Algorithms for mean-field variational inference via polyhedral optimization in the Wasserstein space

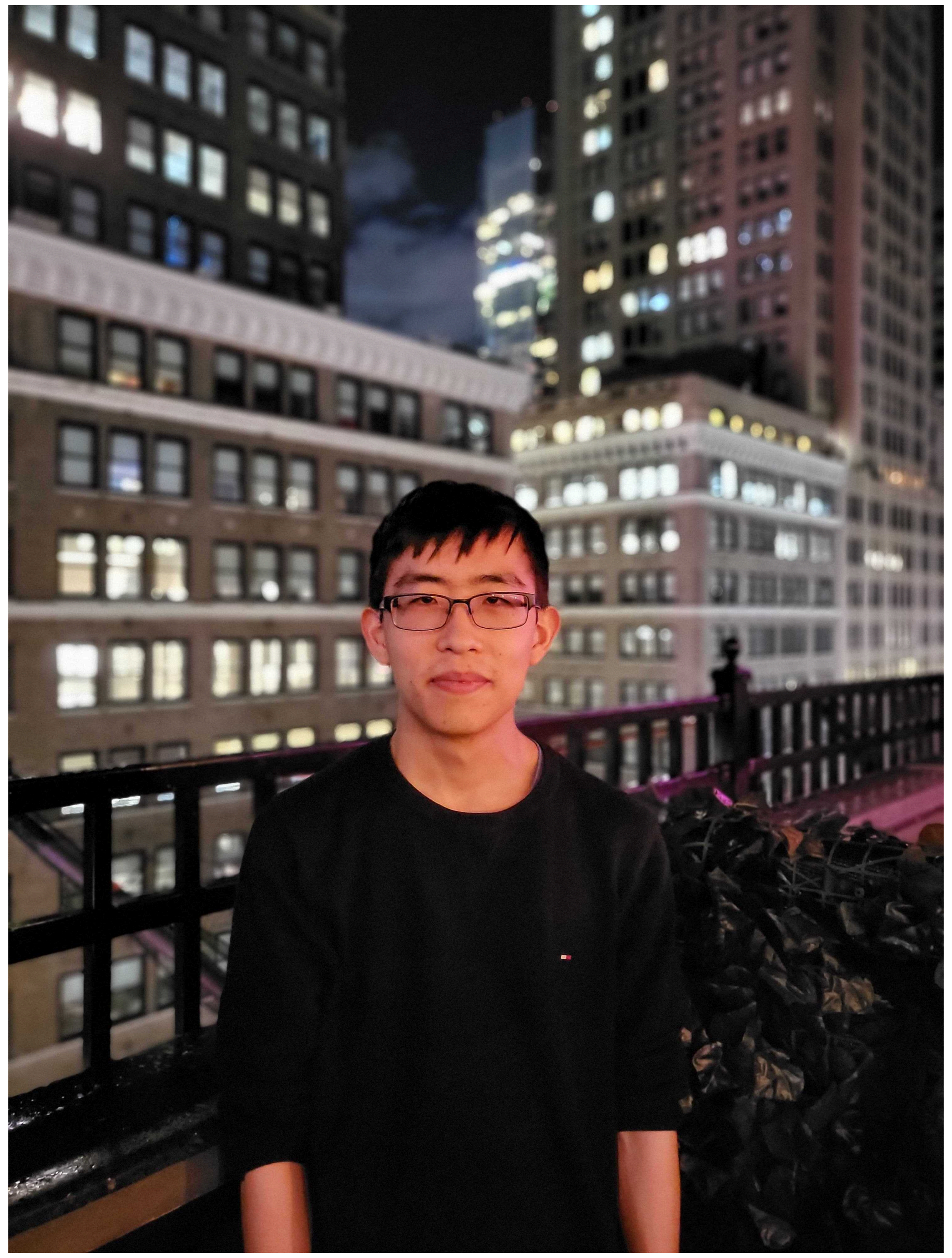
Aram-Alexandre Pooladian
New York University

ETH (DACO Seminar)

March 8, 2024

Joint work with

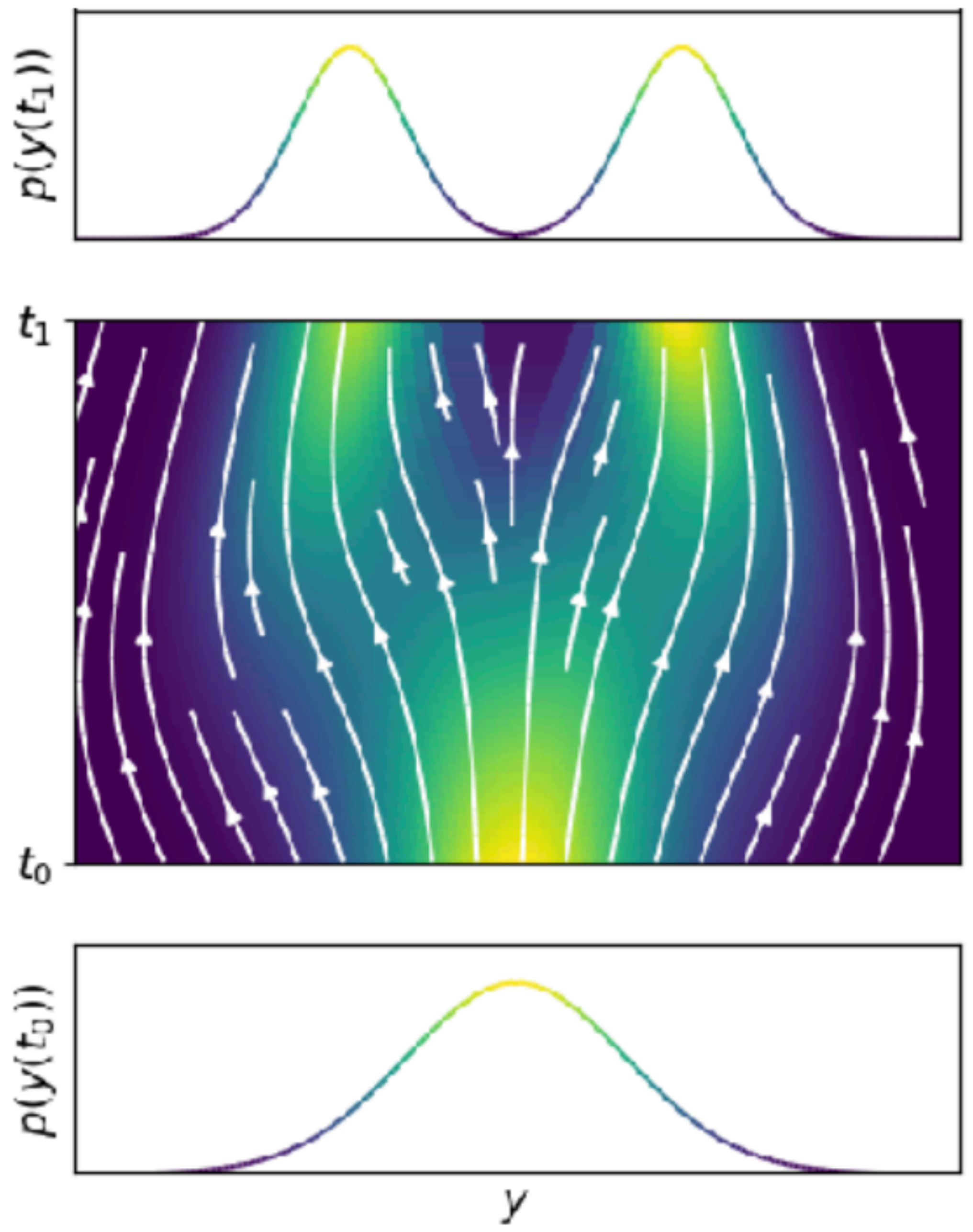
Roger Jiang
NYU



Sinho Chewi
IAS

What to expect in this talk

A principled algorithm for mean-field variational inference
with convergence guarantees



Generative modeling

Optimal transport

Sampling from posterior measures

Task: given $\pi \propto e^{-V}$, draw samples from π to estimate parameters

Method (a): Langevin Monte Carlo (LMC)

Variational Inference

Task: given $\pi \propto e^{-V}$, draw samples from $\pi^\star$ where

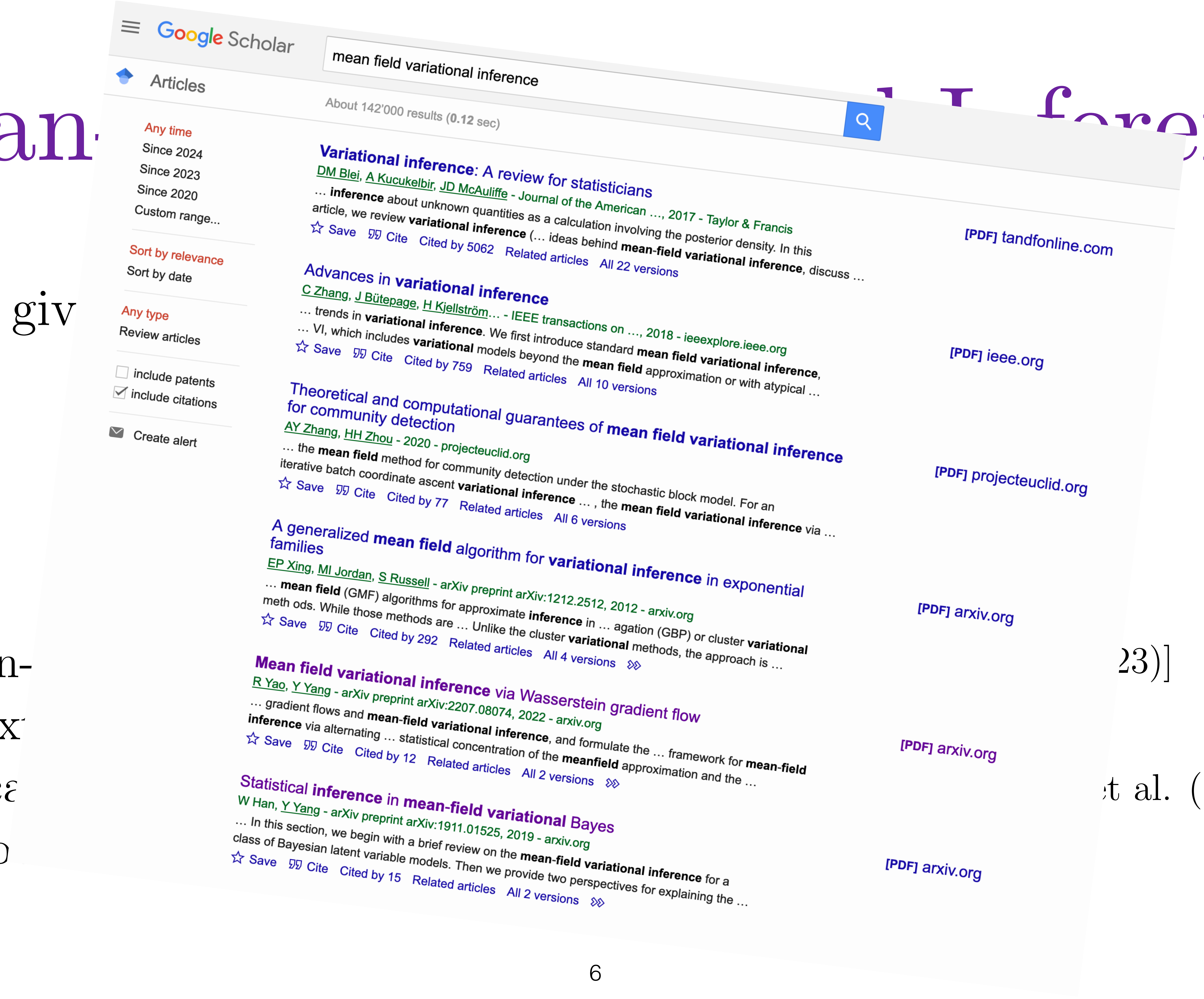
$$\pi^\star \in \arg\min_{\mu \in \mathcal{C}} \mathrm{KL}(\mu \| \pi) = \arg\min_{\mu \in \mathcal{C}} \int \log\left(\frac{\mathrm{d}\mu}{\mathrm{d}\pi}\right) \mathrm{d}\mu$$

where $\mathcal{C}$ is a family of probability measures

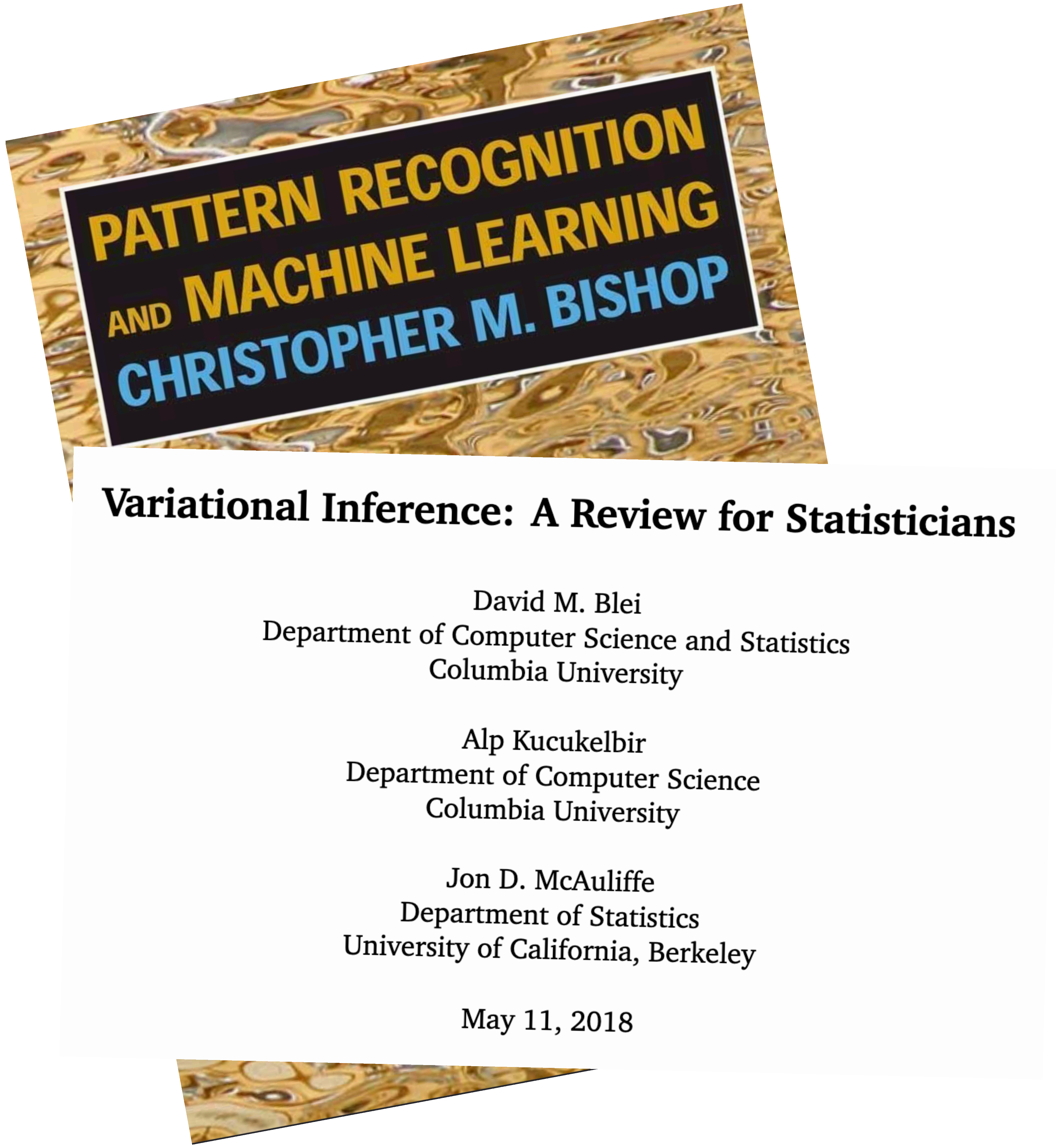
Mean- ... ference

Task: giv

- non- ... 23)]
- mix
- loca ... t al. (2023)]
- pro



Few existing guarantees for VI



Variational Inference: A Review for Statisticians

David M. Blei
Department of Computer Science and Statistics
Columbia University

Alp Kucukelbir
Department of Computer Science
Columbia University

Jon D. McAuliffe
Department of Statistics
University of California, Berkeley

May 11, 2018

- VI is a widely used computational paradigm

the *Wasserstein geometry*

Optimal transport and Wasserstein geometry

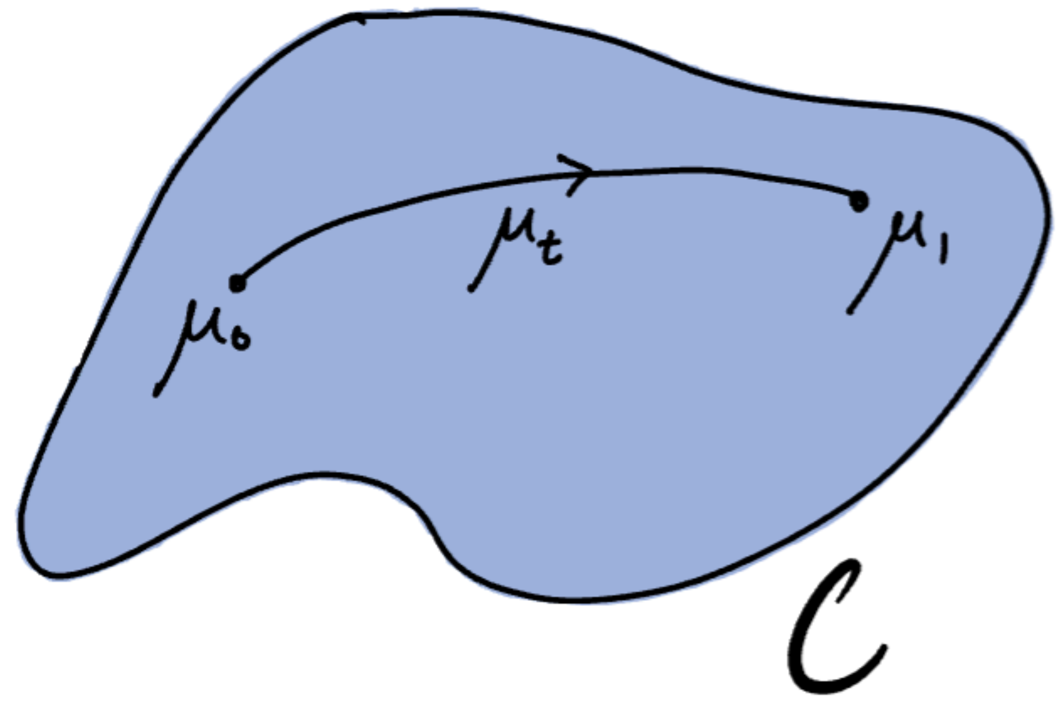
Optimal transport map $T^{0\to 1} := \underset{T\in\mathcal{T}(p_0,p_1)}{\operatorname{argmin}} \|\mathrm{id} - T\|^2_{L^2(p_0)}$

Gradient of a convex function
[Brenier (1991)]

$\mathcal{T}(p_0, p_1) = \{T : T_\sharp p_0 = p_1\}$
i.e., for $X \sim p_0, T(X) \sim p_1$

Wasserstein distance $W_2^2(p_0, p_1) := \|\mathrm{id} - T^{0\to 1}\|^2_{L^2(p_0)}$

W_2-geodesically convex sets



Paths in W_2
[McCann (1997)]

$$p_t := ((1-t)\mathrm{id} + tT^{0\to 1})_\sharp p_0 \in \mathcal{C}$$

(examples include Gaussians and space of product measures)

Current algorithms for MF-VI

Recall $\pi^\star(\theta_1, \ldots, \theta_d) = (\pi_1^\star(\theta_1), \ldots, \pi_d^\star(\theta_d)) = \otimes_{i=1}^d \pi_i^\star(\theta_i)$

<u>Implementation issues</u>

- Requires conjugacy priors
- Problem becomes *parametric*

- Particle approximations...
- Neural networks....

Can we implement an algorithm that (better) exploits the Wasserstein geometry?

Optimization over product measures

To compute $\pi^\star = \arg\min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \mathrm{KL}(\mu \,\|\, \pi)$ $\xrightarrow{\text{gradient flow}}$ $\partial_t \mu_t = \text{“} - \nabla_{\mathbb{W}} \mathrm{KL}(\mu_t \,\|\, \pi)\big|_{\mathcal{P}(\mathbb{R})^{\otimes d}}\text{”}$

$\nabla^2 V \succeq \alpha I \implies \mathrm{KL}(\cdot \,\|\, \pi)$ is α-strongly (geod.) convex over $\mathcal{P}(\mathbb{R})^{\otimes d}$ (See Lacker (2023))

Problem: hard to implement gradient flows over probability measures!

Inspiration from generative modeling

At the end of the day, we just want *samples* from $\pi^\star$

Inspired by generative modeling, we want to find $T : \mathbb{R}^d \to \mathbb{R}^d$ such that

$$\text{for } X \sim \rho,\ T(X) \sim \pi^\star \qquad (\text{e.g., } \rho = \mathcal{N}(0, I))$$

Optimal transport provides a canonical choice for the map:

$$\begin{aligned} T^\star(x) &= (T_1^\star(x_1), \ldots, T_d^\star(x_d)) \\ &= ((\varphi_1^\star)'(x_1), \ldots, (\varphi_d^\star)'(x_d)) \end{aligned}$$

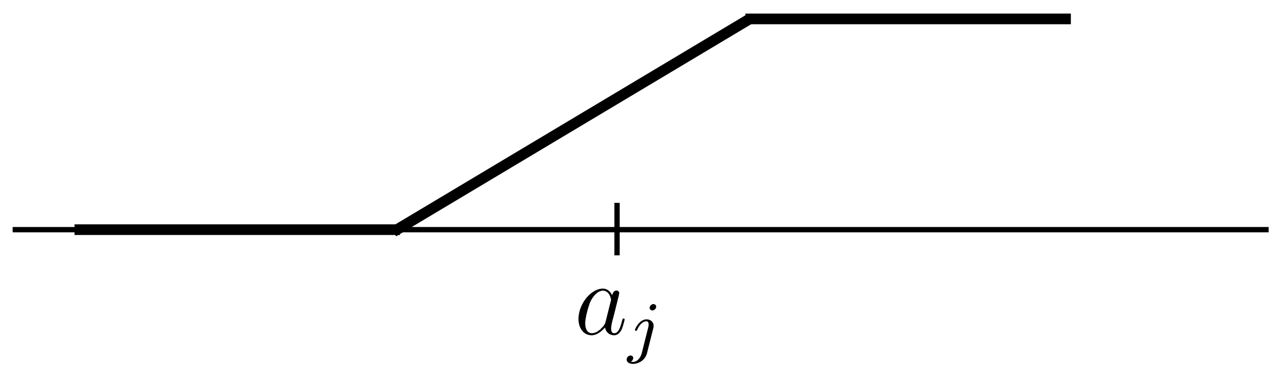
where $\varphi_i^\star$ is some convex function
i.e., $(\varphi_i^\star)'$ is monotone

New goal: find $T^\star$ using only query access to V and ∇V

Mathematical approximation (in 1D)

How to fit $T_1^\star$?

$$\psi_j(x) = \tilde{\psi}(\delta^{-1}(x - a_j))$$



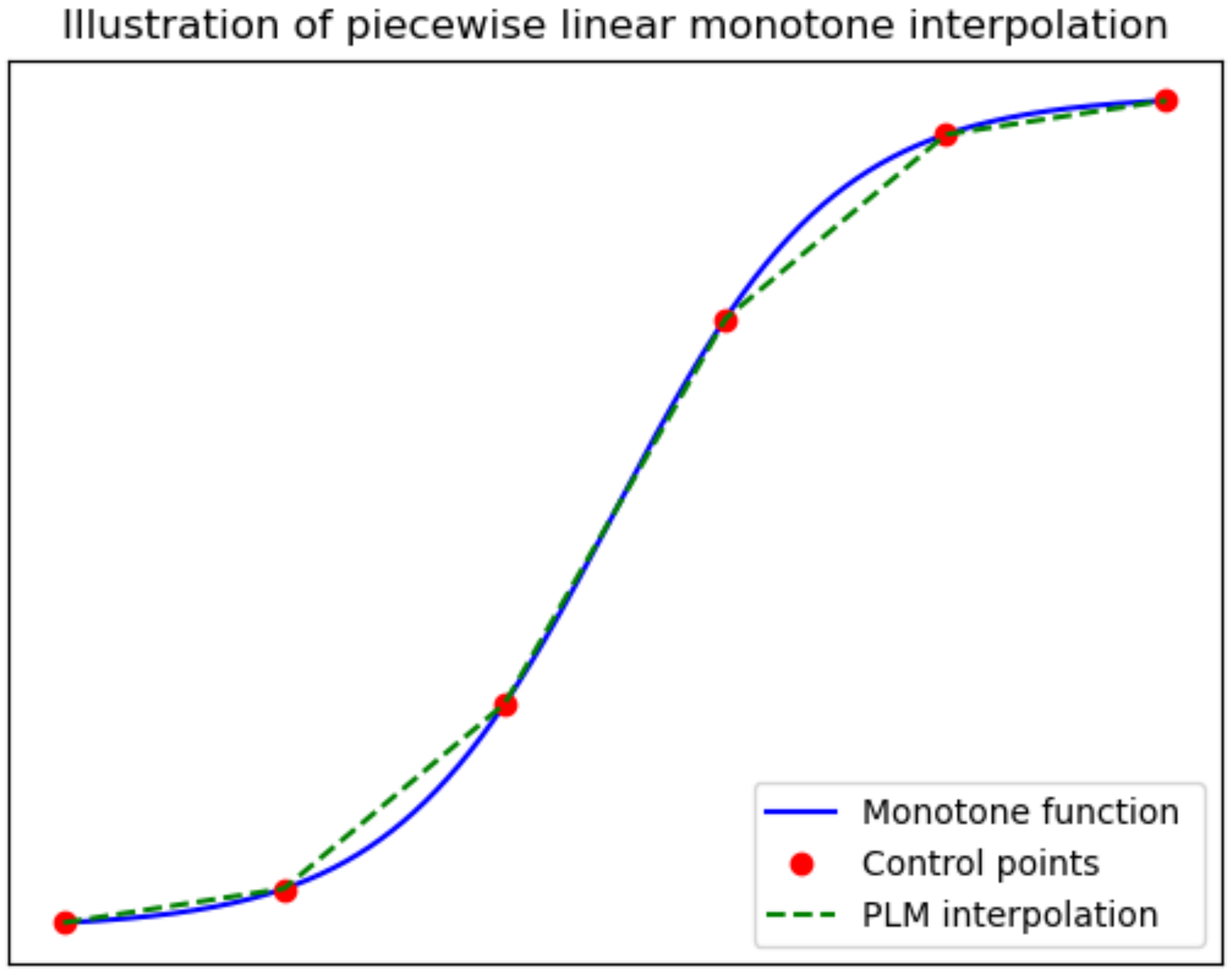
Mathematical approximation (in 1D)

How to fit $T_1^\star$? With piecewise linear monotone functions

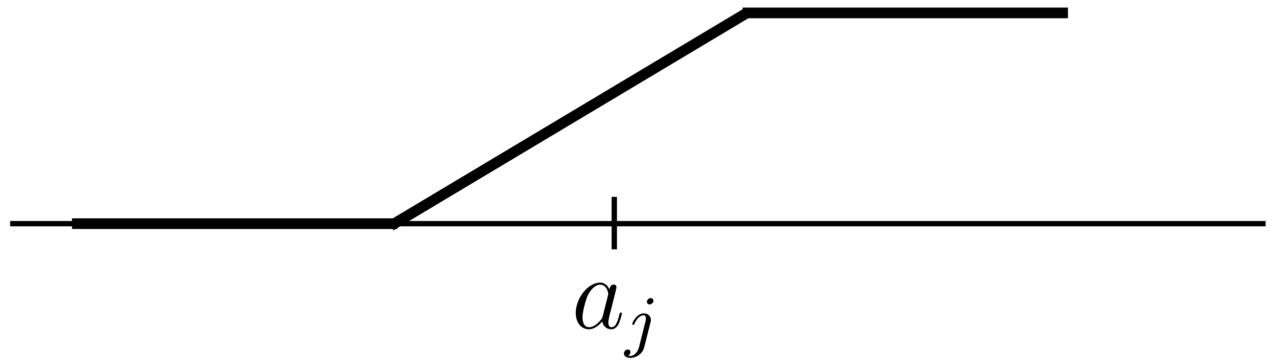
$$T^\lambda(x) = \sum_{j=1}^J \lambda_j \psi_j(x)$$

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$

and $\lambda \in \mathbb{R}_+^J$



$$\psi_j(x) = \tilde{\psi}(\delta^{-1}(x - a_j))$$



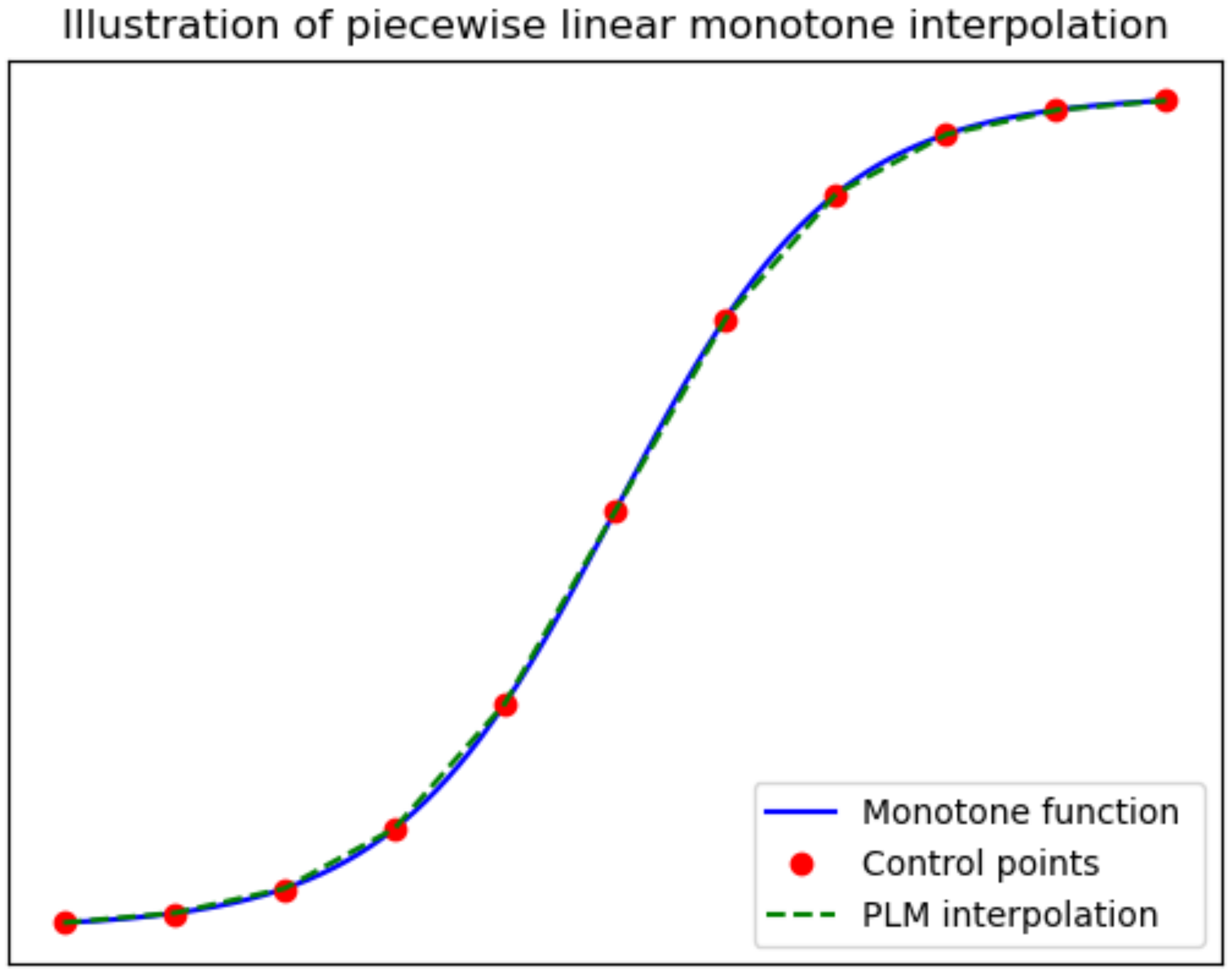
Mathematical approximation (in 1D)

How to fit $T_1^\star$? With piecewise linear monotone functions

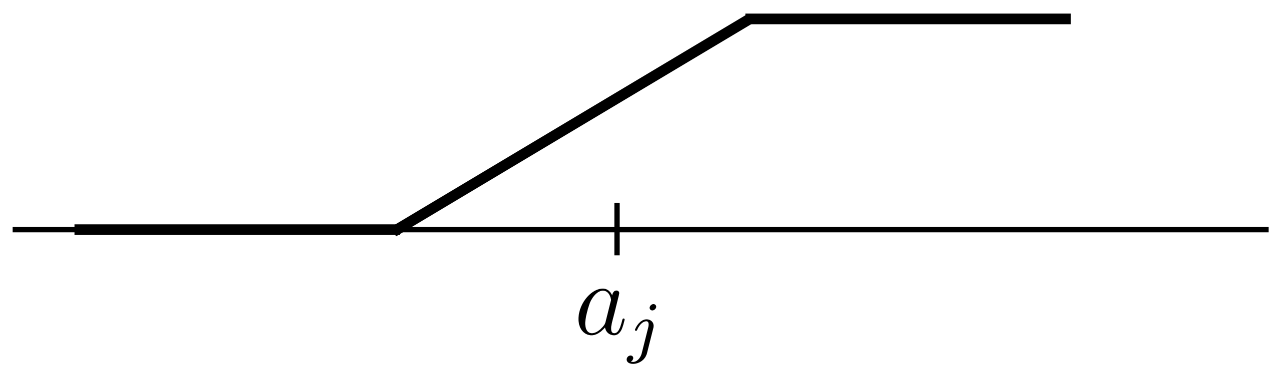
$$T^{\hat{\lambda}}(x) = \sum_{j=1}^{J} \hat{\lambda}_j \psi_j(x)$$

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$

and $\lambda \in \mathbb{R}_+^J$



$$\psi_j(x) = \tilde{\psi}(\delta^{-1}(x - a_j))$$



Mathematical approximation (higher dim.)

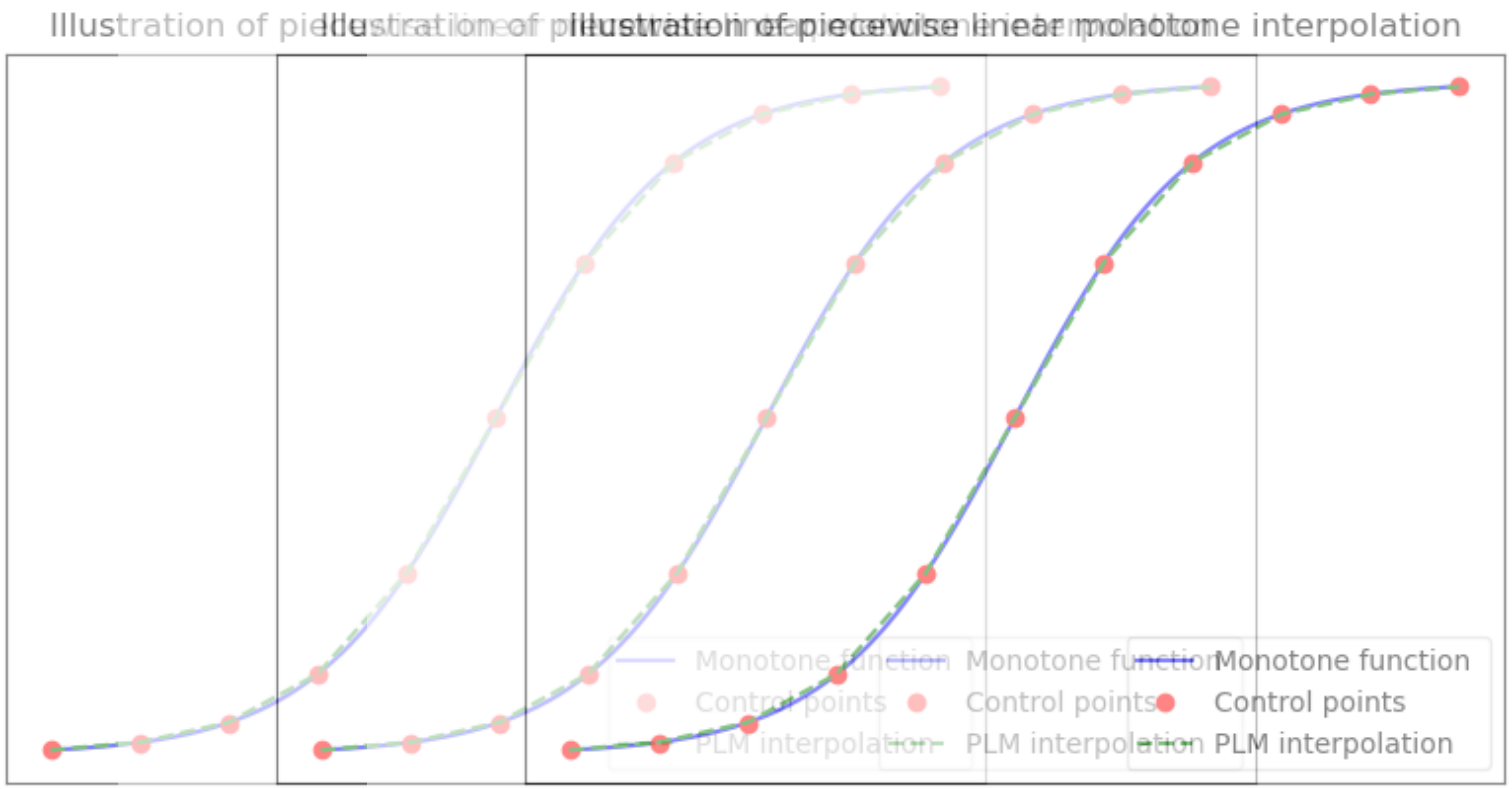
In higher dimensions, there is a natural extension:

$$T^{\hat{\lambda}}(x) = \sum_{i=1}^{d} \sum_{j=1}^{J} \hat{\lambda}_{i,j} \psi_j(x_i) e_i$$

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$

and $\lambda \in \mathbb{R}_+^{dJ}$

$$\hat{\pi}_\diamond := (\hat{T}_\diamond)_\sharp \rho := (T^{\hat{\lambda}})_\sharp \rho \in \mathcal{P}(\mathbb{R})^{\otimes d}$$

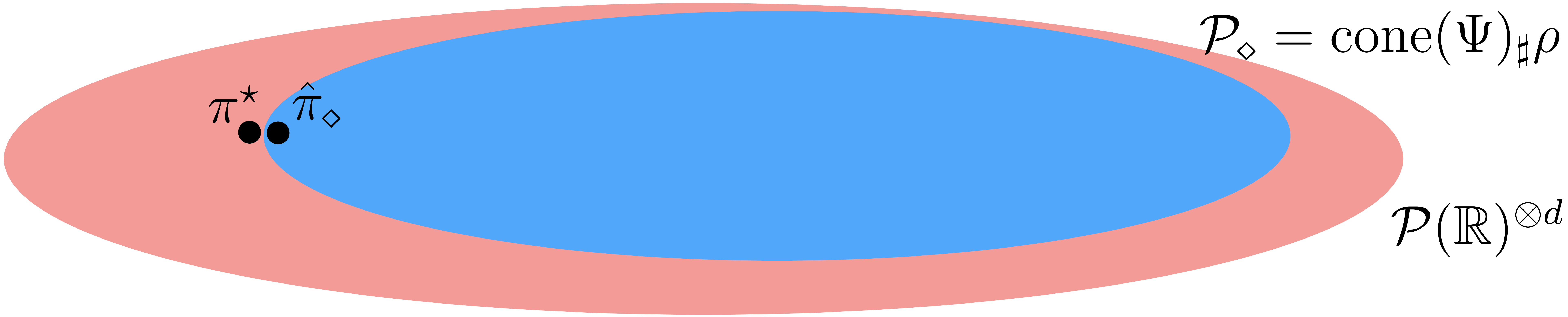


Unfortunately, approximation is not possible

$$\pi^\star = \arg\min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \mathrm{KL}(\mu \,\|\, \pi) \simeq \hat{\pi}_\diamond$$

True for J large enough

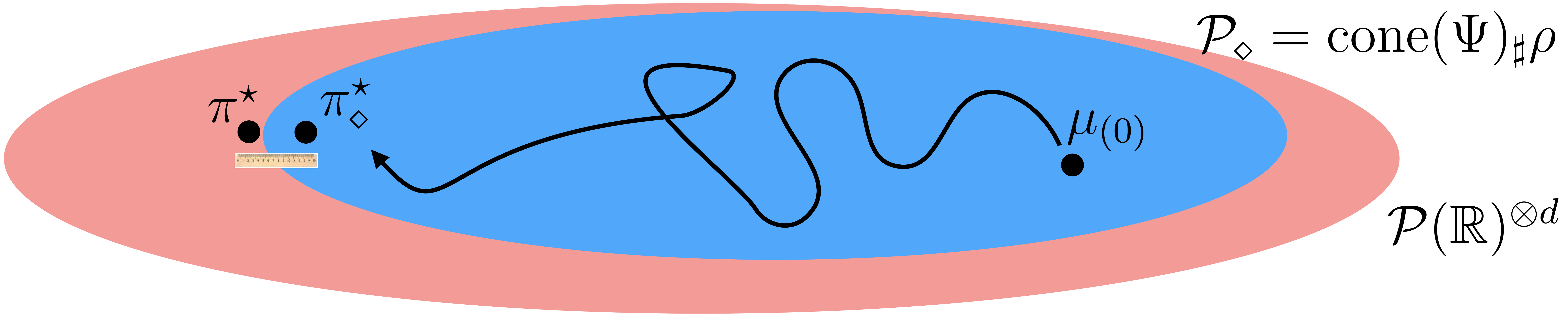
Fitting to $T^\star$ is not possible
(because we don't know what $T^\star$ is!)



$$\mathrm{cone}(\Psi) := \ell x + \sum_{i=1}^{d} \sum_{j=1}^{J} \lambda_{i,j} \psi_j(x_i) e_i\,, \quad \lambda \in \mathbb{R}_+^{dJ}$$

Let's optimize directly over the parameterization

$$\pi^\star = \arg\min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \mathrm{KL}(\mu \,\|\, \pi) \overset{?}{\simeq} \pi^\star_\diamond = \arg\min_{\mu \in \mathcal{P}_\diamond} \mathrm{KL}(\mu \,\|\, \pi)$$



$$\mathrm{cone}(\Psi) := \ell x + \sum_{i=1}^d \sum_{j=1}^J \lambda_{i,j} \psi_j(x_i) e_i\,, \quad \lambda \in \mathbb{R}_+^{dJ}$$

Main results for piecewise linear family

(WC) $\pi \propto e^{-V}$ with $\alpha I \preceq \nabla^2 V \preceq \beta I$ for $\alpha, \beta > 0$, with $\kappa := \beta/\alpha$

Theorem (Approximation). *If* $J = \tilde{O}(\kappa^2 d^{1/2}/\varepsilon)$, $\sqrt{\alpha} W_2(\pi^\star_\diamond, \pi^\star) \le \varepsilon$.

Theorem (Computation). *The number of iterations to find* $\pi^\star_\diamond$ *is* $O(\sqrt{\kappa}\log(\sqrt{\kappa d}/\varepsilon))$.

Properties of pushforward cones

Proposing to solve $\pi_\diamond^\star = \arg\min_{\mu \in \mathcal{P}_\diamond} \mathrm{KL}(\mu \,\|\, \pi) \iff \lambda_\diamond^\star = \arg\min_{\lambda \in \mathbb{R}_+^{dJ}} \mathrm{KL}(\mu_\lambda \,\|\, \pi)$

$\mathrm{cone}(\Psi) \coloneqq \ell x + \sum_{i=1}^{d} \sum_{j=1}^{J} \lambda_{i,j} \psi_j(x_i) e_i \,, \quad \lambda \in \mathbb{R}_+^{dJ}$ and $\mathcal{P}_\diamond = \mathrm{cone}(\Psi)_\sharp \rho$

These properties hold for *polyhedral sets*

Proposing to solve $\pi^\star_\diamond = \arg\min_{\mu \in \mathcal{P}_\diamond} \mathrm{KL}(\mu \,\|\, \pi) \iff \lambda^\star_\diamond = \arg\min_{\lambda \in \mathcal{K}} \mathrm{KL}(\mu_\lambda \,\|\, \pi)$

- **Theorem:** $(\mathcal{P}_\diamond, W_2) \cong (\mathcal{K}, \|\cdot\|_Q)$ with $Q_{ij} = \langle \psi_i, \psi_j \rangle_\rho$

Proof: Let $\mu_\lambda = (T^\lambda)_\sharp \rho\,, \mu_\eta = (T^\eta)_\sharp \rho \in \mathrm{cone}(\Psi)_\sharp \rho$, then

$$W_2^2(\mu_\lambda, \mu_\eta) = \|T^\lambda - T^\eta\|^2_{L^2(\rho)} = \|\sum_{i=1}^d \sum_{j=1}^J (\lambda_{i,j} - \eta_{i,j}) \psi_j e_i\|^2_{L^2(\rho)} = \|\lambda - \eta\|^2_Q$$

- **Corollary:** $\mathcal{P}_\diamond$ is a *geodesically convex set* (optimization is meaningful)

(convex subset)

$\mathrm{cone}(\Psi) := \ell x + \sum_{i=1}^d \sum_{j=1}^J \lambda_{i,j} \psi_j(x_i) e_i\,, \quad \lambda \in \mathcal{K} \subseteq \mathbb{R}^{dJ}_+$ and $\mathcal{P}_\diamond = \mathrm{cone}(\Psi)_\sharp \rho$

How to optimize over pushforward cones

Proposing to solve $\pi^\star_\diamond = \arg\min_{\mu \in \mathcal{P}_\diamond} \mathrm{KL}(\mu \,\|\, \pi) \iff \lambda^\star_\diamond = \arg\min_{\lambda \in \mathbb{R}^{dJ}_+} \mathrm{KL}(\mu_\lambda \,\|\, \pi)$

Gradient flows over polyhedral sets: "$\nabla_{\mathbb{W}} \mathrm{KL}(\mu_t \,\|\, \pi)\big|_{\mathcal{P}_\diamond} = Q^{-1} \nabla_\lambda \mathrm{KL}(\mu_\lambda \,\|\, \pi)$"

Discretizing gradient flows over $\mathcal{P}_\diamond$: $\lambda^{(k+1)} = \mathrm{Proj}_{\mathbb{R}^{dJ}_+, Q}\big(\lambda^{(k)} - hQ^{-1}\nabla_\lambda \mathrm{KL}(\mu_\lambda \| \pi)\big)$

(and with Nesterov momentum!)

Need smoothness and strong convexity for convergence guarantees

Road to convergence guarantees

Strong convexity is free ($\mathcal{P}_\diamond$ is geodesically convex, and $\nabla^2 V \succeq \alpha I$)

Remains to assert that $\lambda \mapsto \mathrm{KL}(\mu_\lambda \,\|\, \pi)$ is ?-smooth and α-strongly convex

$$\mathrm{KL}(\mu_\lambda \,\|\, \pi) = \mathcal{V}(\mu_\lambda) + \mathcal{H}(\mu_\lambda) + \log(Z) = \int V \,\mathrm{d}\mu_\lambda + \int \log \mu_\lambda \,\mathrm{d}\mu_\lambda + \log(Z)$$

- If $\nabla^2 V \preceq \beta I$ then $\lambda \mapsto \mathcal{V}(\mu_\lambda)$ is also β-smooth

$\mathrm{cone}(\Psi) := \ell x + \sum_{i=1}^{d} \sum_{j=1}^{J} \lambda_{i,j} \psi_j(x_i) e_i \,, \quad \lambda \in \mathbb{R}_+^{dJ}$

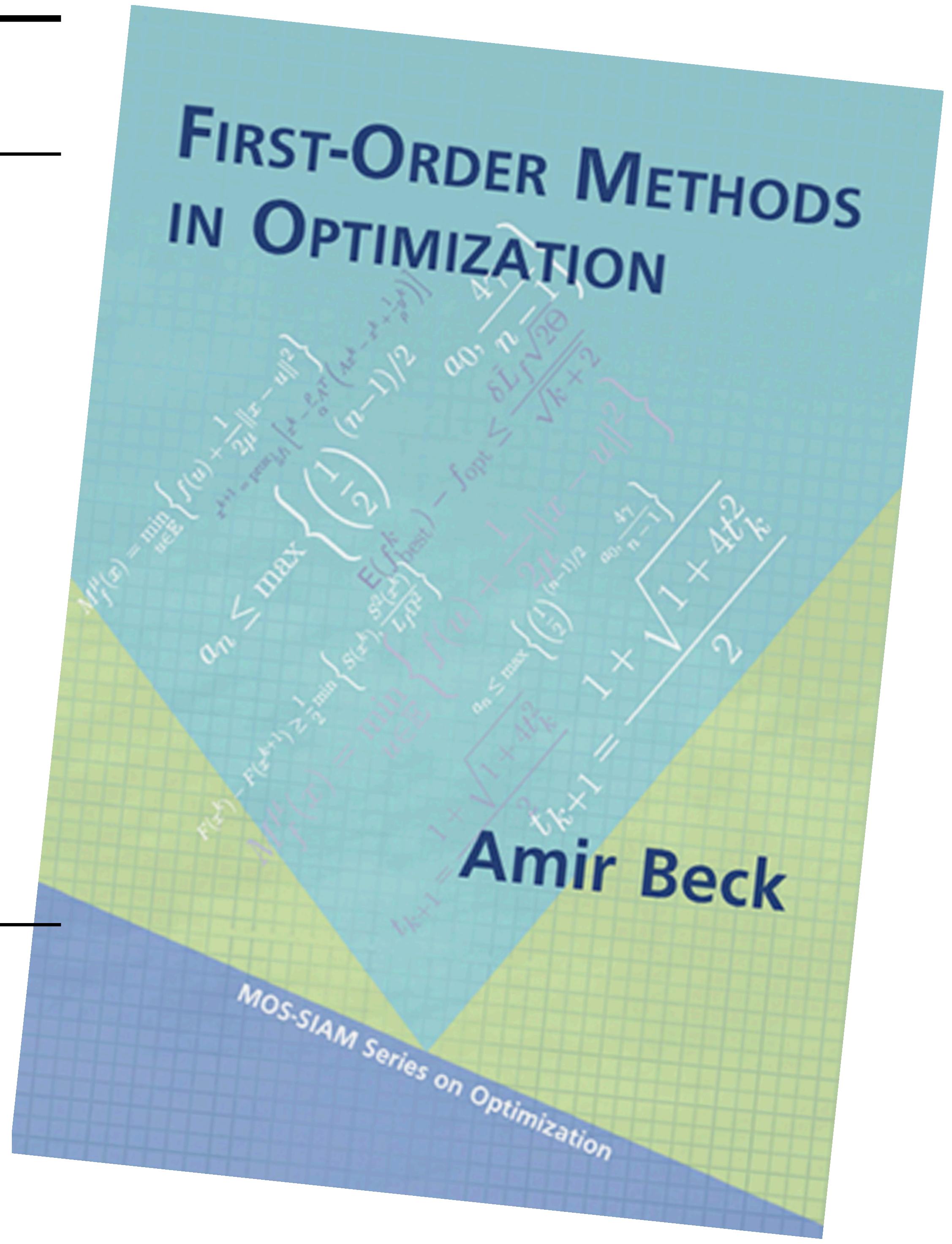
→ Choose $\ell = 1/\sqrt{\beta}$

Accelerated gradient descent for VI

$\lambda \mapsto \mathrm{KL}(\mu_\lambda \,\|\, \pi)$ is $\beta(1+\Upsilon)$-smooth and α-strongly convex w.r.t $(\mathbb{R}_+^{dJ}, \|\cdot\|_Q)$

Algorithm 1 Accelerated projected gradient descent over cone(Ψ)

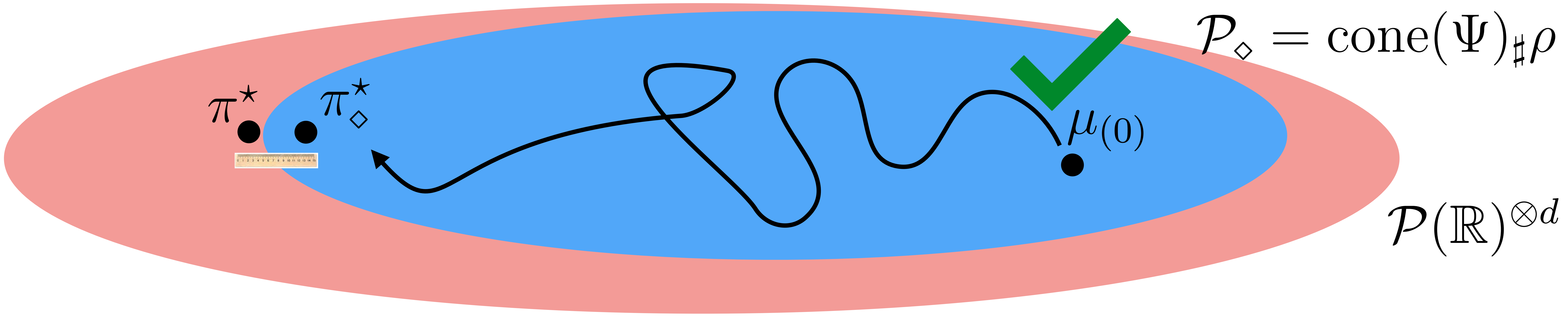
Input: $\lambda^{(0)} \in \mathbb{R}_+^{dJ}$, functional $\mathrm{KL}(\cdot \,\|\, \pi)$
Set $\eta^{(0)} = \lambda^{(0)}$, $\kappa := \beta(1+\Upsilon)/\alpha$
for $t = 0, 1, 2, 3, \ldots$ **do**
$\quad \lambda^{(t+1)} \leftarrow \mathrm{proj}_{\mathbb{R}_+^{dJ}, Q}\left(\eta^{(t)} - \frac{1}{\beta(1+\Upsilon)} Q^{-1} \nabla_\lambda \mathrm{KL}(\mu_{\eta^{(t)}} \,\|\, \pi)\right)$
$\quad \eta^{(t+1)} \leftarrow \lambda^{(t+1)} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}(\lambda^{(t+1)} - \lambda^{(t)})$
end for



Theorem (Computation). *The number of iterations to find* $\pi^\star_\diamond$ *is* $O(\sqrt{\kappa}\log(\sqrt{\kappa d}/\varepsilon))$.

But are these minimizers actually close?

$$\pi^\star = \arg\min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \mathrm{KL}(\mu \,\|\, \pi) \overset{?}{\simeq} \pi^\star_\diamond = \arg\min_{\mu \in \mathcal{P}_\diamond} \mathrm{KL}(\mu \,\|\, \pi)$$



$$\mathrm{cone}(\Psi) := \ell x + \sum_{i=1}^{d} \sum_{j=1}^{J} \lambda_{i,j} \psi_j(x_i) e_i \,, \quad \lambda \in \mathbb{R}_+^{dJ}$$

Quick proof sketch of closeness:

$$W_2^2(\pi_\diamond^\star, \pi^\star) = W_2^2((T_\diamond^\star)_\sharp \rho, (T^\star)_\sharp \rho) = \|T_\diamond^\star - T^\star\|_{L^2(\rho)}^2$$

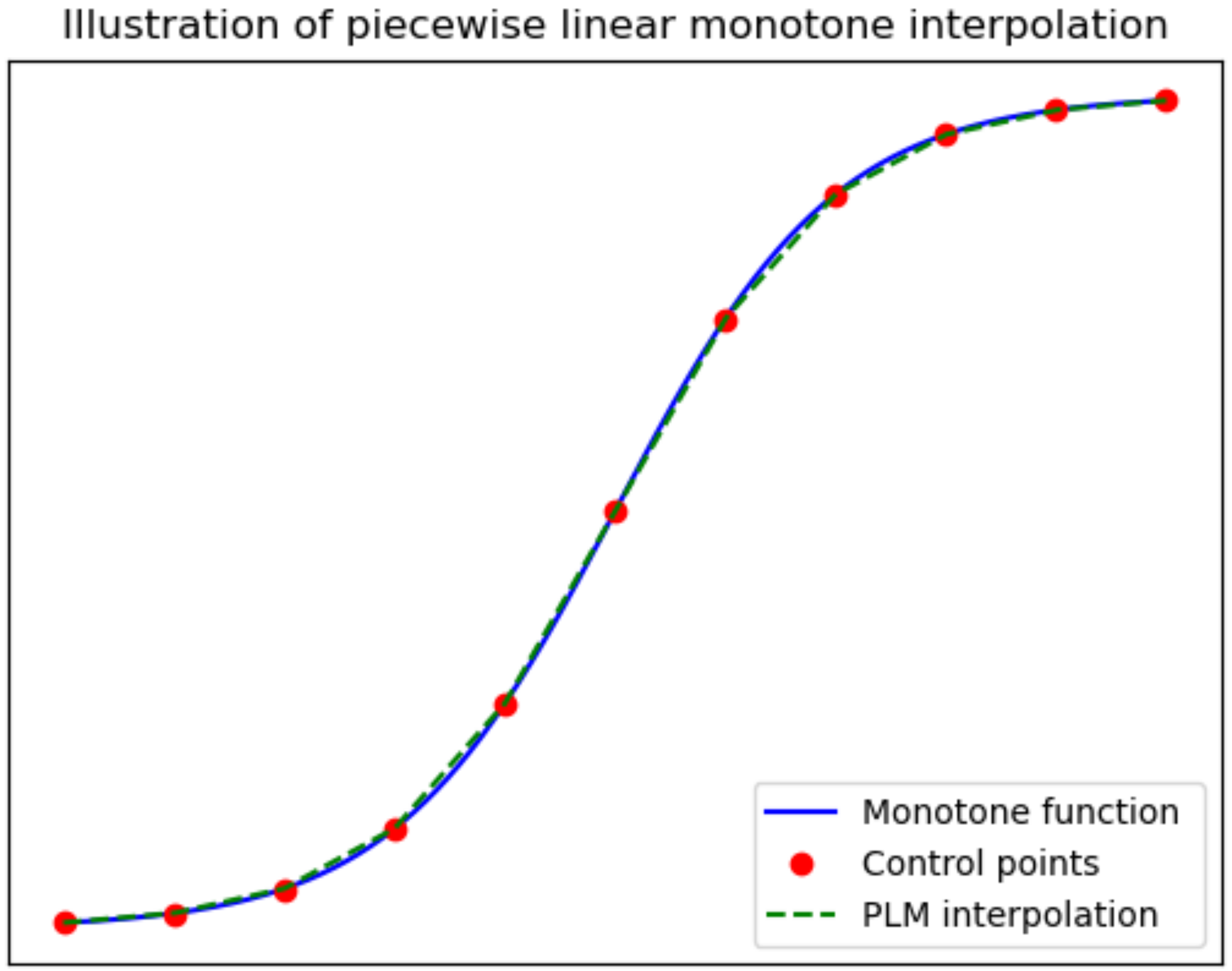
$$\|T_\diamond^\star - T^\star\|_{L^2(\rho)}^2 \lesssim \|T_\diamond^\star - \hat{T}_\diamond\|_{L^2(\rho)}^2 + \|\hat{T}_\diamond - T^\star\|_{L^2(\rho)}^2$$

How close is this approximation?

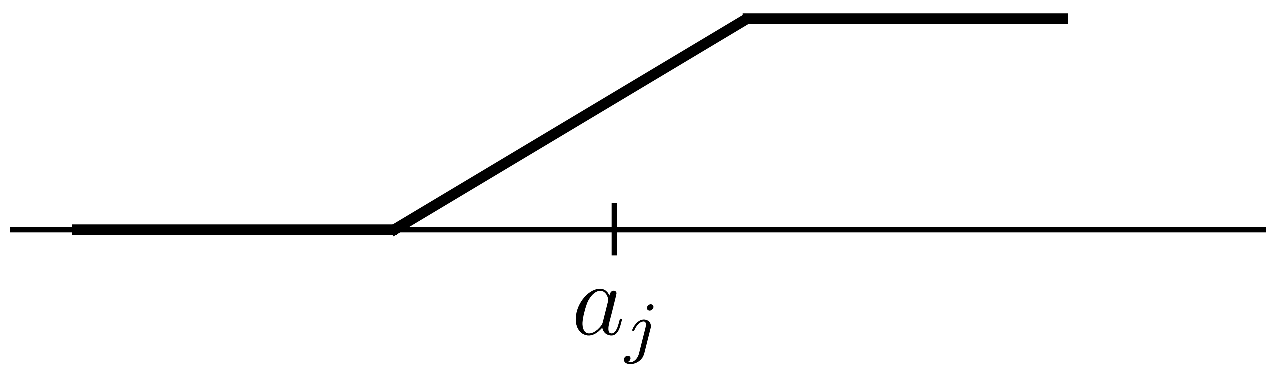
$$T^{\hat{\lambda}}(x) = \sum_{j=1}^{J} \hat{\lambda}_j \psi_j(x)$$

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$

and $\lambda \in \mathbb{R}_+^J$



$$\psi_j(x) = \tilde{\psi}(\delta^{-1}(x - a_j))$$



Quick proof sketch of closeness:

$$W_2^2(\pi_\diamond^\star, \pi^\star) = W_2^2((T_\diamond^\star)_\sharp\rho, (T^\star)_\sharp\rho) = \|T_\diamond^\star - T^\star\|_{L^2(\rho)}^2$$

$$\begin{aligned}\|T_\diamond^\star - T^\star\|_{L^2(\rho)}^2 &\lesssim \|T_\diamond^\star - \hat{T}_\diamond\|_{L^2(\rho)}^2 + \|\hat{T}_\diamond - T^\star\|_{L^2(\rho)}^2 \\ &\leq \kappa^2\|\hat{T}_\diamond - T^\star\|_{H^1(\rho)}^2 + \|\hat{T}_\diamond - T^\star\|_{L^2(\rho)}^2 \\ &\leq \varepsilon + \varepsilon\end{aligned}$$

(smoothness + strong convexity)

(approximation is close to optimal)

Proof requires new regularity properties of the Monge–Ampère equation

(WC) $\Longrightarrow \frac{1}{\sqrt{\beta}} \leq (T_i^\star)' \leq \frac{1}{\sqrt{\alpha}}$

(Caffarelli (2000))

Improvement under smoothness?

Improvement under smoothness!

Theorem (Smoother maps). *There exists a different generating family such that with* $J = \tilde{O}(\kappa^{3/2} d^{1/4} / \varepsilon^{1/2})$, *then it holds that* $\sqrt{\alpha} W_2(\pi^\star_\diamond, \pi^\star) \leq \varepsilon$.

Implementation?

Implementation (yes, we coded it!)

```python
import numpy as np

from optimization_utils import *
from misc_utils import *
from gaussian_utils import rho_gaussian_samples

class MFVI:
    def __init__(self, V, grad_V, mesh, trunc,dim):
        self.V = V
        self.grad_V = grad_V
        self.mesh = mesh
        self.trunc = trunc
        self.dim = dim
        self.J = int(2 * self.trunc / self.mesh)

        self.lamb_opt, self.v_opt, self.T_opt = None, None, None

        self.KL_vals, self.W2_vals = None, None

        self.M_1d, _, self.Q, self.Qinv, means , self.gradent_num = build_M_FAST(dim=self.dim, mesh=self.mesh, truncation=self.trunc)
    def SPGD(self,alpha, h, h_v, lamb0, batch_size=1, num_iters=1000, tol=1e-3, compute_KL=False, compute_W2=False,ground_cov=None,stopping_cond=0,save_vals=
        self.alpha = alpha
        if save_vals:
            self.lamb_opt, self.v_opt, self.KL_vals, self.W2_vals, self.lamb_vals, self.v_vals = spgd(self.M_1d, self.dim, h, h_v, alpha, self.Q, self.Qinv,
                                                                   self.grad_V, self.V, num_iter=num_iters, stochastic_samples = batch_size,
                                                                   compute_KL=compute_KL, compute_W2=compute_W2, ground_cov = ground_cov,KL_tol=tol, stopp
        else:
            self.lamb_opt, self.v_opt, self.KL_vals, self.W2_vals = spgd(self.M_1d, self.dim, h, h_v, alpha, self.Q, self.Qinv, self.gradent_num , lamb0, se
                                                                   self.grad_V, self.V, num_iter=num_iters, stochastic_samples = batch_size,
                                                                   compute_KL=compute_KL, compute_W2=compute_W2, ground_cov = ground_cov,KL_tol=tol, stopp
```

For the full implementation, visit: https://github.com/APooladian/MFVI

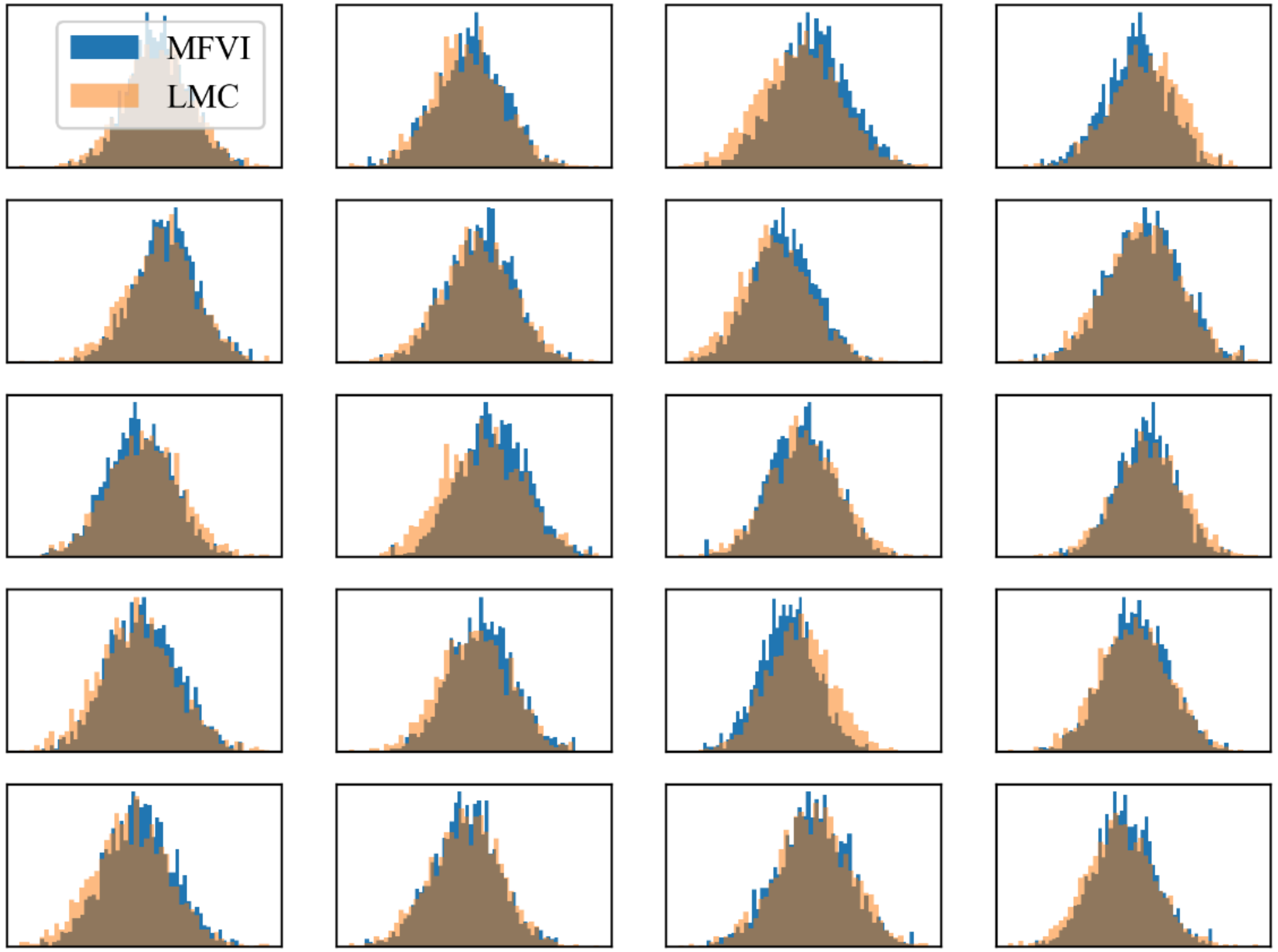
Example: Bayesian Logistic Regression

We generate (for random X_i and θ)

$$Y_i \mid X_i \sim \mathrm{Bern}(\exp(\theta^\top X_i))\,,$$

$$V(\theta) = \sum_{i=1}^{n} \left[\log(1 + \exp(\theta^\top X_i)) - Y_i\, \theta^\top X_i\right].$$

Here, we considered $d = 20$ and $n = 100$.



Visualization of 2000 samples drawn from the posterior using MFVI and LMC

Not strongly log-concave, but it still works!

Recap and more:

Recap:

- "Nonparametric" parameterization of product measures
- Optimization is easy due to isometries; convergence rates are free

More:

Open questions:

- Regarding Wasserstein polyhedra:
  - Investigate statistical convergence guarantees
  - Develop further applications of polyhedral optimization
  - Analogues with other geometries (e.g., sphere?)
- Regarding mean-field VI:
  - Explicitly quantify constants (e.g., Υ)
  - Moving beyond setting where $\nabla^2 V \succeq \alpha I$
  - Other algorithms?

Thank you

Code for repo: