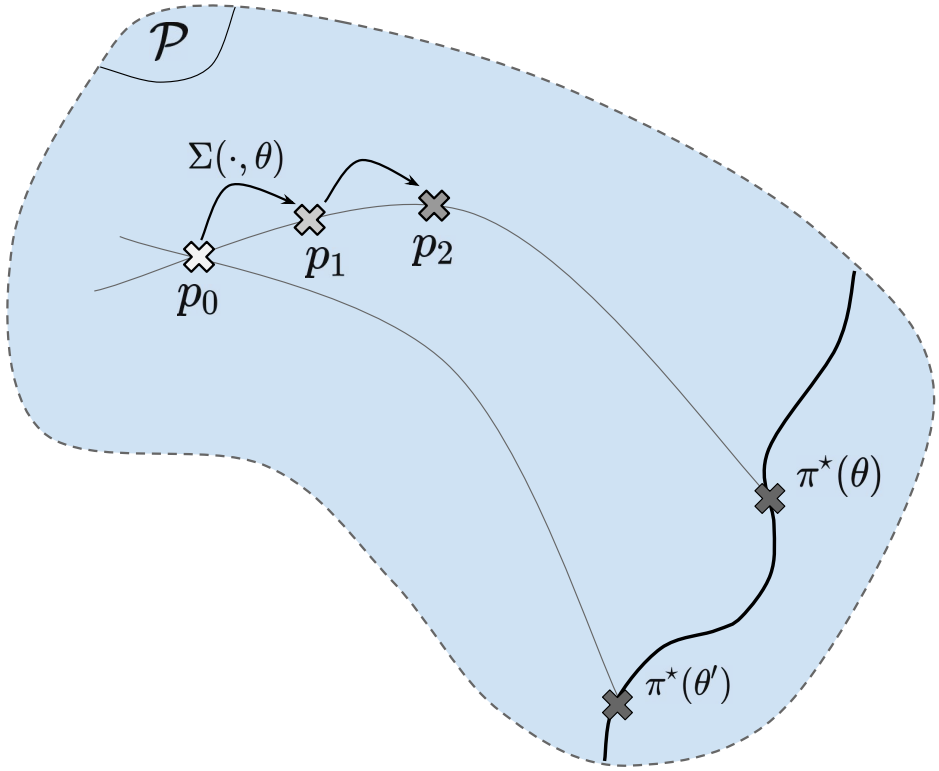
Implicit Diffusion:

Efficient Optimization through Stochastic Sampling



AISTATS 2025 - National University of Singapore

Quentin Berthet, Google DeepMind

Optimization through sampling

Sampling on $\mathcal{X}$ depending on a parameter $\theta \in \mathbf{R}^d$

$$\text{Mapping} : \theta \in \mathbf{R}^d \to \pi^\star(\theta) \in \mathcal{P}(\mathcal{X})\,.$$

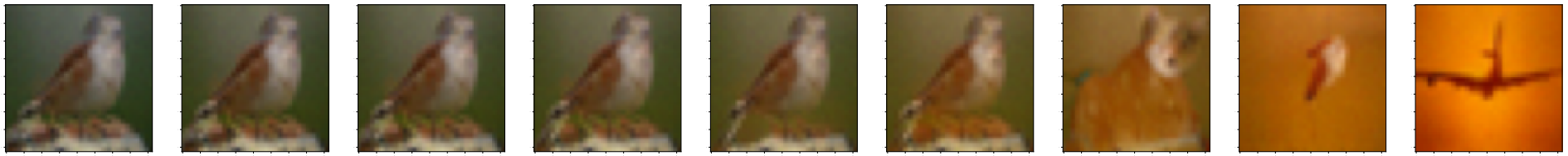
Examples: Langevin dynamics with $V(x, \theta)$, denoising diffusion with $s_\theta(x, t)$.

Optimization on $\mathcal{P}(\mathcal{X})$ "through" sampling

$$\min_{\theta \in \mathbf{R}^d} \ell(\theta) := \min_{\theta \in \mathbf{R}^d} \mathcal{F}(\pi^\star(\theta))$$

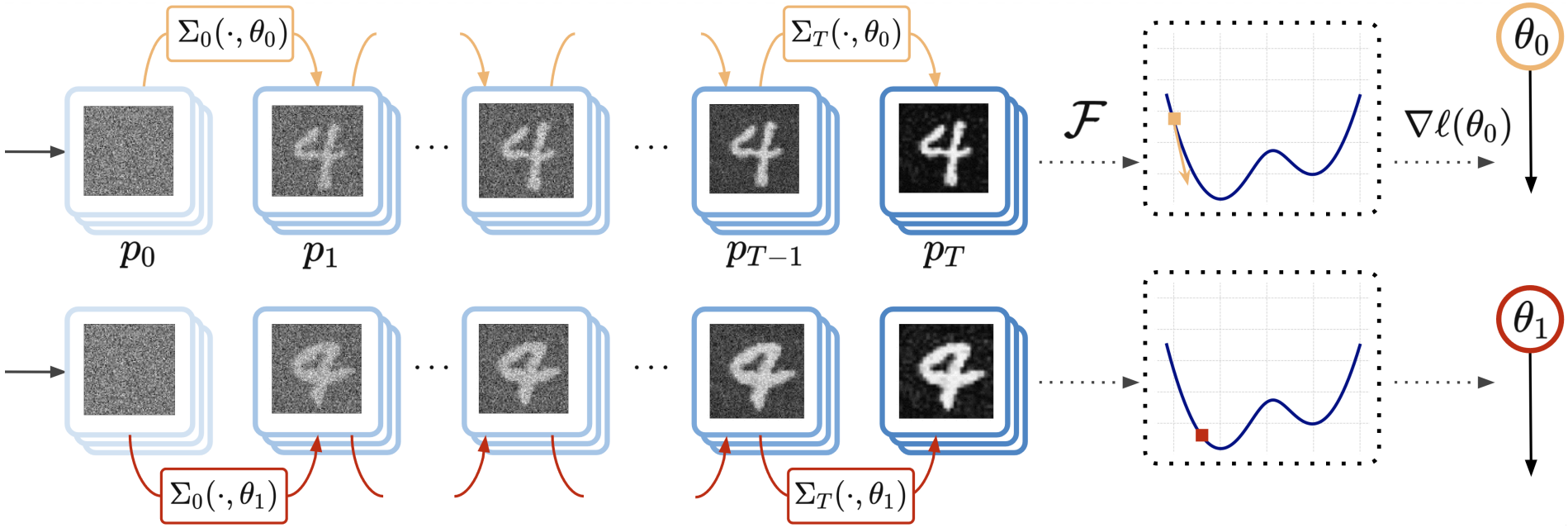
Examples: Finetuning with reward $\mathcal{F}(\pi) := -\lambda \mathbf{E}_{x\sim\pi}[R(x)] + \beta \mathsf{KL}(\pi, \pi_{\mathsf{ref}})\,.$

Figure: Increasing the **red** reward, for a large $\lambda > 0$



Optimization through iterative sampling

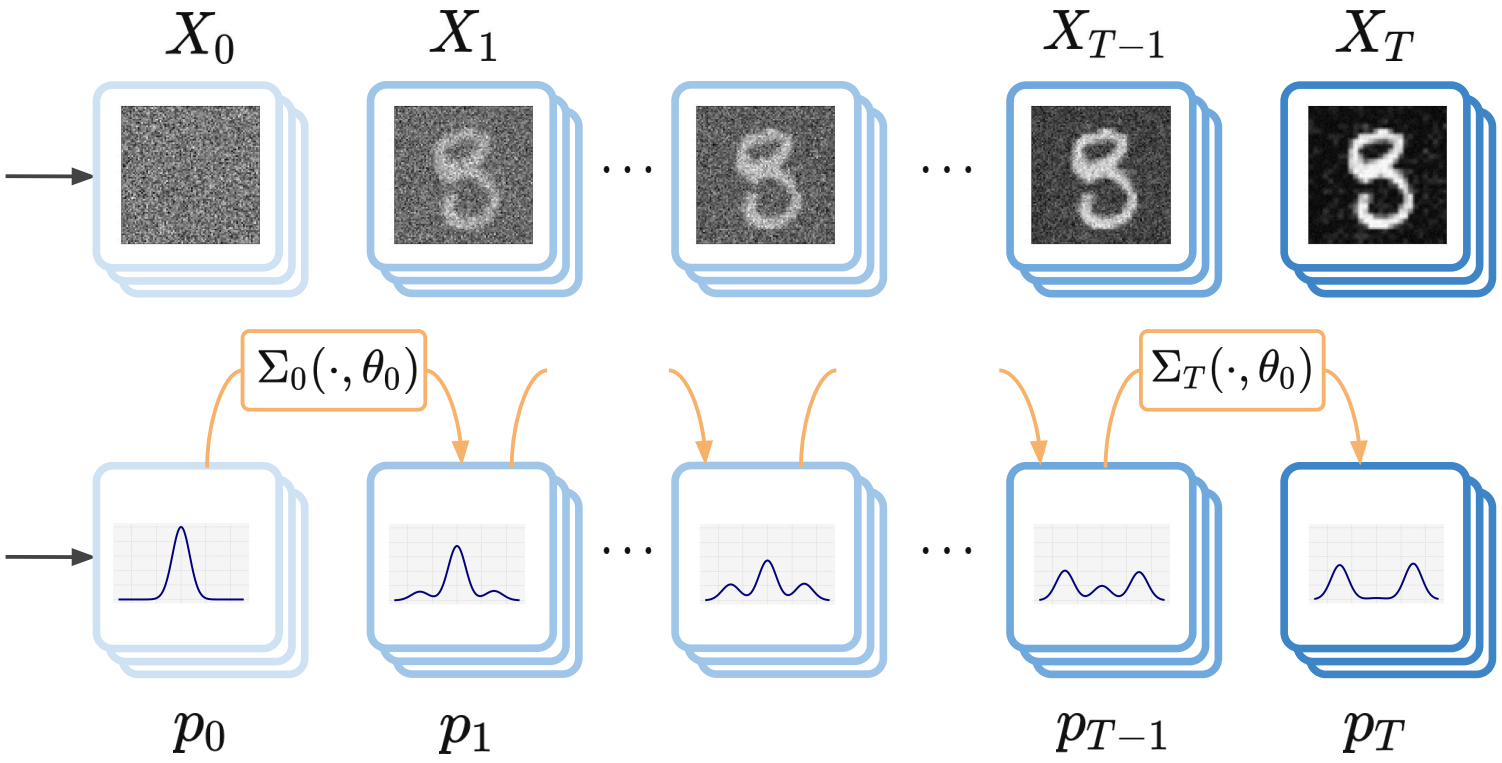
Optimize the parameters $\theta \in \mathbf{R}^d$ of sampling process with output $\pi^\star(\theta)$.



Challenges: Evaluating gradients of ℓ, efficiently optimizing.

Iterative sampling with parameters

Sampling with an iterative algorithm, using a **parametric** function



Acting on particles X_s, distribution p_s, for some $\theta \in \mathbf{R}^d$:

$$p_{s+1} = \Sigma_s(p_s, \theta)$$

End result: distribution $\pi^\star(\theta)$, either $s \to \infty$ or $s = T$.

Iterative sampling- Examples

Langevin dynamics: For $V(\cdot, \theta)$, SDE (Roberts and Tweedie, 96)

$$\mathrm{d}X_t = -\nabla V(X_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

When $t \to \infty$, $p_t \to \pi^\star(\theta) = \exp(-V(x, \theta))/Z_\theta$ - Gibbs distribution

Discrete step-size approximation: $X_{k+1} = X_k - 2\delta\nabla_1 V(X_k, \theta) + \sqrt{2\delta}\Delta B_k$.

Denoising diffusion: For $s_\theta(\cdot, t)$, model for parametrized score function

$$\mathrm{d}Y_t = \{Y_t + 2s_\theta(Y_t, T - t)\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,, \quad Y_0 \sim \mathcal{N}(0, I_d)\,.$$

Law of $Y_T = \pi^\star(\theta)$, used to approximate p_{data} (after training score matching).
(Hyvarinen, 05, Vincent, 11, Ho et al., 20)

Iterative sampling

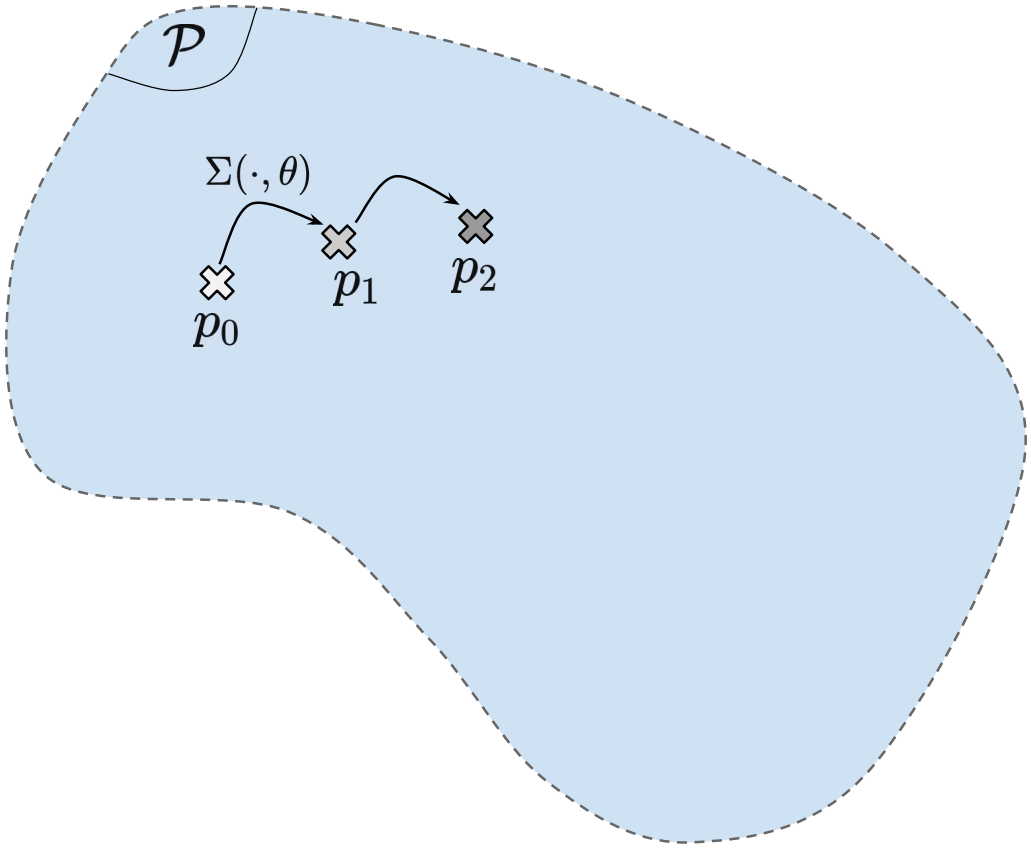
- Formalism for iterative sampling

$$p_{s+1} = \Sigma_s(p_s, \theta)\,.$$

- Implicitly defines a mapping

$$\pi^\star : \mathbf{R}^d \to \mathcal{P}$$

- Similar to optimization algorithms.



Sometimes, this perspective is formal, e.g. Langevin dynamics

$$\mathrm{d}X_t = -\nabla V(X_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

Distribution μ_t of X_t follows a Wasserstein gradient flow on $\mathrm{KL}(\mu, \pi^\star(\theta))$.

(Jordan et al. 98, Korba and Salim 22)

Iterative sampling

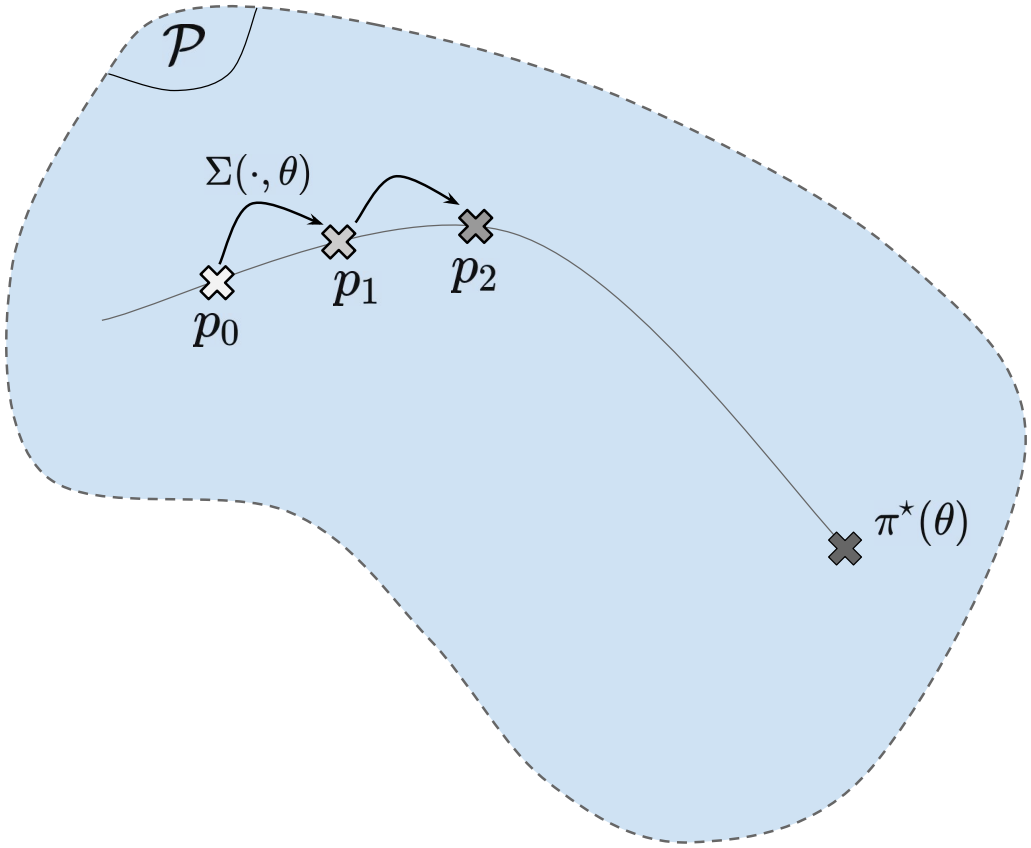
- Formalism for iterative sampling

$$p_{s+1} = \Sigma_s(p_s, \theta)\,.$$

- Implicitly defines a mapping

$$\pi^\star : \mathbf{R}^d \to \mathcal{P}$$

- Similar to optimization algorithms.



Sometimes, this perspective is formal, e.g. Langevin dynamics

$$\mathrm{d}X_t = -\nabla V(X_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

Distribution μ_t of X_t follows a Wasserstein gradient flow on $\mathrm{KL}(\mu, \pi^\star(\theta))$.

(Jordan et al. 98, Korba and Salim 22)

Iterative sampling

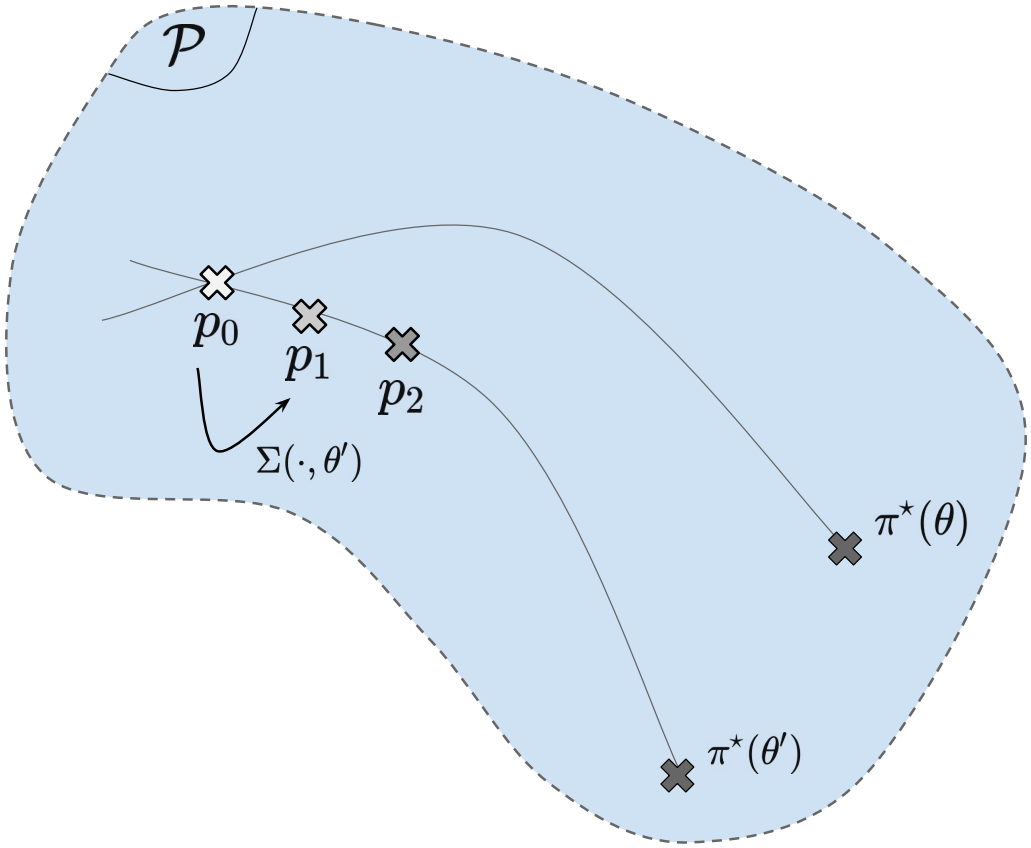
- Formalism for iterative sampling

$$p_{s+1} = \Sigma_s(p_s, \theta)\,.$$

- Implicitly defines a mapping

$$\pi^\star : \mathbf{R}^d \to \mathcal{P}$$

- Similar to optimization algorithms.



Sometimes, this perspective is formal, e.g. Langevin dynamics

$$\mathrm{d}X_t = -\nabla V(X_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

Distribution μ_t of X_t follows a Wasserstein gradient flow on $\mathrm{KL}(\mu, \pi^\star(\theta))$.

(Jordan et al. 98, Korba and Salim 22)

Iterative sampling

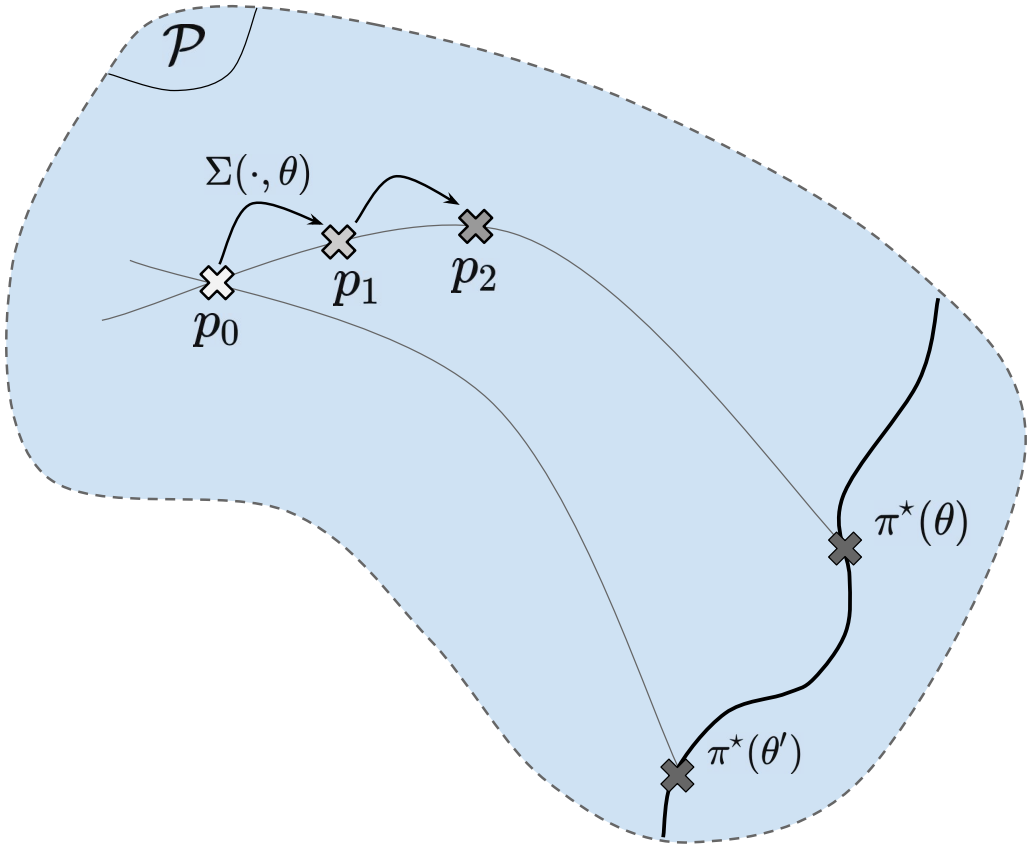
- Formalism for iterative sampling

$$p_{s+1} = \Sigma_s(p_s, \theta)\,.$$

- Implicitly defines a mapping

$$\pi^\star : \mathbf{R}^d \to \mathcal{P}$$

- Similar to optimization algorithms.



Sometimes, this perspective is formal, e.g. Langevin dynamics

$$\mathrm{d}X_t = -\nabla V(X_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

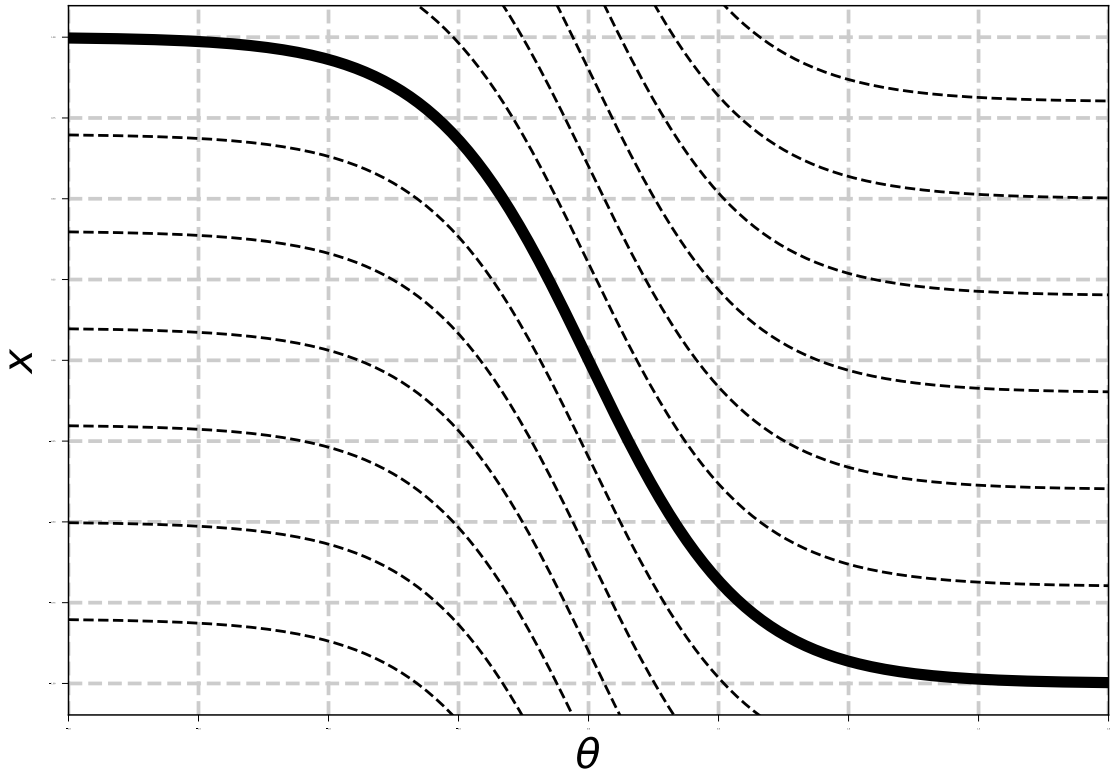
Distribution μ_t of X_t follows a Wasserstein gradient flow on $\mathrm{KL}(\mu, \pi^\star(\theta))$.

(Jordan et al. 98, Korba and Salim 22)

Bilevel optimization

Solving an **outer** optimization problem, subject to **inner** optimization constraints

$$\min_{x \in \mathcal{X},\, \theta \in \mathbf{R}^d} f(x) \quad \text{such that} \quad x \in \operatorname*{argmin}_{x \in \mathcal{X}} g(x, \theta)\,.$$

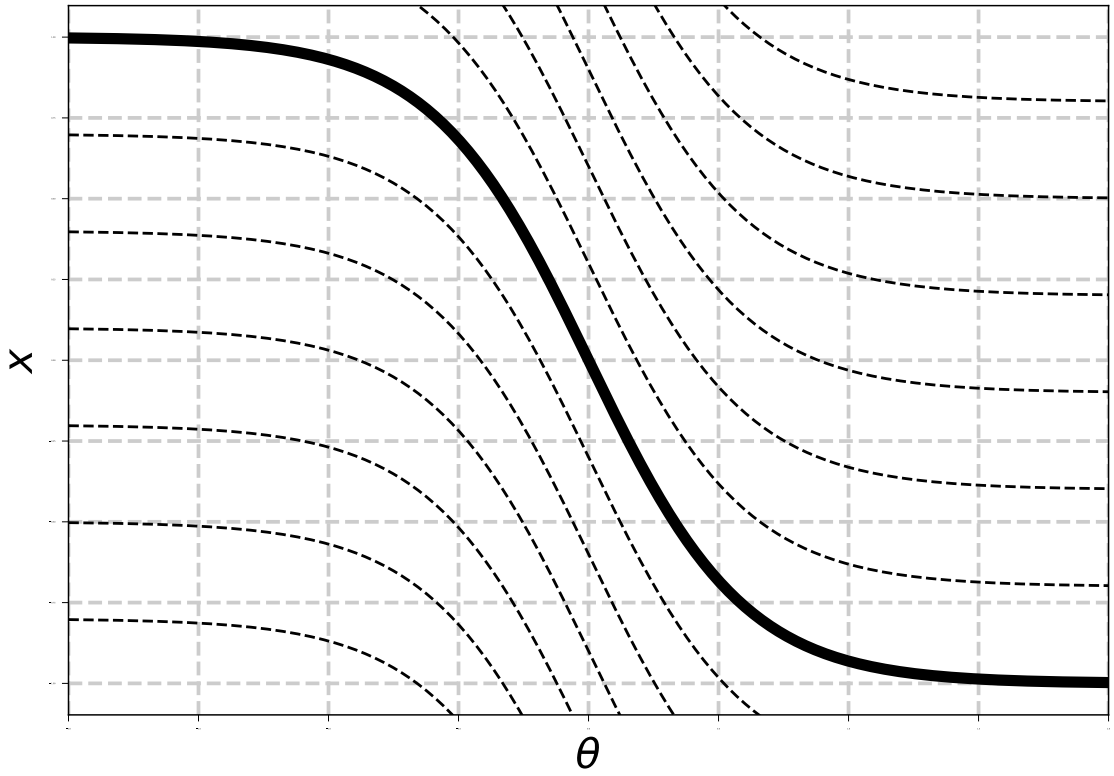


Often used to optimize the outcomes of a training process for another objective.

Bilevel optimization

Solving an **outer** optimization problem, subject to **inner** optimization constraints

$$\min_{\theta \in \mathbf{R}^d} f(x^\star(\theta)) \quad \text{where} \quad x^\star(\theta) \in \operatorname*{argmin}_{x \in \mathcal{X}} g(x, \theta)\,.$$

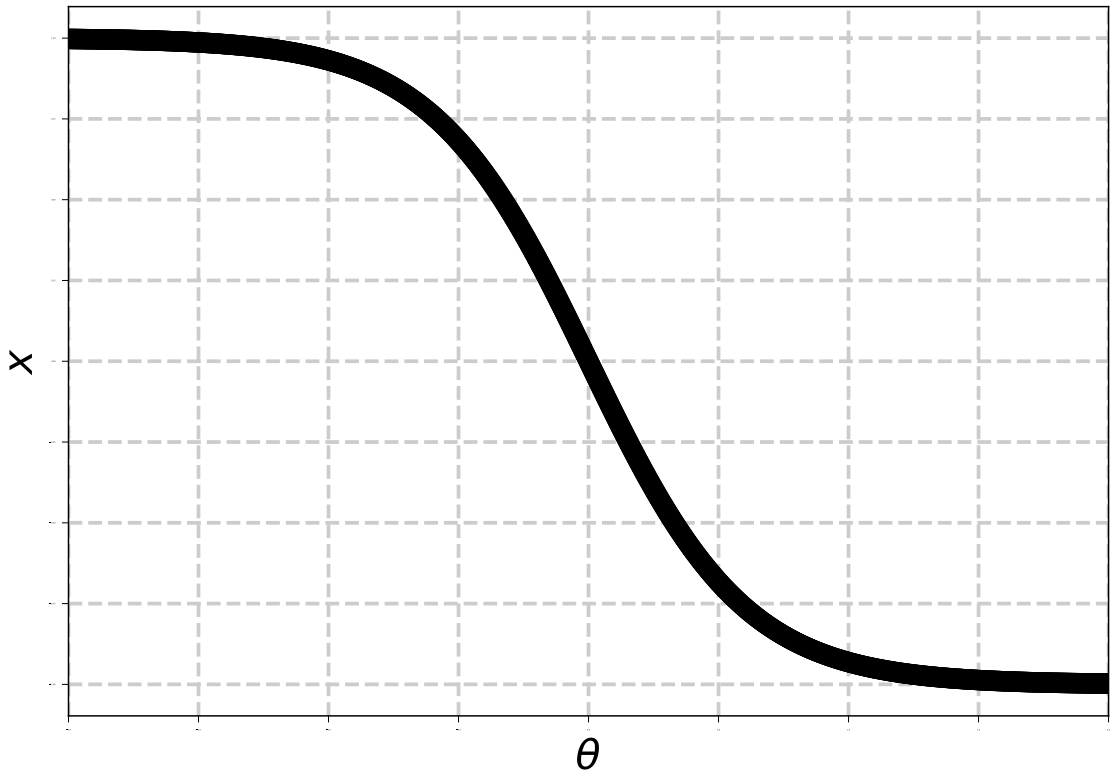


Often used to optimize the outcomes of a training process for another objective.

Bilevel optimization

Solving an **outer** optimization problem, subject to **inner** optimization constraints

$$\min_{\theta \in \mathbf{R}^d} f(x^\star(\theta)) \quad \text{where} \quad x^\star(\theta) \in \operatorname*{argmin}_{x \in \mathcal{X}} g(x, \theta)\,.$$

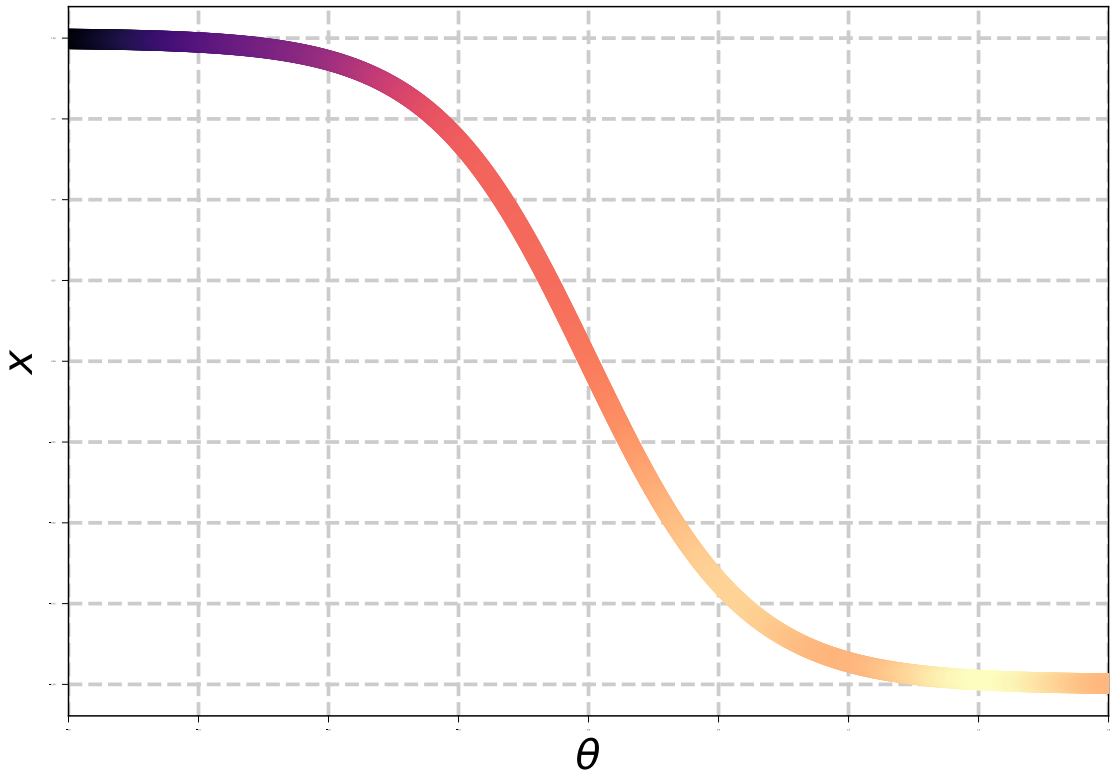


Often used to optimize the outcomes of a training process for another objective.

Bilevel optimization

Solving an **outer** optimization problem, subject to **inner** optimization constraints

$$\min_{\theta \in \mathbf{R}^d} f(x^\star(\theta)) \quad \text{where} \quad x^\star(\theta) \in \operatorname*{argmin}_{x \in \mathcal{X}} g(x, \theta)\,.$$



Often used to optimize the outcomes of a training process for another objective.

Bilevel optimization - a few examples

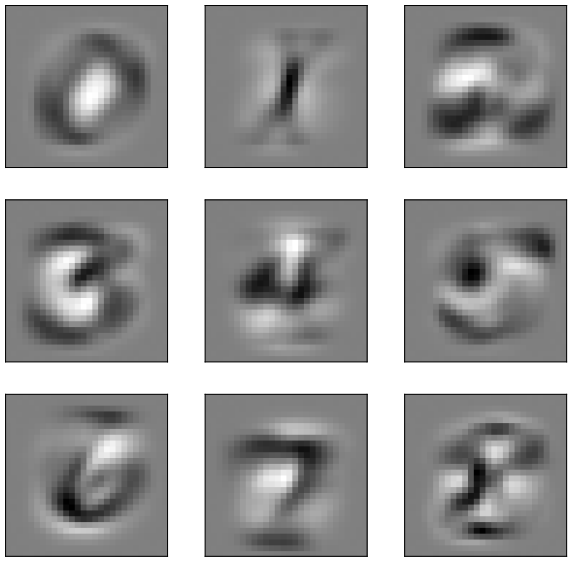
- Hyperparameter optimization, e.g. regularization strength (Franchesci et al, 18)

$$\min_{\theta \in \mathbf{R}^d} \ell(A_{\text{val}} x^\star(\theta), y_{\text{val}}) \quad \text{where} \quad x^\star(\theta) \in \operatorname*{argmin}_{x \in \mathcal{X}} \ell(A_{\text{train}} x, y_{\text{train}}) + e^\theta \odot R(x)\,.$$

- Dataset distillation

  (Wang et al., 18, Blondel et al., 22)

  - Dataset $\theta \in \mathbf{R}^{10 \times 28 \times 28}$.
  - **Inner**: train model with weights x on tiny dataset θ. $\to x^\star(\theta)$
  - **Outer**: optimize the loss of model with weights $x^\star(\theta)$ on MNIST.



- Learning dynamics: minimize loss on noisy observations of $\dot{x}_t = T(x_t, \theta)$.

Bilevel optimization - Joint optimization

Vanilla algorithm for bilevel optimization:

- For $t \geq 0$, with $\theta_0 \in \mathbf{R}^d$
  - Solve inner problem $x^\star(\theta_t)$ (approximately, with an inner loop)
  - Compute / approximate $\partial_\theta x^\star(\theta_t)$ and $\nabla_\theta f(x^\star(\theta_t))$
  - Upgrade with first-order method, e.g. $\theta_{t+1} = \theta_t - \eta_t \nabla_\theta f(x^\star(\theta_t))$

Full inner problem solve and gradient computation at each iteration.

Not well-suited with stochastic optimization.

Bilevel optimization - Joint optimization

Algorithm for **joint bilevel optimization**:

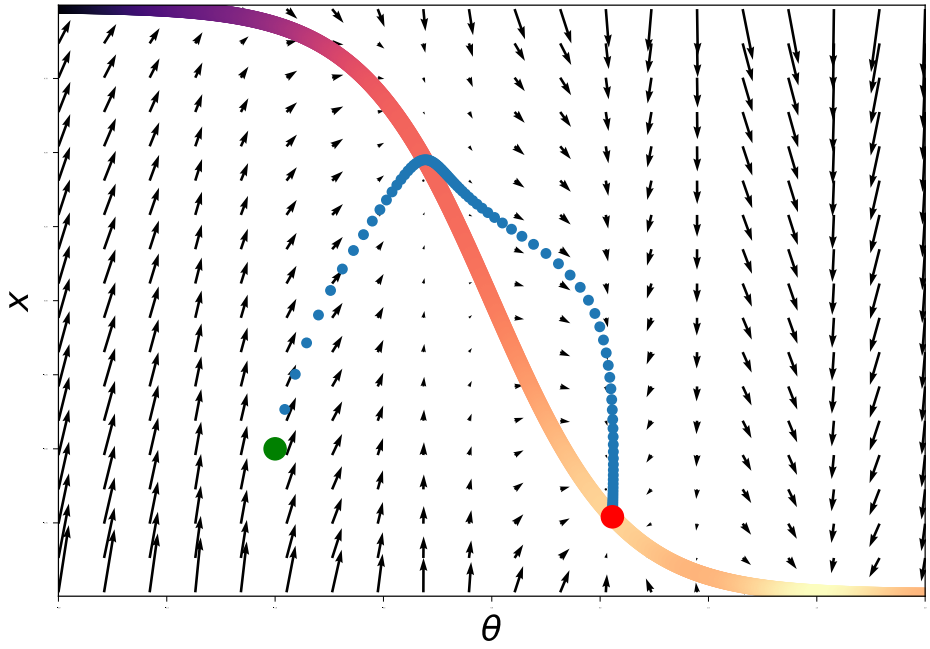
- For $t \geq 0$, with $x_0 \in \mathcal{X}$, $\theta_0 \in \mathbf{R}^d$
  - Compute gradients $\nabla_1 g(x_t, \theta_t)$ and $\Gamma(x_t, \theta_t)$ $\boxed{\approx \nabla_\theta \ell(\theta_t)}$
  - Update with first-order method, at different speeds

$$x_{t+1} = x_t - \eta_t \nabla_1 g(x_t, \theta_t)\,, \quad \theta_{t+1} = \theta_t - \varepsilon_t \eta_t \Gamma(x_t, \theta_t)\,.$$

Remarks

Never perform a full optimization in x for fixed θ, updates are not "proper" gradients for f

Guarantees under smoothness/convexity assumptions. (Dagreou et al., 22)



Implicit diffusion : joint bilevel on distributions

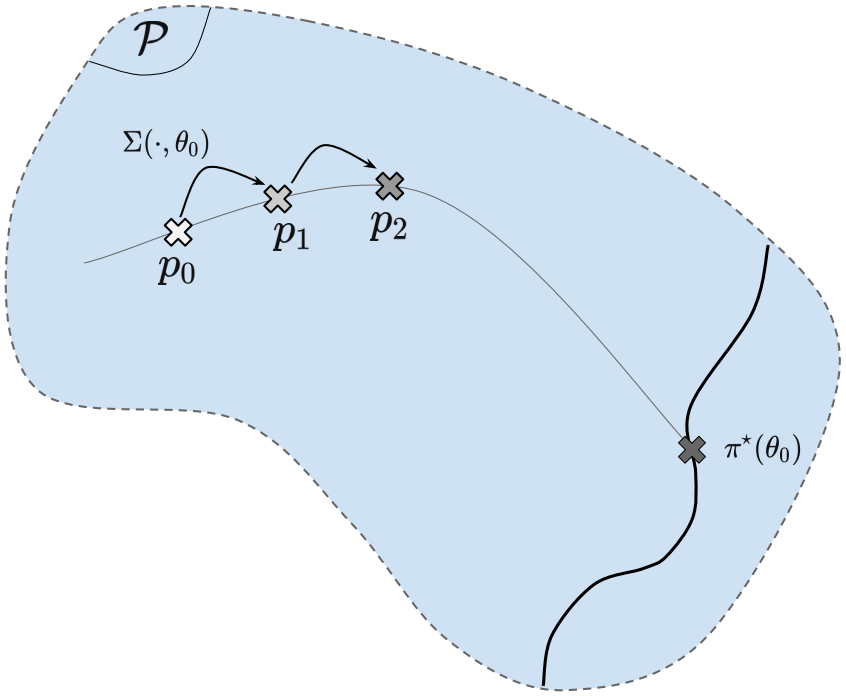
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion.



Implicit diffusion : joint bilevel on distributions

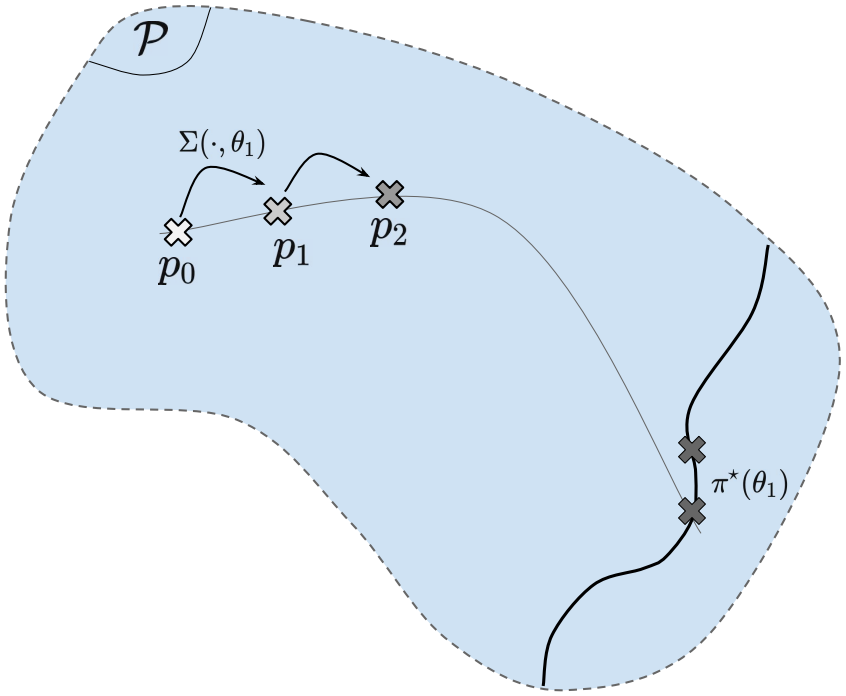
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling in, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion



Implicit diffusion : joint bilevel on distributions

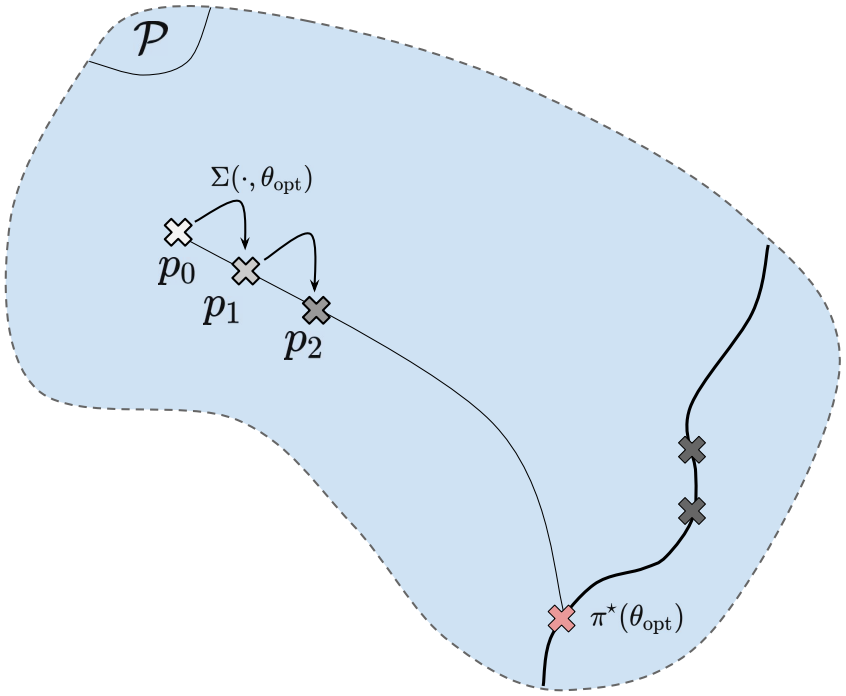
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling in, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion



Implicit diffusion : joint bilevel on distributions

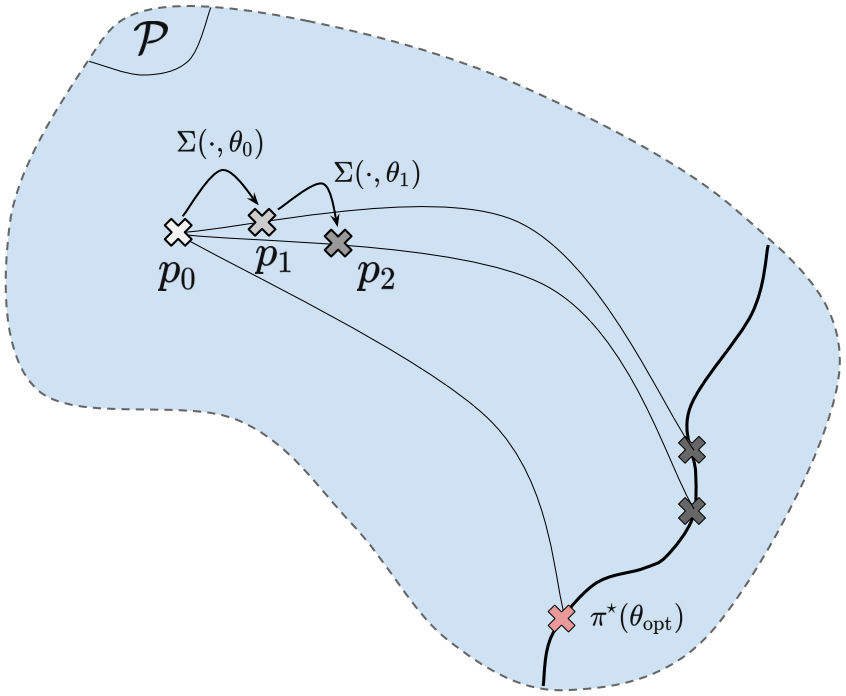
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion.



Implicit diffusion : joint bilevel on distributions

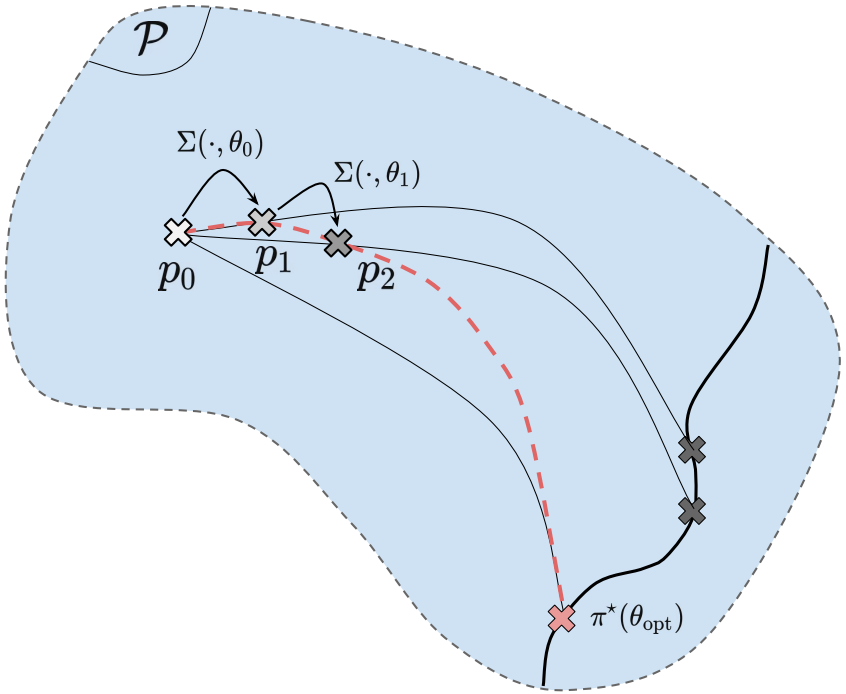
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling in, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion



Implicit diffusion : joint bilevel on distributions

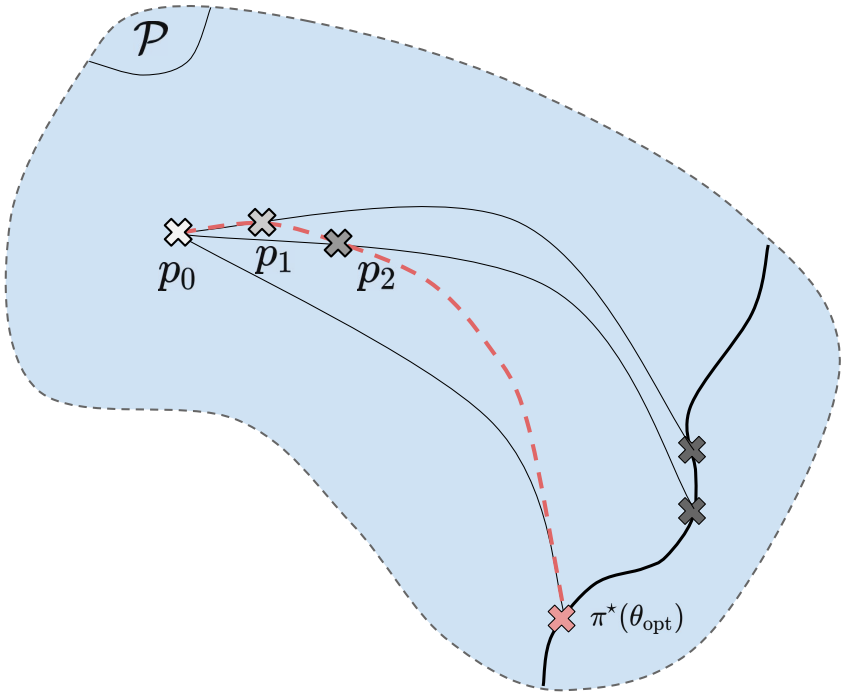
Combining sampling and bilevel optimization

- For $t \geq 0$, with $p_0 \in \mathcal{P}$, $\theta_0 \in \mathbf{R}^d$
  - **Sampling**: Update $p_{t+1} = \Sigma_t(p_t, \theta_t)$ (Inner problem update)
  - **Optimization**: Update $\theta_{t+1} = \theta_t - \eta_t \Gamma(p_t, \theta_t)$ $(\approx \nabla_\theta \ell(\theta_t))$

Remarks

Never perform a full sampling in, obtaining $\pi^\star(\theta)$ for fixed θ.

Assumption: there is some efficient $\Gamma(p, \theta)$, context dependent e.g. Langevin, denoising diffusion



Method and results - Langevin dynamics

Sampling outputs $\pi^\star(\theta) = \exp(-V(x,\theta))/Z_\theta$

Gradients take special forms in specific cases, e.g. for $\mathcal{F}_{\mathsf{rew}}(p) = -\mathbf{E}_{X\sim p}[R(X)]$

$$\nabla \ell_{\mathsf{rew}}(\theta) = \mathsf{Cov}_{X\sim\pi^\star(\theta)}[R(X), \nabla_2 V(X,\theta)]\,,$$
$$\Gamma_{\mathsf{rew}}(p,\theta) = \mathsf{Cov}_{X\sim p}[R(X), \nabla_2 V(X,\theta)]\,.$$

For $\mathcal{F}_{\mathsf{ref}}(p) = \mathsf{KL}(p, p_{\mathsf{ref}})$

$$\nabla \ell_{\mathsf{ref}}(\theta) = \mathbf{E}_{X\sim p_{\mathsf{ref}}}[\nabla_2 V(X,\theta)] - \mathbf{E}_{X\sim\pi^\star(\theta)}[\nabla_2 V(X,\theta)].$$
$$\Gamma_{\mathsf{ref}}(p,\theta) := \mathbf{E}_{X\sim p_{\mathsf{ref}}}[\nabla_2 V(X,\theta)] - \mathbf{E}_{X\sim p}[\nabla_2 V(X,\theta)]\,.$$

Remarks - Can be estimated from samples, used in Implicit Diffusion algorithm

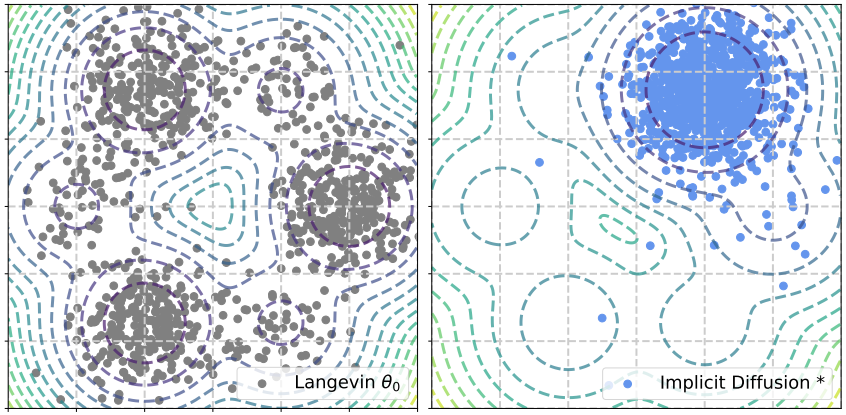
Can be efficiently used, with first order access to V and zero-th order access to R.

(De Bortoli et al, 21)

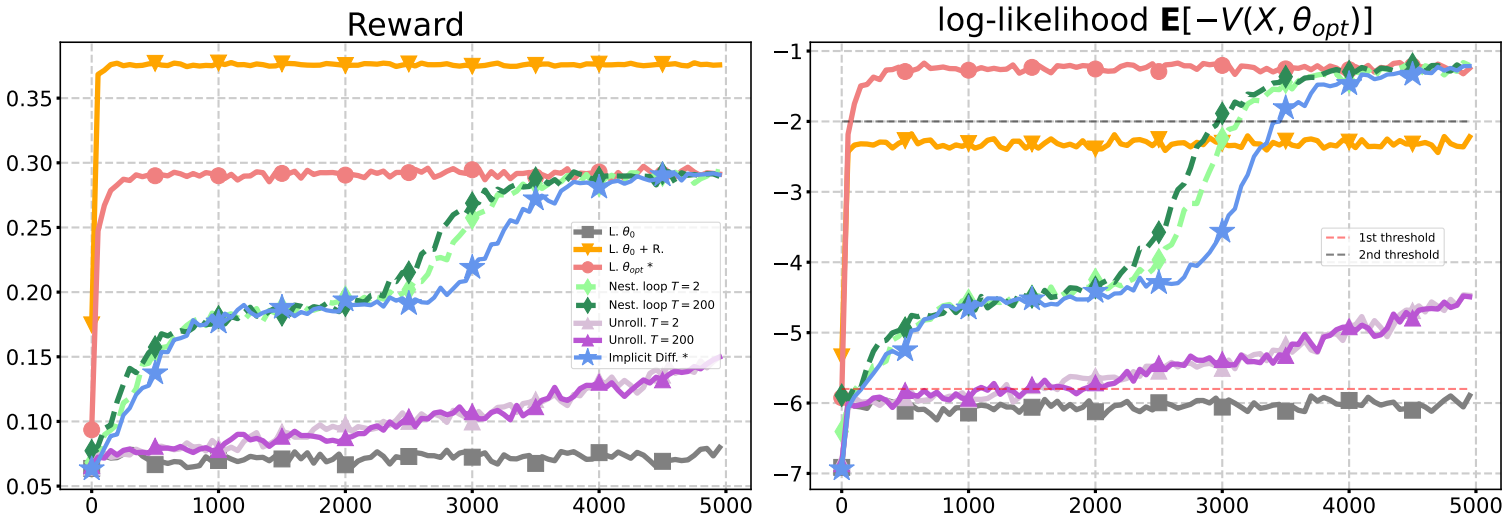
Experiments - Langevin dynamics

Illustration - on mixture of Gaussians in 2D, optimizing on weights.

Initialized at $p_0 \sim \mathcal{N}(0, I_d)$, some $\theta_0 \in \mathbf{R}^6$, and some reward function R.
Single large batch of particles, over time.



Favorable comparison to other approaches, variants with several inner steps.



Theoretical guarantees - Langevin dynamics

Smoothness and Log-Sobolev assumptions on V, π^*, and Γ.

- **Continuous flow** - Implicit diffusion algorithm represented by SDE

$$\begin{aligned} \mathrm{d}X_t &= -\nabla_1 V(X_t, \theta_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,, \\ \mathrm{d}\theta_t &= -\varepsilon_t \Gamma(p_t, \theta_t)\,\mathrm{d}t\,. \end{aligned}$$

- **Discrete flow** - for step sizes γ_k

$$\begin{aligned} X_{k+1} &= X_k - \gamma_k \nabla_1 V(X_k, \theta_k) + \sqrt{2\gamma_k}\Delta B_{k+1}\,, \\ \theta_{k+1} &= \theta_k - \gamma_k \varepsilon_k \Gamma(p_k, \theta_k)\,. \end{aligned}$$

Average of gradient norms objective (stationarity) - slow-fast analysis

$$\frac{1}{T}\int_0^T \|\nabla \ell(\theta_t)\|^2\,\mathrm{d}t \le \frac{c(\ln T)^2}{T^{1/2}}\,, \quad \frac{1}{K}\sum_{k=1}^K \|\nabla \ell(\theta_k)\|^2 \le \frac{c_2 \ln K}{K^{1/3}}\,.$$

(Vempala and Wibisino 19, Cheng and Bartlett 18, Arbel and Mairal 22, Marion and Berthier 23)

Method and results - Denoising diffusion

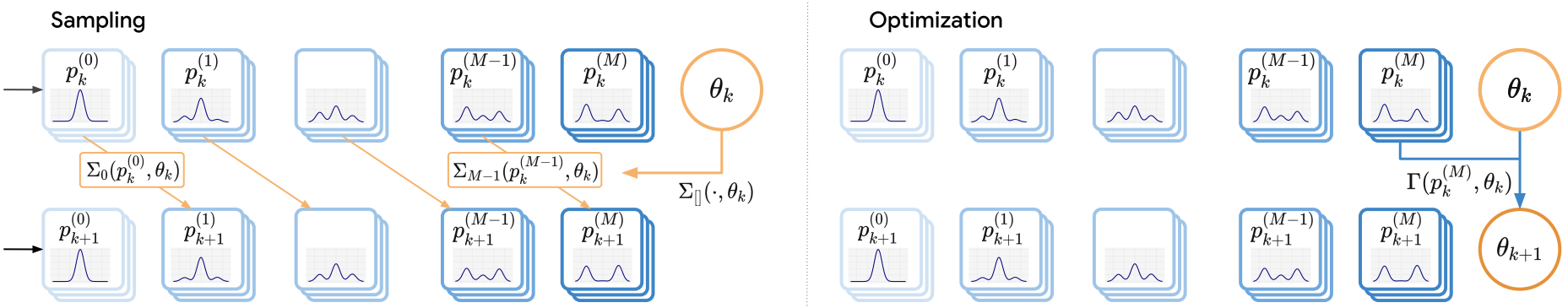
SDE between 0 and T, with $Y_0 \sim \mathcal{N}(0, I_d)$ and $\mathcal{F}(p) = -\mathbf{E}_{X\sim p}[R(X)]$

$$\mathrm{d}Y_t = \mu(t, Y_t, \theta)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,,$$

No obvious shortcut to compute gradients use of adjoint method

$$A_0 = \nabla R(Y_T)\,, \qquad \mathrm{d}A_t = A_t^\top \nabla_2 \mu(T-t, Y_{T-t}, \theta)\,\mathrm{d}t\,,$$
$$G_0 = 0\,, \qquad \mathrm{d}G_t = A_t^\top \nabla_3 \mu(T-t, Y_{T-t}, \theta)\,\mathrm{d}t\,.$$

Gradient $\nabla_\theta \ell(\theta) = G_T$, requires simulation both ways, parallelism tricks

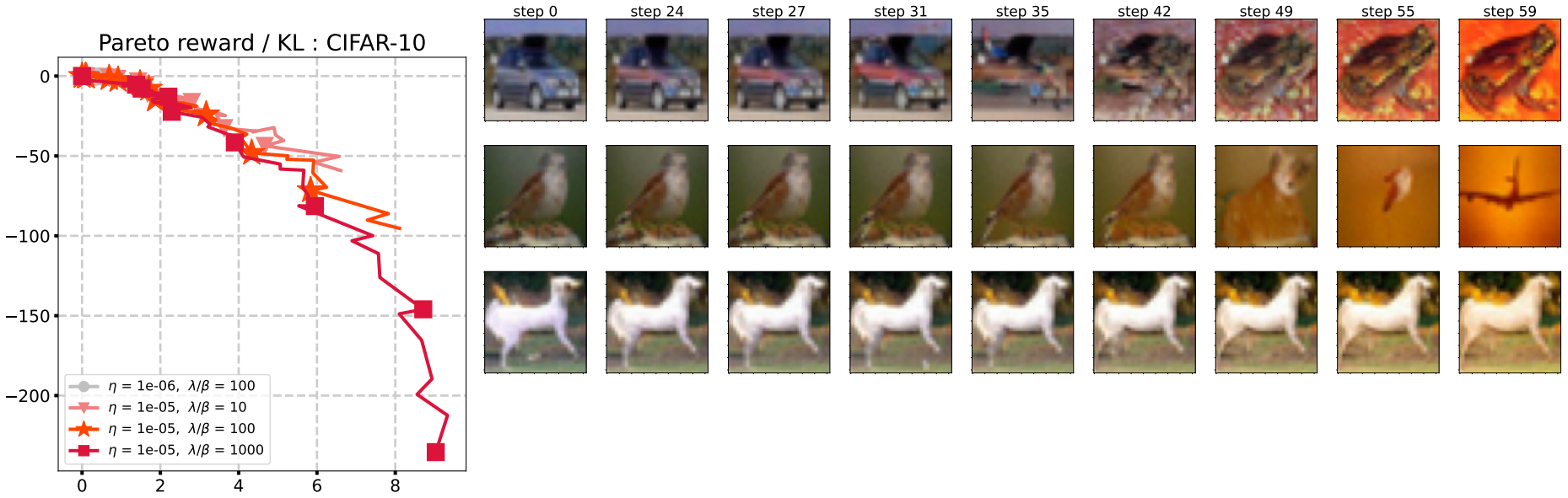


Experimental results - Denoising diffusion

Model pretrained on score matching, on p_{data} has weights θ_0

Objective driven by any reward R and KL term - no need to see data

$$\mathcal{F}(\pi) := -\lambda \mathbf{E}_{x\sim\pi}[R(x)] + \beta \mathsf{KL}(\pi, \pi^\star(\theta_0))\,.$$



Reward **red brightness**, model pre-trained on **CIFAR-10**, biases the distribution.

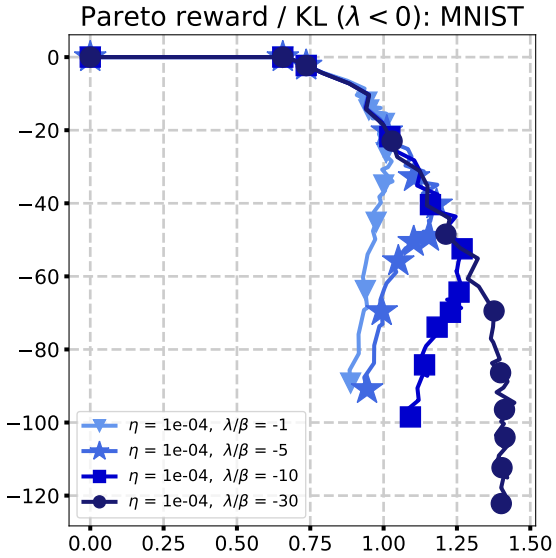
Trade-off between reward and KL regularization term, depending on λ/β.

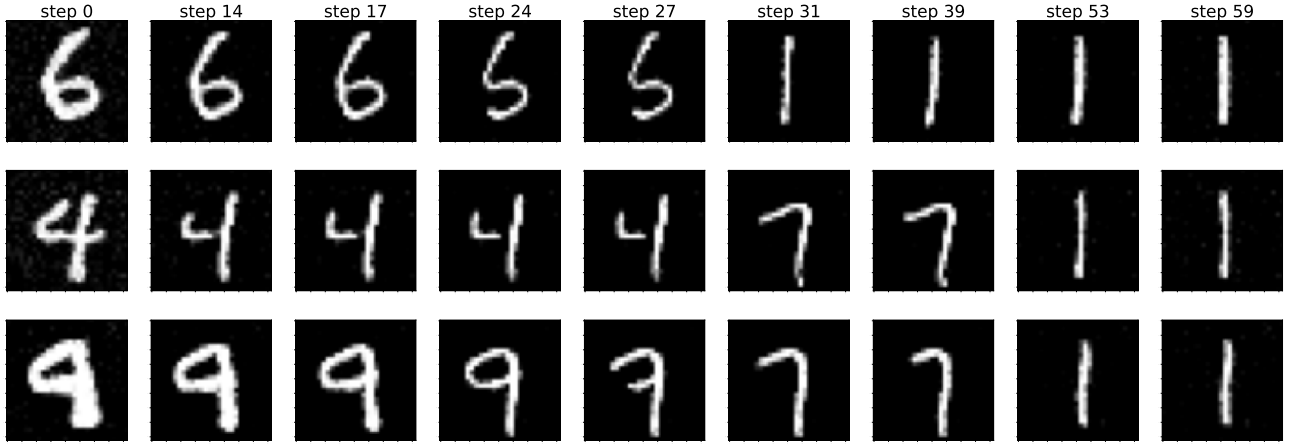
Experimental results - Denoising diffusion

Model pretrained on score matching, on p_{data} has weights θ_0

Objective driven by any reward R and KL term - no need to see data

$$\mathcal{F}(\pi) := -\lambda \mathbf{E}_{x\sim\pi}[R(x)] + \beta \mathsf{KL}(\pi, \pi^\star(\theta_0))\,.$$





Reward **brightness**, model pre-trained on **MNIST**, biases the distribution.

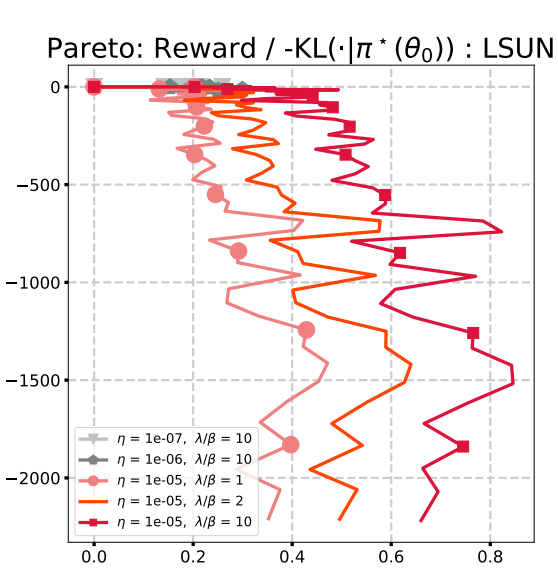
Trade-off between reward and KL regularization term, depending on λ/β.

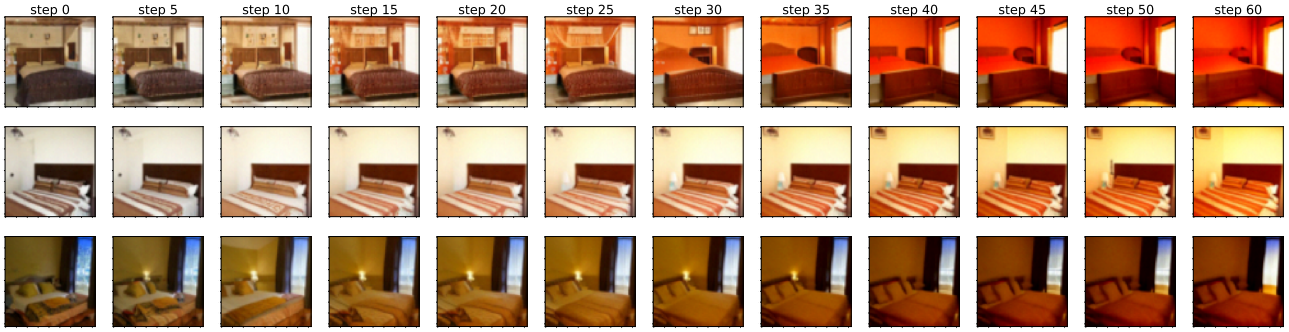
Experimental results - Denoising diffusion

Model pretrained on score matching, on p_{data} has weights θ_0

Objective driven by any reward R and KL term - no need to see data

$$\mathcal{F}(\pi) := -\lambda \mathbf{E}_{x\sim\pi}[R(x)] + \beta \mathsf{KL}(\pi, \pi^\star(\theta_0))\,.$$





Reward **red brightness**, model pre-trained on **LSUN**, biases the distribution.

Trade-off between reward and KL regularization term, depending on λ/β.

Implicit diffusion

- Algorithm for efficient optimization through sampling.
- Part of a literature on single-loop joint optimization, with a single inner step.

  ((Guo et al., 21; Yang et al., 21; Chen et al., 22; Dagréou et al., 22; Hong et al., 23)
- Growing literature on fine-tuning diffusion models

  (Dvijotham et al. 23, Fan et al. 23, Clarck et al., 24, Black et al. 24)