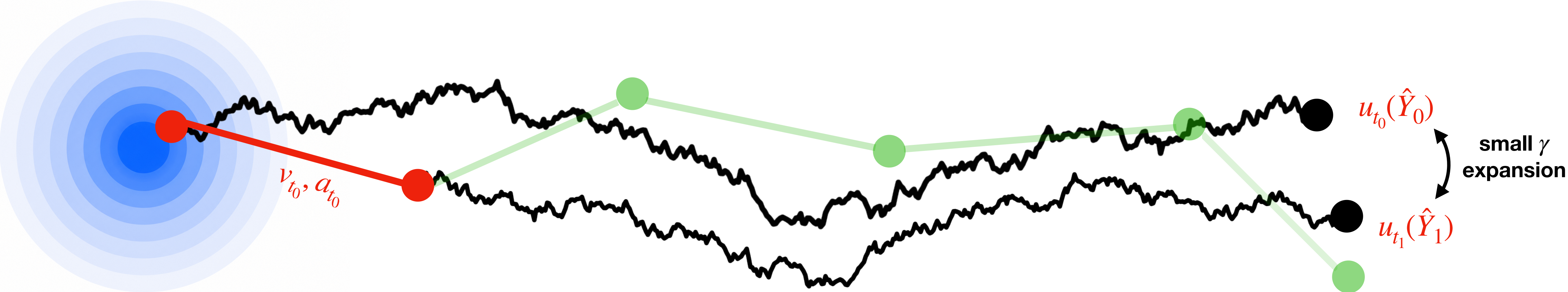
For $\alpha = 0$

$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$ (flow Jacobian) local acceleration (material derivative)

Sketch of the proof: Let $u_t(x) = \mathbb{E}[f(Y_T) | Y_t = x]$. We have:

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \mathbb{E}[u_{t_k}(\hat{Y}_k)] - \mathbb{E}[u_0(\hat{Y}_0)] \overset{\text{Telescopic sum}}{=} \sum_{k=0}^K \overset{\text{Expansion for small } \gamma}{u_{t_{k+1}}(\hat{Y}_{k+1}) - u_{t_k}(\hat{Y}_k)} = \sum_{k=0}^K \left(\gamma^2 \psi_{t_k}(\hat{Y}_k) + o(\gamma^2) \right) \overset{\text{Riemann-sum error}}{=} \gamma \int_0^T \psi_s(Y_s) ds + o(\gamma)$$



Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process End continuous process

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

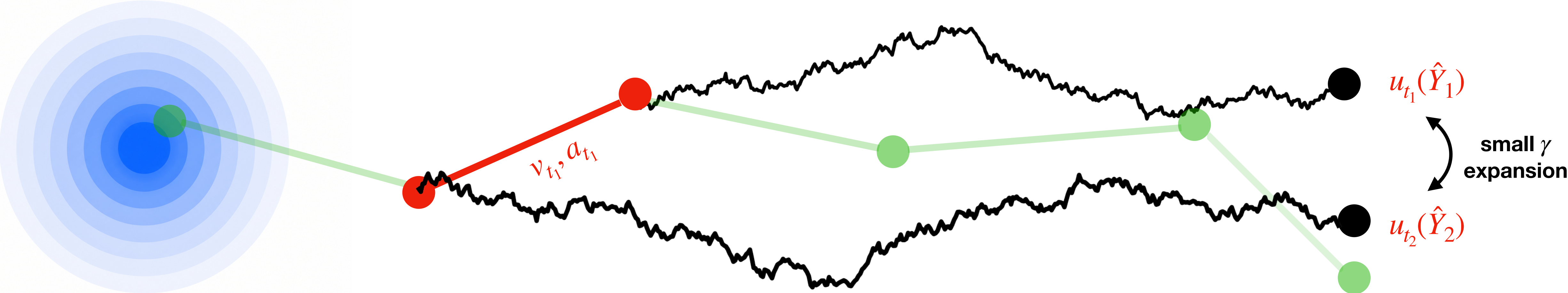
For $\alpha = 0$

$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$ (flow Jacobian) local acceleration (material derivative)

Sketch of the proof: Let $u_t(x) = \mathbb{E}[f(Y_T) | Y_t = x]$. We have:

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \mathbb{E}[u_{t_k}(\hat{Y}_k)] - \mathbb{E}[u_0(\hat{Y}_0)] \overset{\text{Telescopic sum}}{=} \sum_{k=0}^K \overset{\text{Expansion for small } \gamma}{u_{t_{k+1}}(\hat{Y}_{k+1}) - u_{t_k}(\hat{Y}_k)} = \sum_{k=0}^K \left(\gamma^2 \psi_{t_k}(\hat{Y}_k) + o(\gamma^2) \right) \overset{\text{Riemann-sum error}}{=} \gamma \int_0^T \psi_s(Y_s) ds + o(\gamma)$$



Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process End continuous process

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

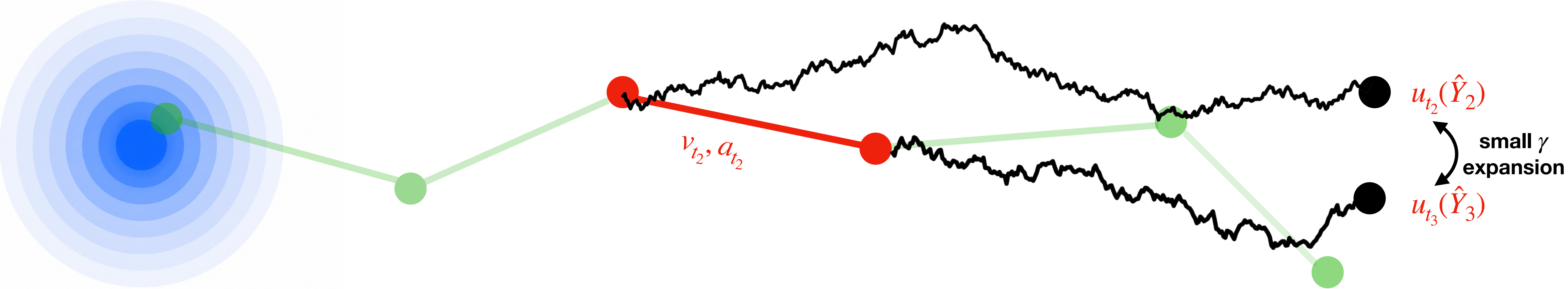
For $\alpha = 0$

$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$ (flow Jacobian) local acceleration (material derivative)

Sketch of the proof: Let $u_t(x) = \mathbb{E}[f(Y_T) | Y_t = x]$. We have:

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \mathbb{E}[u_{t_k}(\hat{Y}_k)] - \mathbb{E}[u_0(\hat{Y}_0)] \overset{\text{Telescopic sum}}{=} \sum_{k=0}^K u_{t_{k+1}}(\hat{Y}_{k+1}) - u_{t_k}(\hat{Y}_k) \overset{\text{Expansion for small } \gamma}{=} \sum_{k=0}^K \left(\gamma^2 \psi_{t_k}(\hat{Y}_k) + o(\gamma^2) \right) \overset{\text{Riemann-sum error}}{=} \gamma \int_0^T \psi_s(Y_s) ds + o(\gamma)$$



Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process End continuous process

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

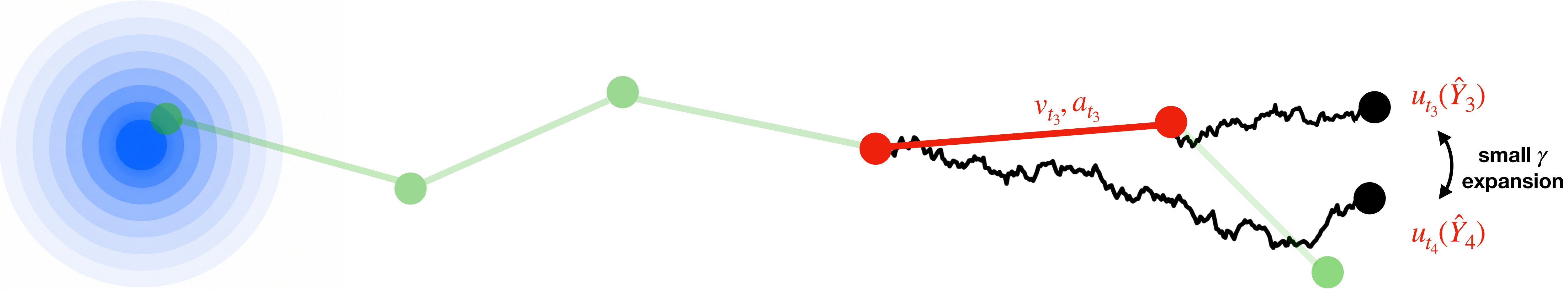
For $\alpha = 0$

$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$ (flow Jacobian) local acceleration (material derivative)

Sketch of the proof: Let $u_t(x) = \mathbb{E}[f(Y_T) | Y_t = x]$. We have:

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \mathbb{E}[u_{t_k}(\hat{Y}_k)] - \mathbb{E}[u_0(\hat{Y}_0)] \overset{\text{Telescopic sum}}{=} \sum_{k=0}^K u_{t_{k+1}}(\hat{Y}_{k+1}) - u_{t_k}(\hat{Y}_k) \overset{\text{Expansion for small } \gamma}{=} \sum_{k=0}^K \left(\gamma^2 \psi_{t_k}(\hat{Y}_k) + o(\gamma^2) \right) \overset{\text{Riemann-sum error}}{=} \gamma \int_0^T \psi_s(Y_s) ds + o(\gamma)$$



Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma \alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process
 $\mathbb{E}[f(\hat{Y}_K)]$

End continuous process
 $\mathbb{E}[f(Y_T)]$

$$= \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

For $\alpha = 0$

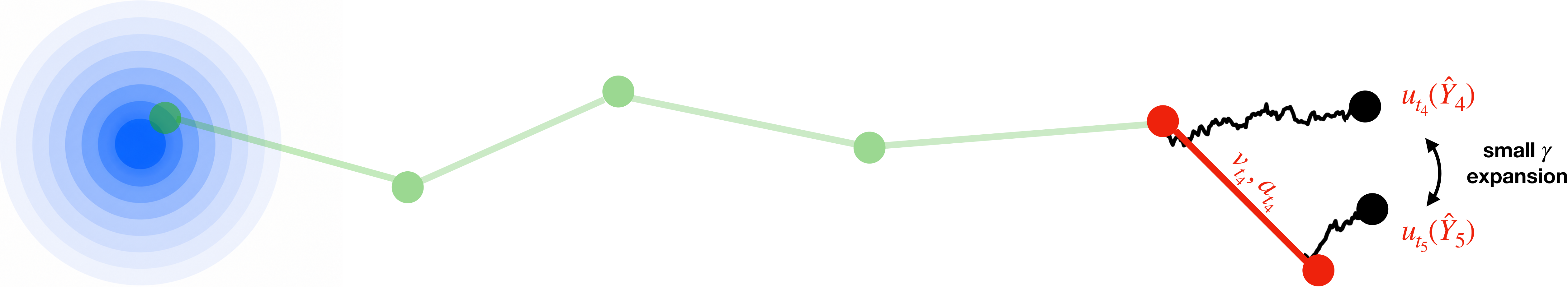
$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$
(flow Jacobian)

local acceleration
(material derivative)

Sketch of the proof: Let $u_t(x) = \mathbb{E}[f(Y_T) | Y_t = x]$. We have:

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \mathbb{E}[u_{t_k}(\hat{Y}_k)] - \mathbb{E}[u_0(\hat{Y}_0)] \overset{\text{Telescopic sum}}{=} \sum_{k=0}^K u_{t_{k+1}}(\hat{Y}_{k+1}) - u_{t_k}(\hat{Y}_k) \overset{\text{Expansion for small } \gamma}{=} \sum_{k=0}^K \left(\gamma^2 \psi_{t_k}(\hat{Y}_k) + o(\gamma^2) \right) \overset{\text{Riemann-sum error}}{=} \gamma \int_0^T \psi_s(Y_s) ds + o(\gamma)$$



Discretization error: general weak error

$$\hat{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \text{Id}), \quad \hat{x}_{k+1} = \hat{x}_k + \gamma v_{t_k}(\hat{x}_k) + \sqrt{2\gamma\alpha a_t} \hat{W}_k$$

Under mild assumptions on p_{data}

End discretized process

End continuous process

$$\mathbb{E}[f(\hat{Y}_K)] - \mathbb{E}[f(Y_T)] = \gamma \int_0^T \mathbb{E}[\psi_s(Y_s)] ds + o(\gamma)$$

Euler error due to freezing v_s, a_s on $[s, s + \gamma]$

For $\alpha = 0$

$$\psi_s(Y_s) = -\frac{1}{2} J_s(Y) D_s v_s(Y_s) \cdot \nabla f(Y_T)$$

Propagation $s \rightarrow T$
(flow Jacobian)

local acceleration
(material derivative)

✗ Not explicit in the general case

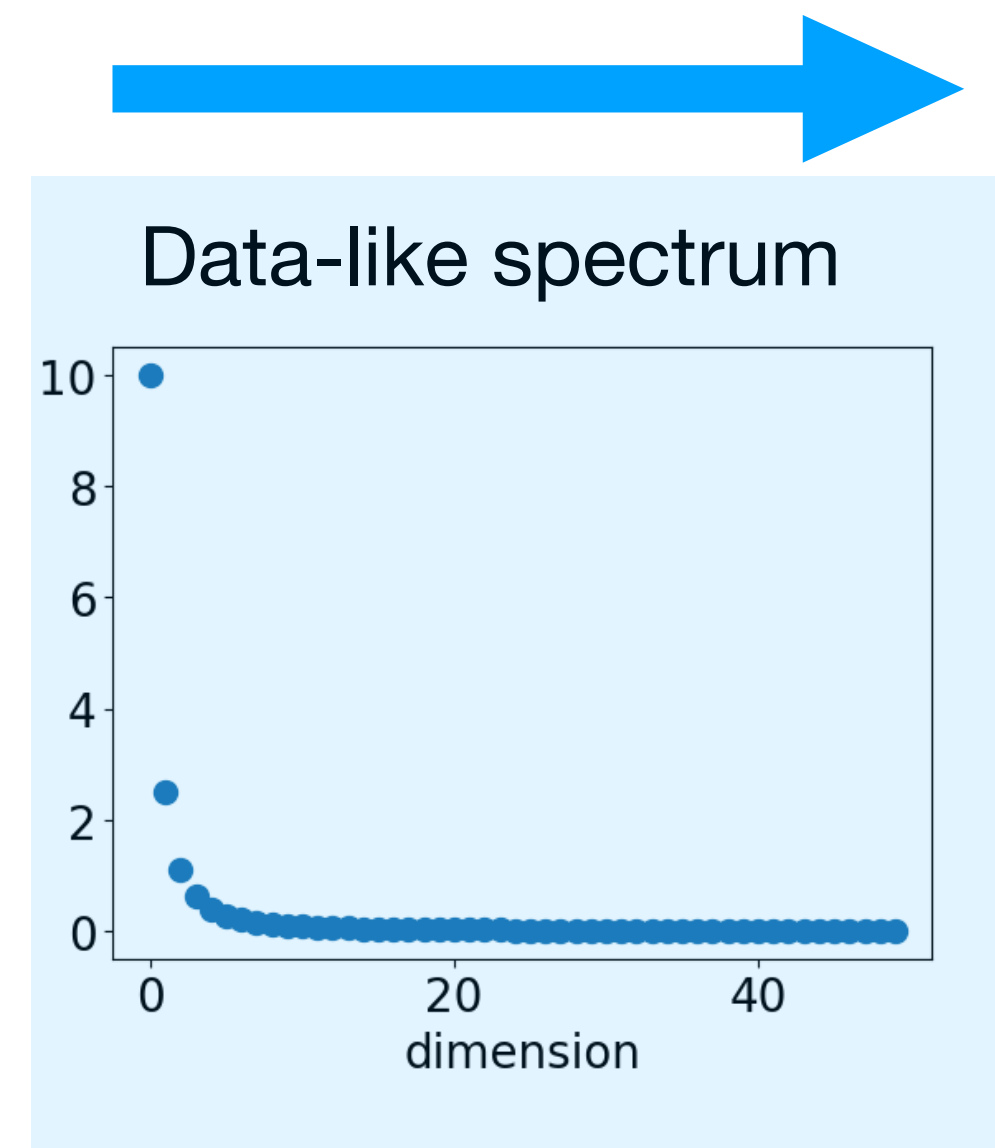
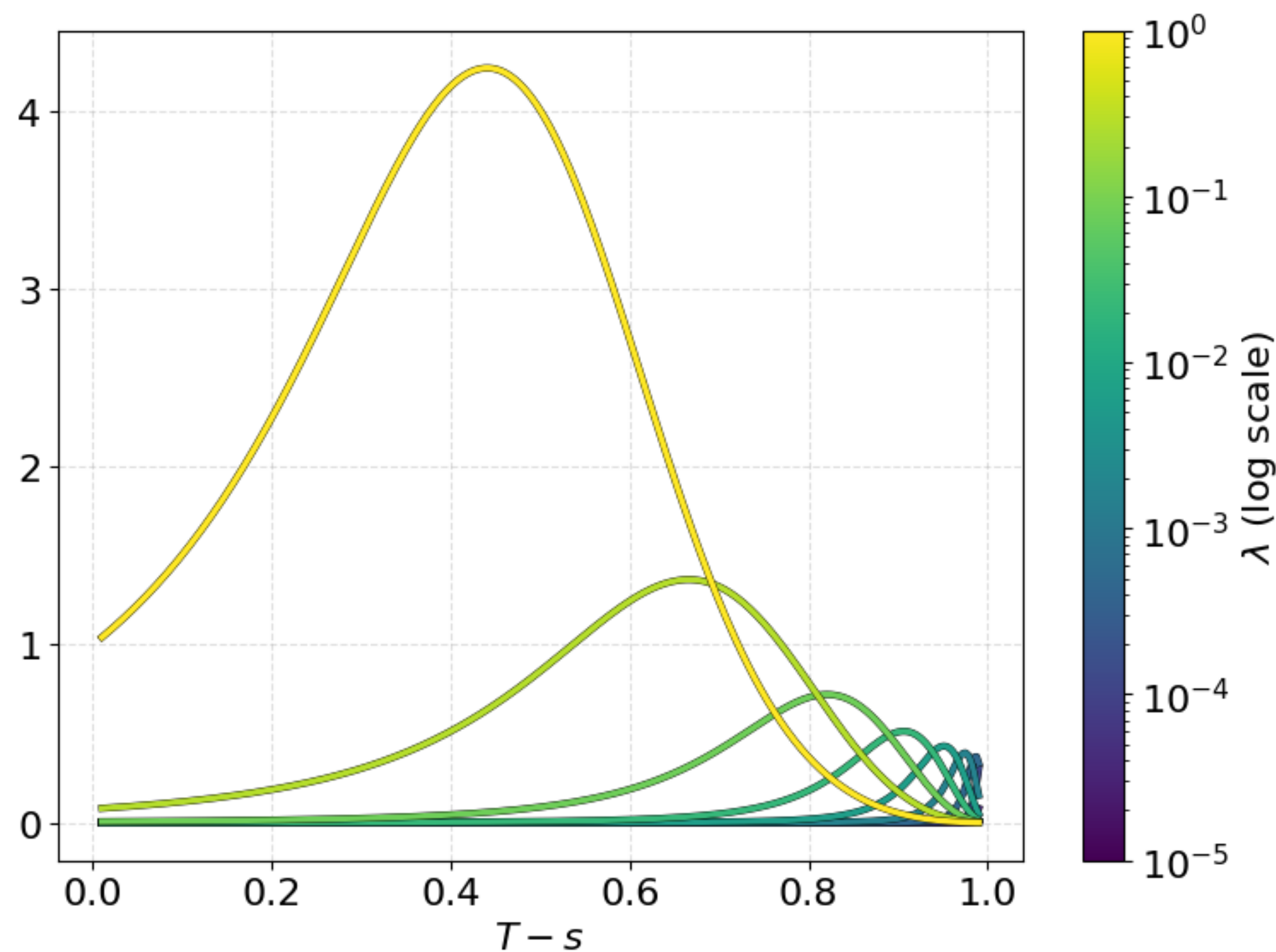
- ➡ $f(x) = x \rightarrow$ error on the mean
- ➡ $f(x) = xx^\top \rightarrow$ error on the covariance
- ➡ Error on higher moments

More details on the generalization error

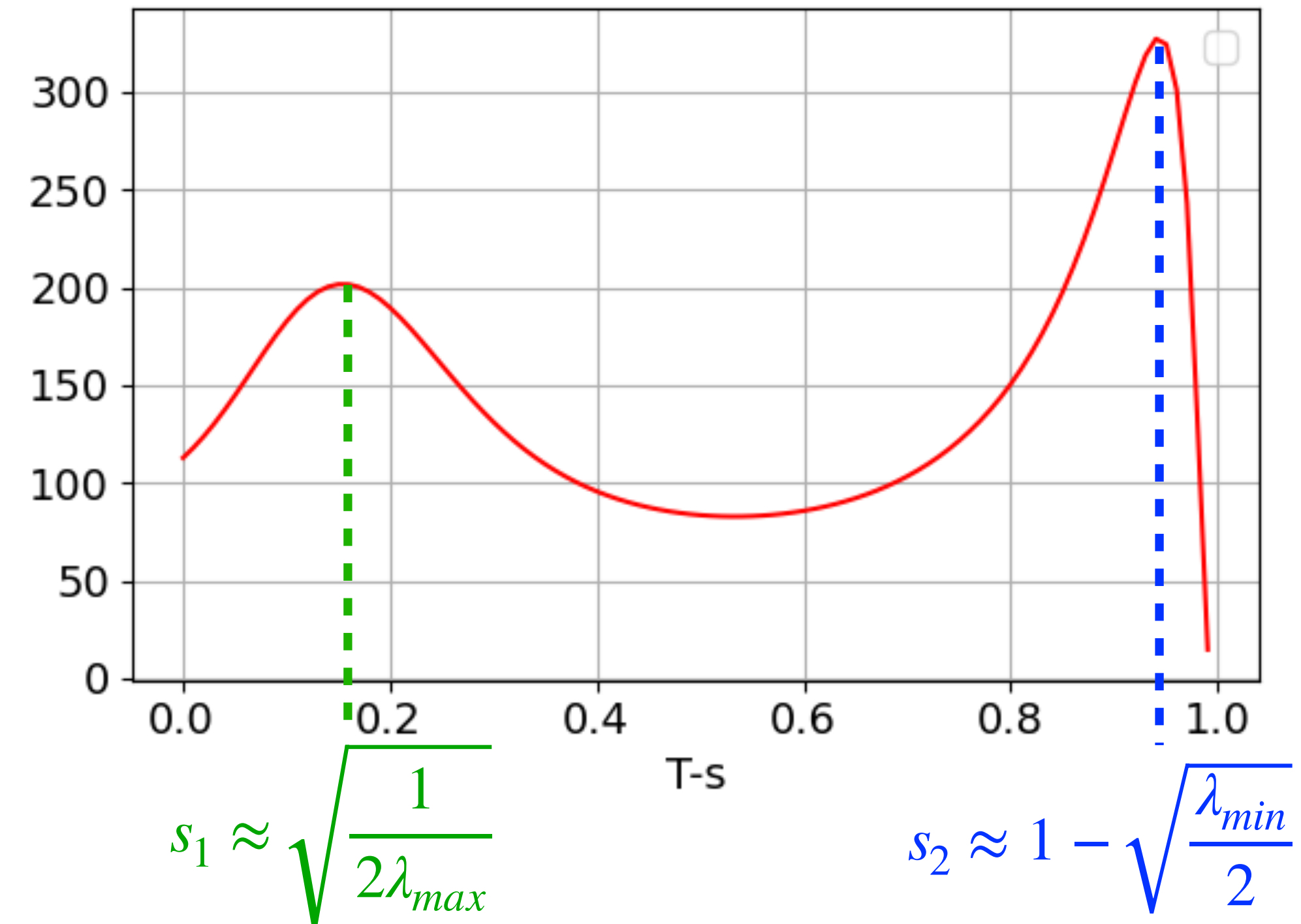
$$\mathcal{E}_t^{gen} = \int_0^T \sum_{i=1}^d \sum_{j=1}^d f(\lambda_i, \lambda_j, \sigma_s) ds \approx \int_0^T \sum_{i=1}^d \underbrace{(1+\alpha)^2 (\sigma_s \dot{\sigma}_s)^2 \frac{2}{\lambda_i} \left(\frac{\lambda_i}{\lambda_i + \sigma_s^2} \right)^{\alpha+3}}_{f_{gen}(s, \lambda_i)} ds$$

$f_{gen}(s, \lambda_i)$: generalization error
at time s / on direction λ_i

$f_{gen}(s, \lambda_i)$ w.r.t time s , for each direction λ_i :



Full error $\sum_{i=1}^d f_{gen}(s, \lambda_i)$ w.r.t time s



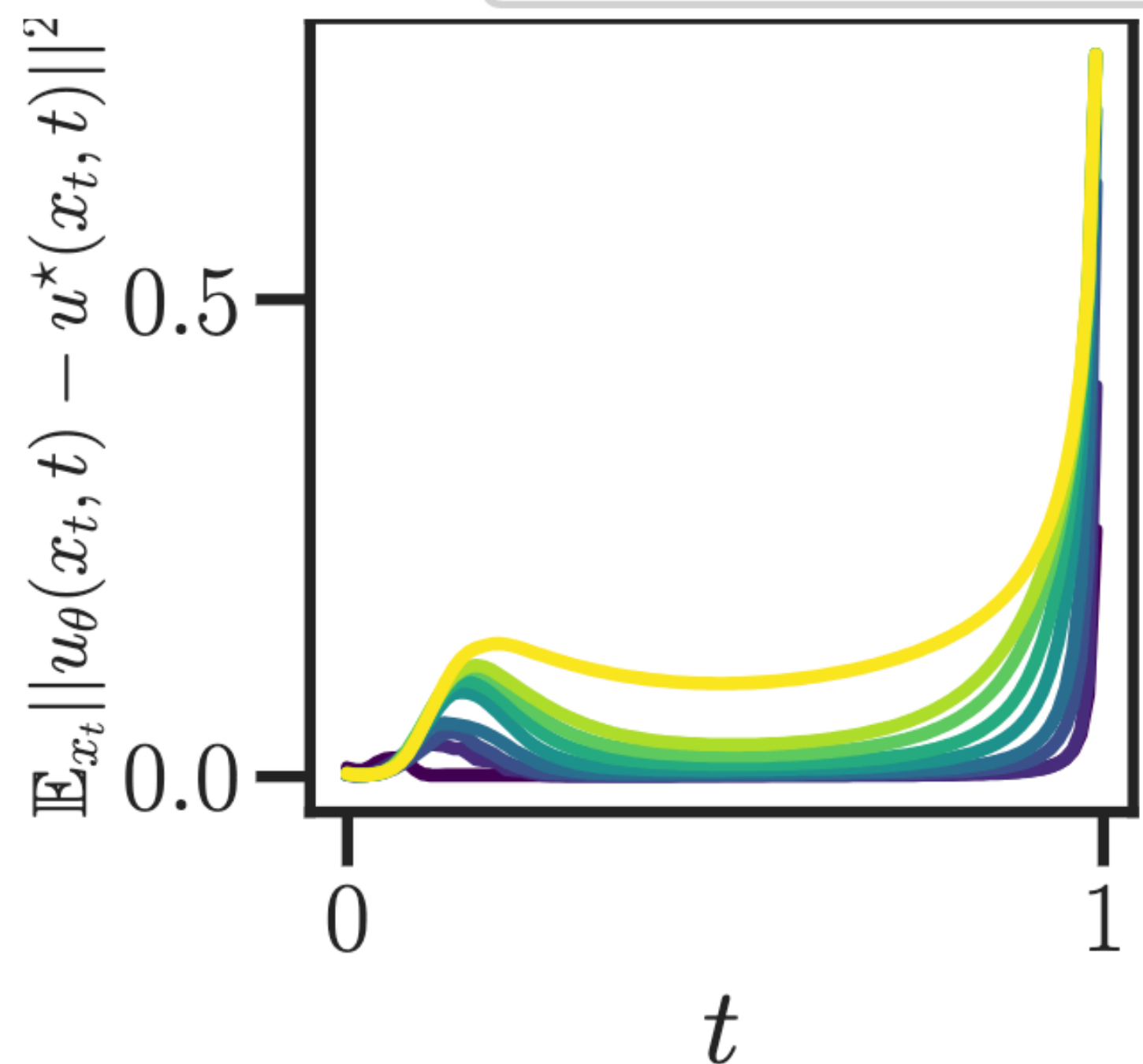
More details on the generalization error

$$\mathcal{E}_t^{gen} = \int_0^T \sum_{i=1}^d \sum_{j=1}^d f(\lambda_i, \lambda_j, \sigma_s) ds \approx \int_0^T \sum_{i=1}^d \underbrace{(1+\alpha)^2 (\sigma_s \dot{\sigma}_s)^2 \frac{2}{\lambda_i} \left(\frac{\lambda_i}{\lambda_i + \sigma_s^2} \right)^{\alpha+3}}_{f_{gen}(s, \lambda_i)} ds$$

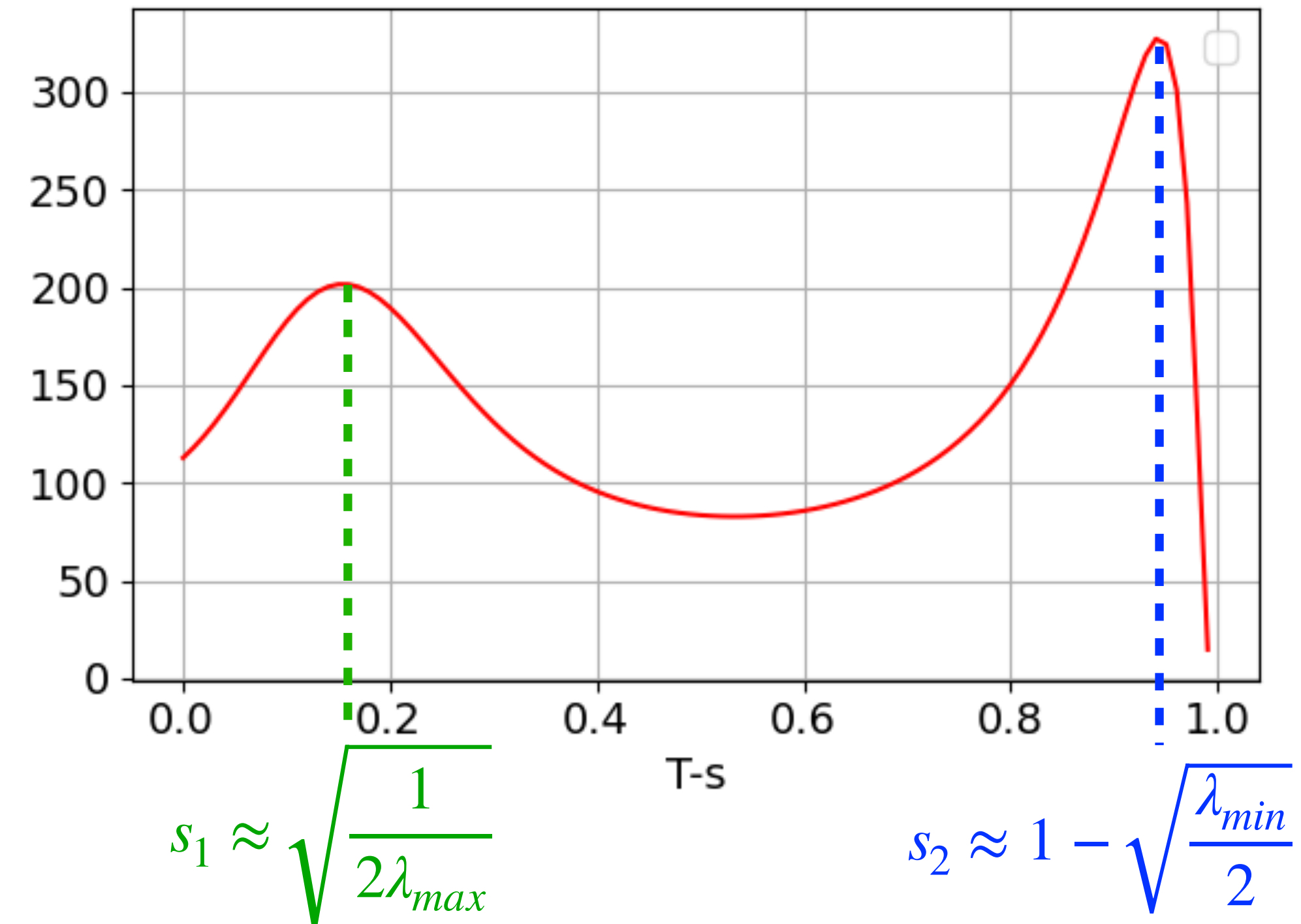
$f_{gen}(s, \lambda_i)$: **generalization error**
at time s / on **direction** λ_i

[Bertrand et al., 2025]

real diffusion model sampling error :



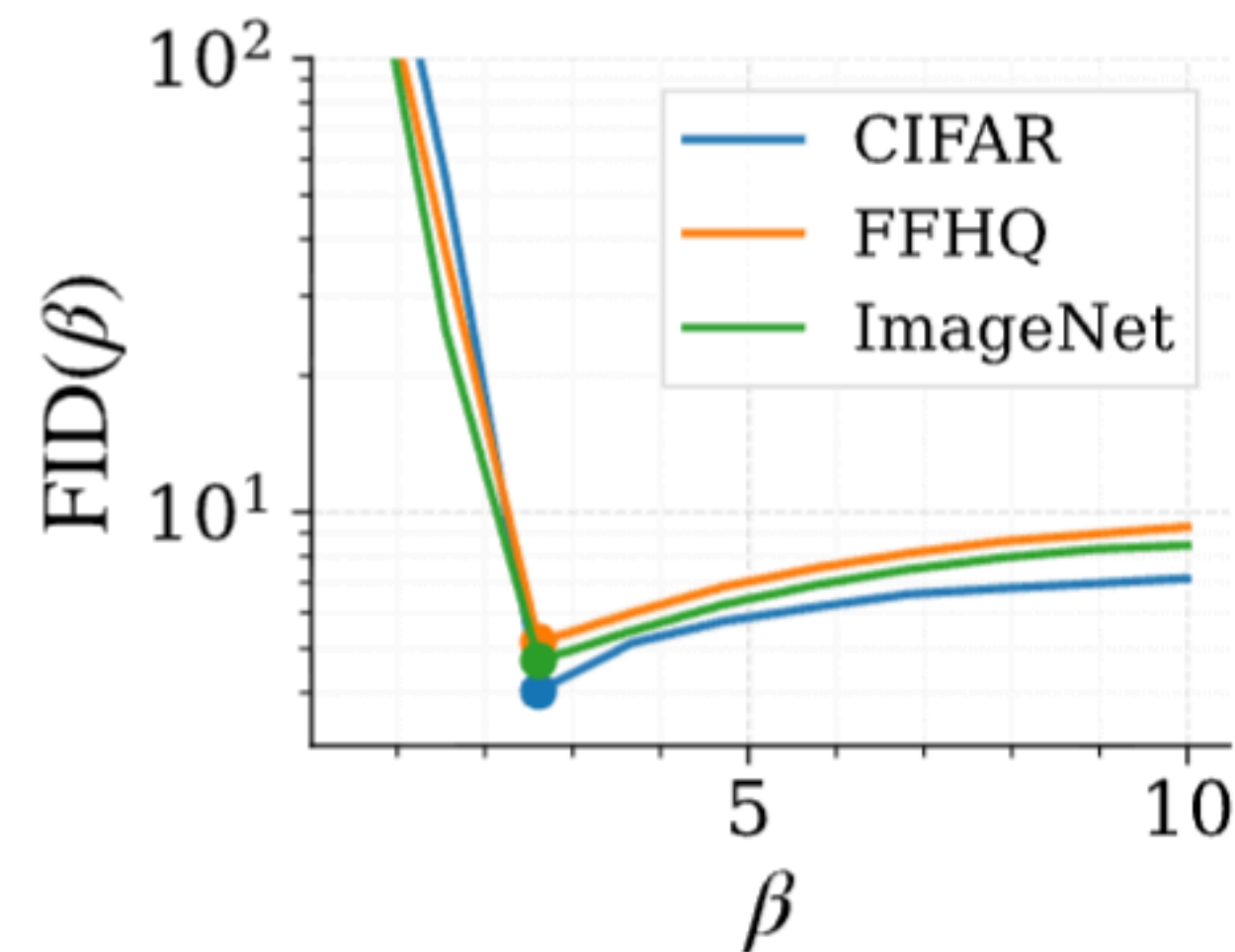
Full error $\sum_{i=1}^d f_{gen}(s, \lambda_i)$ w.r.t time s



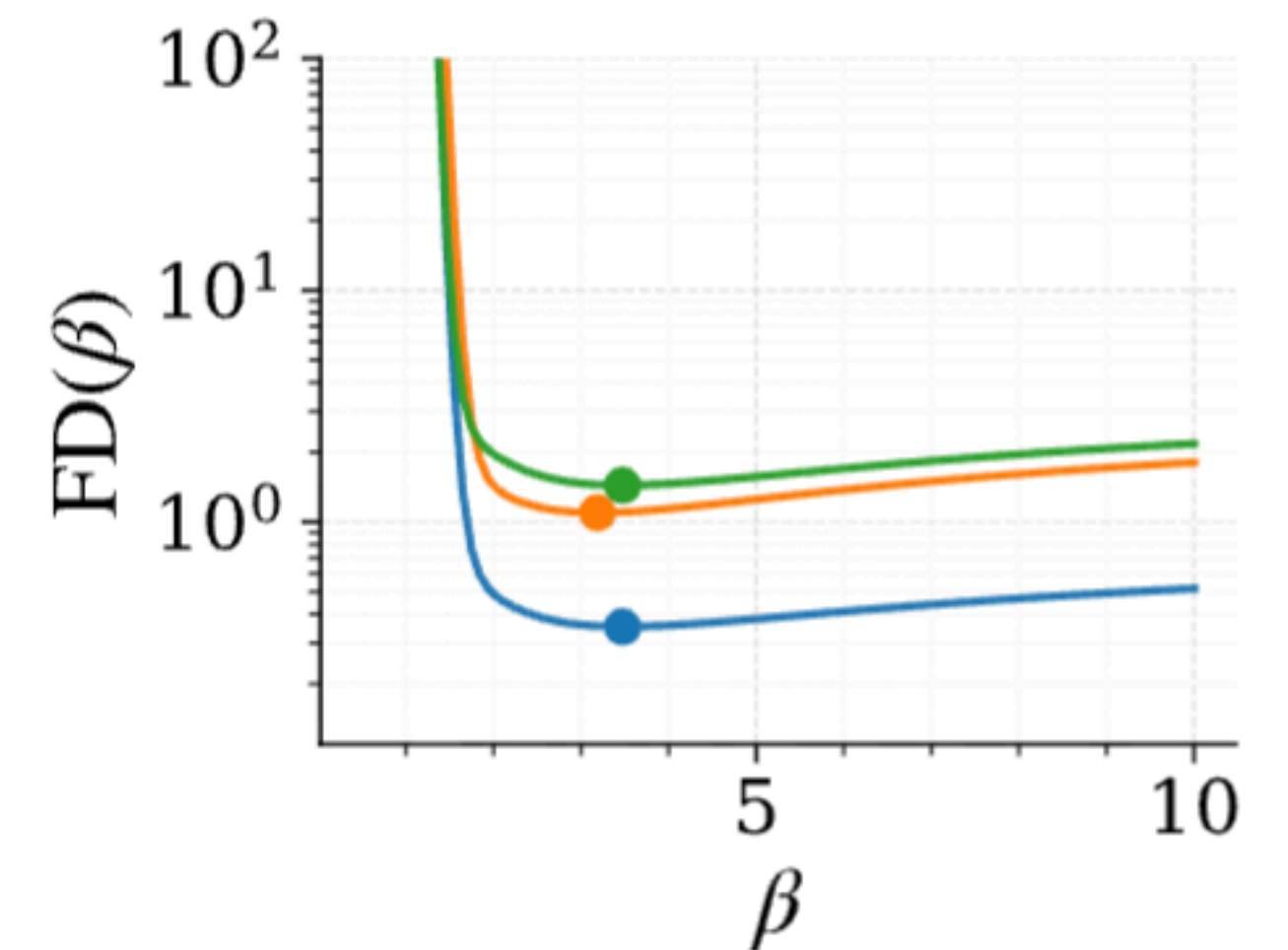
Optimization of the noise schedule σ_t

For « polynomial schedule » $\sigma_t = \propto t^\beta$

We prove that the polynomial schedule is well-adapted to **anisotropic covariance spectrum**



Empirical sampling (FID)



Gaussian theory (W_2)