

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



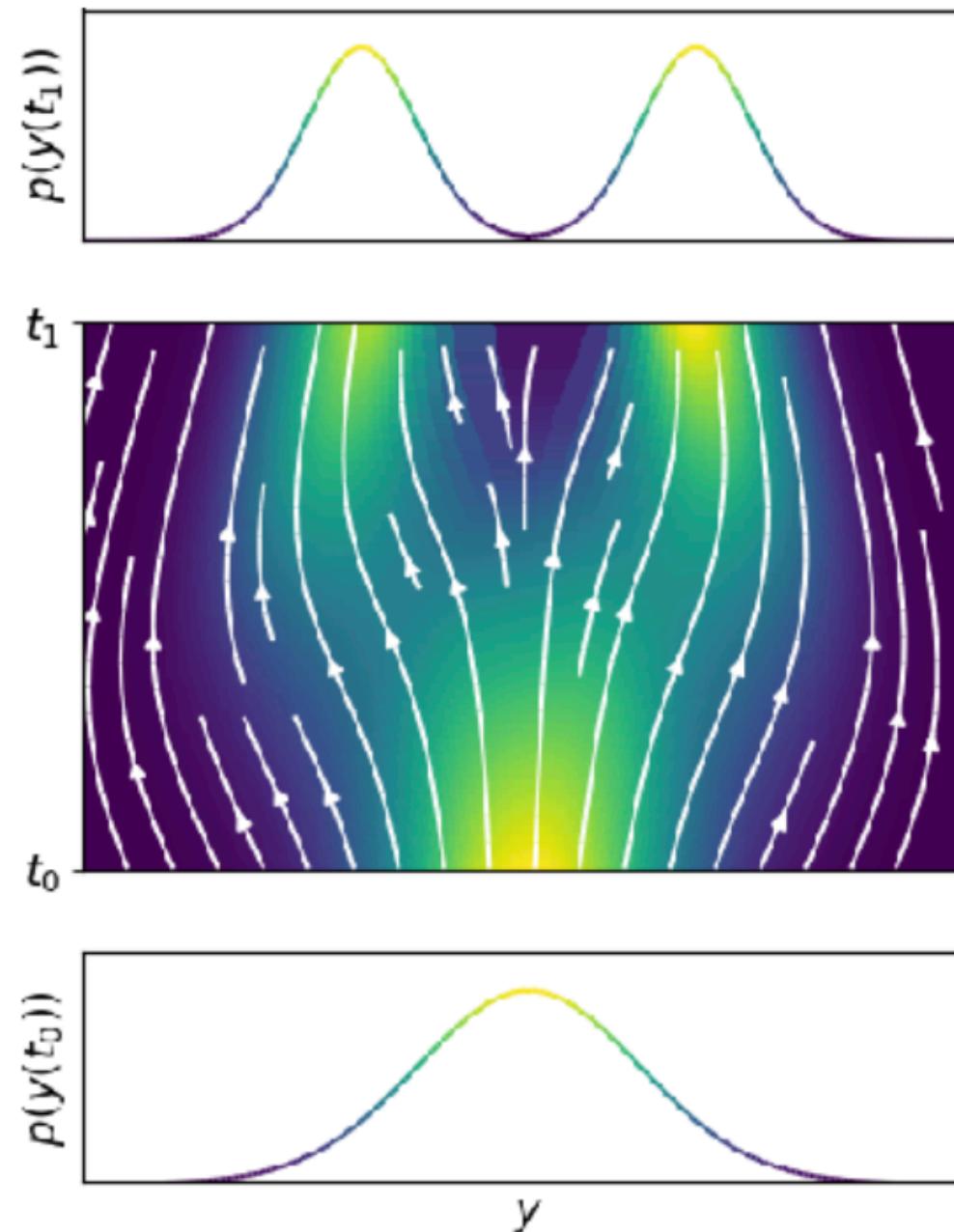
Roger Jiang
NYU



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 π^* where

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

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

Mean-

Task: give

- non-
- mix
- loca
- pro

The screenshot shows a Google Scholar search results page for the query "mean field variational inference". The search bar at the top contains the query. Below it, a search summary says "About 142'000 results (0.12 sec)". On the left, there are filters for "Articles", "Any time" (with options for "Since 2024", "Since 2023", "Since 2020", and "Custom range..."), "Sort by relevance", "Sort by date", "Any type" (with "Review articles" checked), and checkboxes for "include patents" (unchecked) and "include citations" (checked). A "Create alert" button is also present.

The main content area displays five search results, each with a title, author(s), publication details, a brief abstract, citation information, and a link to the full PDF:

- Variational inference: A review for statisticians**
DM Blei, A Kucukelbir, JD McAuliffe - *Journal of the American ...*, 2017 - Taylor & Francis
... inference about unknown quantities as a calculation involving the posterior density. In this article, we review variational inference (... ideas behind mean-field variational inference, discuss ...)
☆ Save ⚡ Cite Cited by 5062 Related articles All 22 versions [PDF] tandfonline.com
- Advances in variational inference**
C Zhang, J Bütepage, H Kjellström... - *IEEE transactions on ...*, 2018 - ieeexplore.ieee.org
... trends in variational inference. We first introduce standard mean field variational inference, ... VI, which includes variational models beyond the mean field approximation or with atypical ...
☆ Save ⚡ Cite Cited by 759 Related articles All 10 versions [PDF] ieee.org
- Theoretical and computational guarantees of mean field variational inference for community detection**
AY Zhang, HH Zhou - 2020 - projecteuclid.org
... the mean field method for community detection under the stochastic block model. For an iterative batch coordinate ascent variational inference ..., the mean field variational inference via ...
☆ Save ⚡ Cite Cited by 77 Related articles All 6 versions [PDF] projecteuclid.org
- A generalized mean field algorithm for variational inference in exponential families**
EP Xing, MI Jordan, S Russell - *arXiv preprint arXiv:1212.2512*, 2012 - arxiv.org
... mean field (GMF) algorithms for approximate inference in ... agation (GBP) or cluster variational meth ods. While those methods are ... Unlike the cluster variational methods, the approach is ...
☆ Save ⚡ Cite Cited by 292 Related articles All 4 versions [PDF] arxiv.org
- Mean field variational inference via Wasserstein gradient flow**
R Yao, Y Yang - *arXiv preprint arXiv:2207.08074*, 2022 - arxiv.org
... gradient flows and mean-field variational inference, and formulate the ... framework for mean-field inference via alternating ... statistical concentration of the meanfield approximation and the ...
☆ Save ⚡ Cite Cited by 12 Related articles All 2 versions [PDF] arxiv.org
- Statistical inference in mean-field variational Bayes**
W Han, Y Yang - *arXiv preprint arXiv:1911.01525*, 2019 - arxiv.org
... In this section, we begin with a brief review on the mean-field variational inference for a class of Bayesian latent variable models. Then we provide two perspectives for explaining the ...
☆ Save ⚡ Cite Cited by 15 Related articles All 2 versions [PDF] arxiv.org

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 $T^{0 \rightarrow 1} := \underset{T \in \mathcal{T}(p_0, p_1)}{\operatorname{argmin}} \| \operatorname{id} - T \|_{L^2(p_0)}^2$
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 $W_2^2(p_0, p_1) := \| \operatorname{id} - T^{0 \rightarrow 1} \|_{L^2(p_0)}^2$

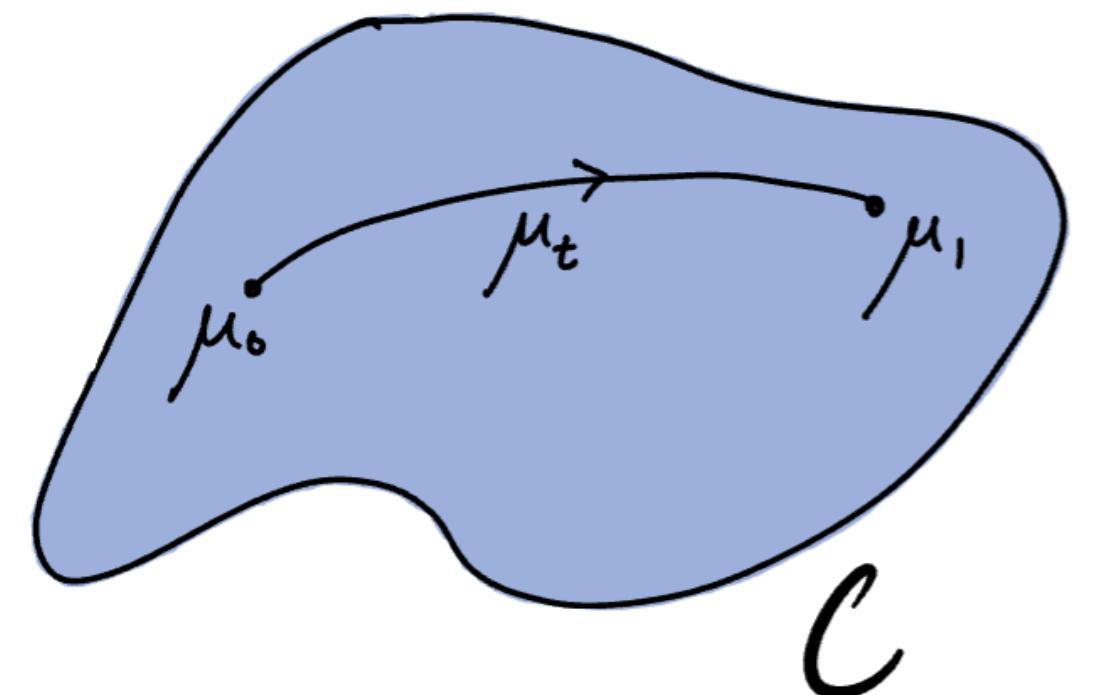
W_2 -geodesically convex sets

Paths in W_2

[McCann (1997)]

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

(examples include Gaussians and space of product measures)



Current algorithms for MF-VI

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

Implementation issues

- 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 $\pi^* = \arg \min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \text{KL}(\mu \| \pi)$  $\partial_t \mu_t = “ - \nabla_{\mathbb{W}} \text{KL}(\mu_t \| \pi) \big|_{\mathcal{P}(\mathbb{R})^{\otimes d}} ”$

$\nabla^2 V \succeq \alpha I \implies \text{KL}(\cdot \| \pi)$ 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 π^*

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

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

Optimal transport provides a canonical choice for the map:

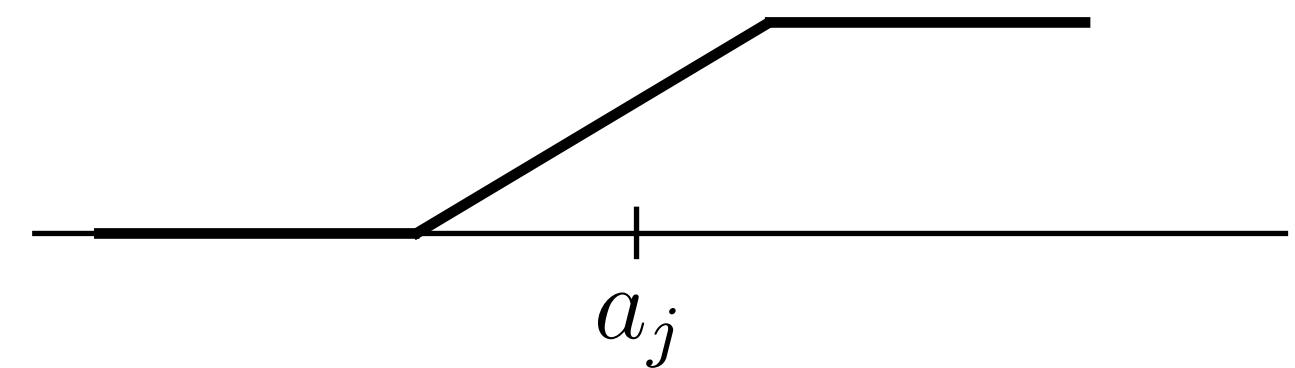
$$\begin{aligned} T^*(x) &= (T_1^*(x_1), \dots, T_d^*(x_d)) && \text{where } \varphi_i^* \text{ is some convex function} \\ &= ((\varphi_1^*)'(x_1), \dots, (\varphi_d^*)'(x_d)) && \text{i.e., } (\varphi_i^*)' \text{ is monotone} \end{aligned}$$

New goal: find T^* using only query access to V and ∇V

Mathematical approximation (in 1D)

How to fit T_1^* ?

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

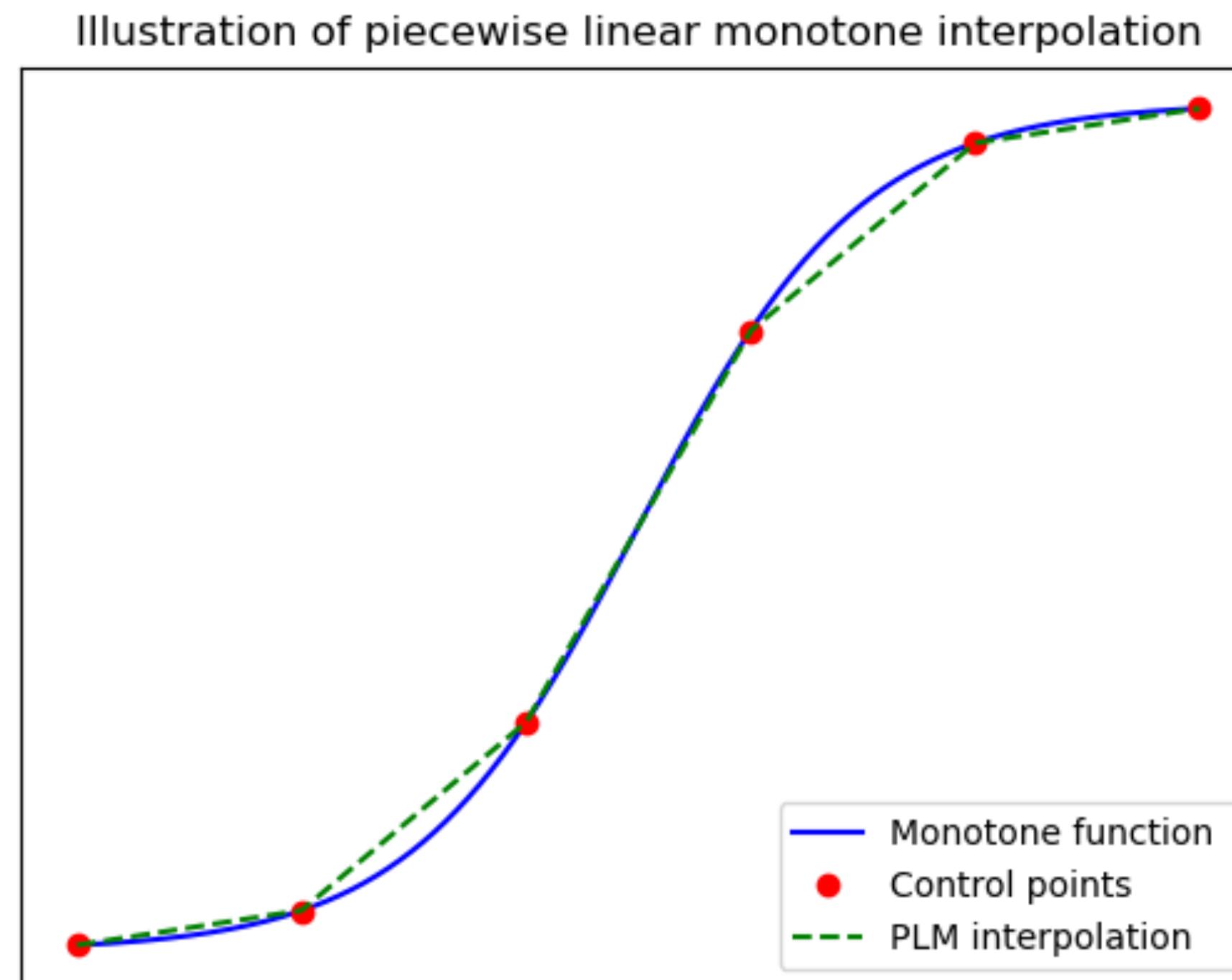


Mathematical approximation (in 1D)

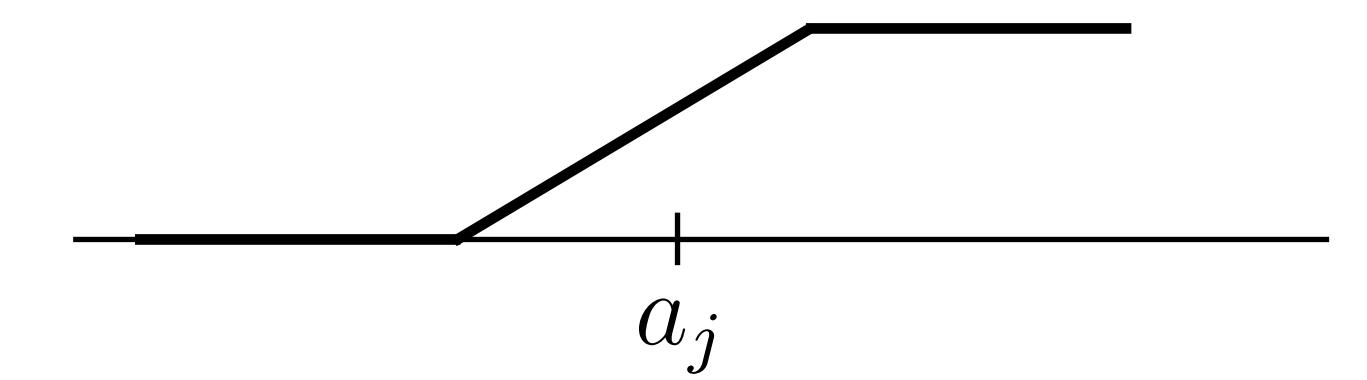
How to fit T_1^* ? With piecewise linear monotone functions

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

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$
and $\lambda \in \mathbb{R}_+^J$



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

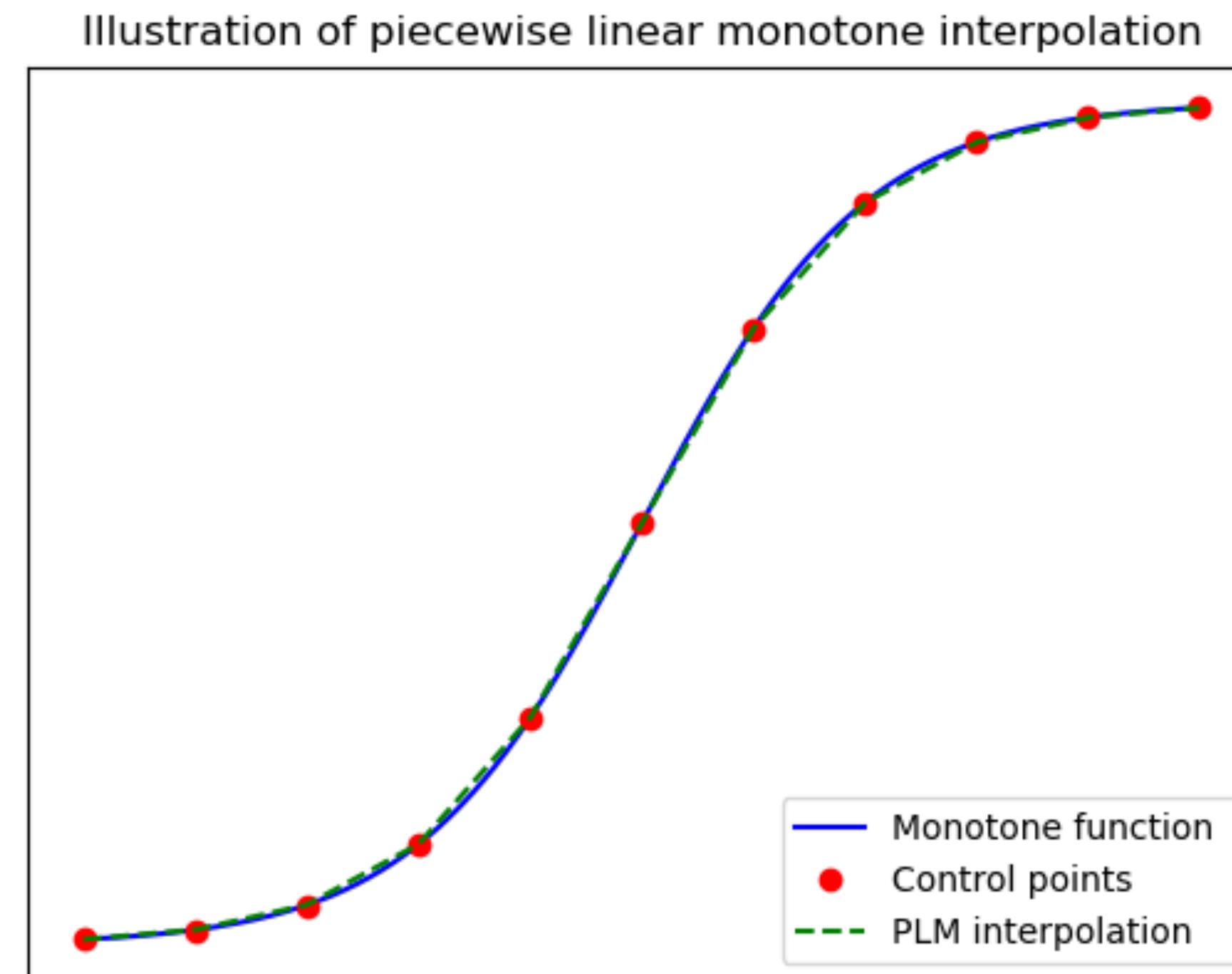


Mathematical approximation (in 1D)

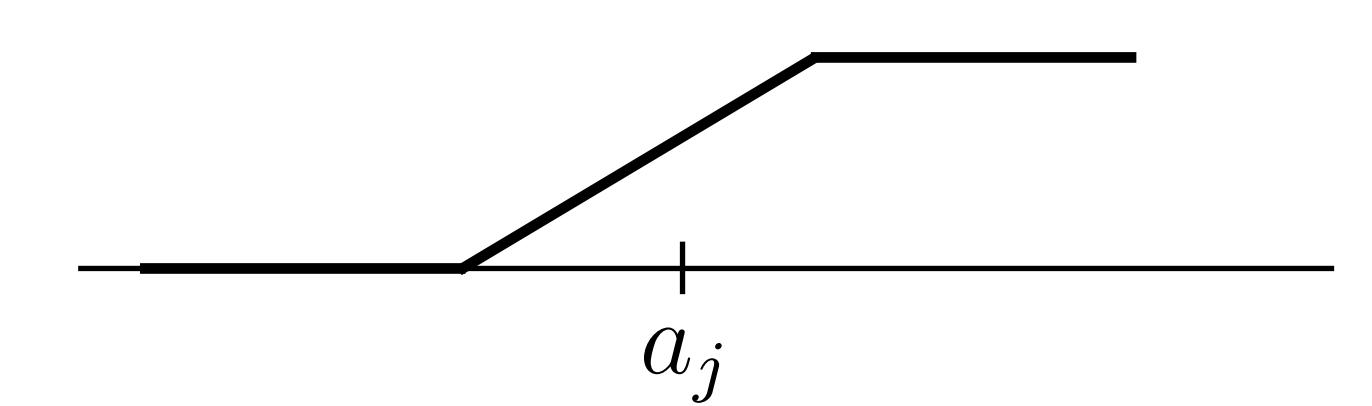
How to fit T_1^* ? With piecewise linear monotone functions

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

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.)

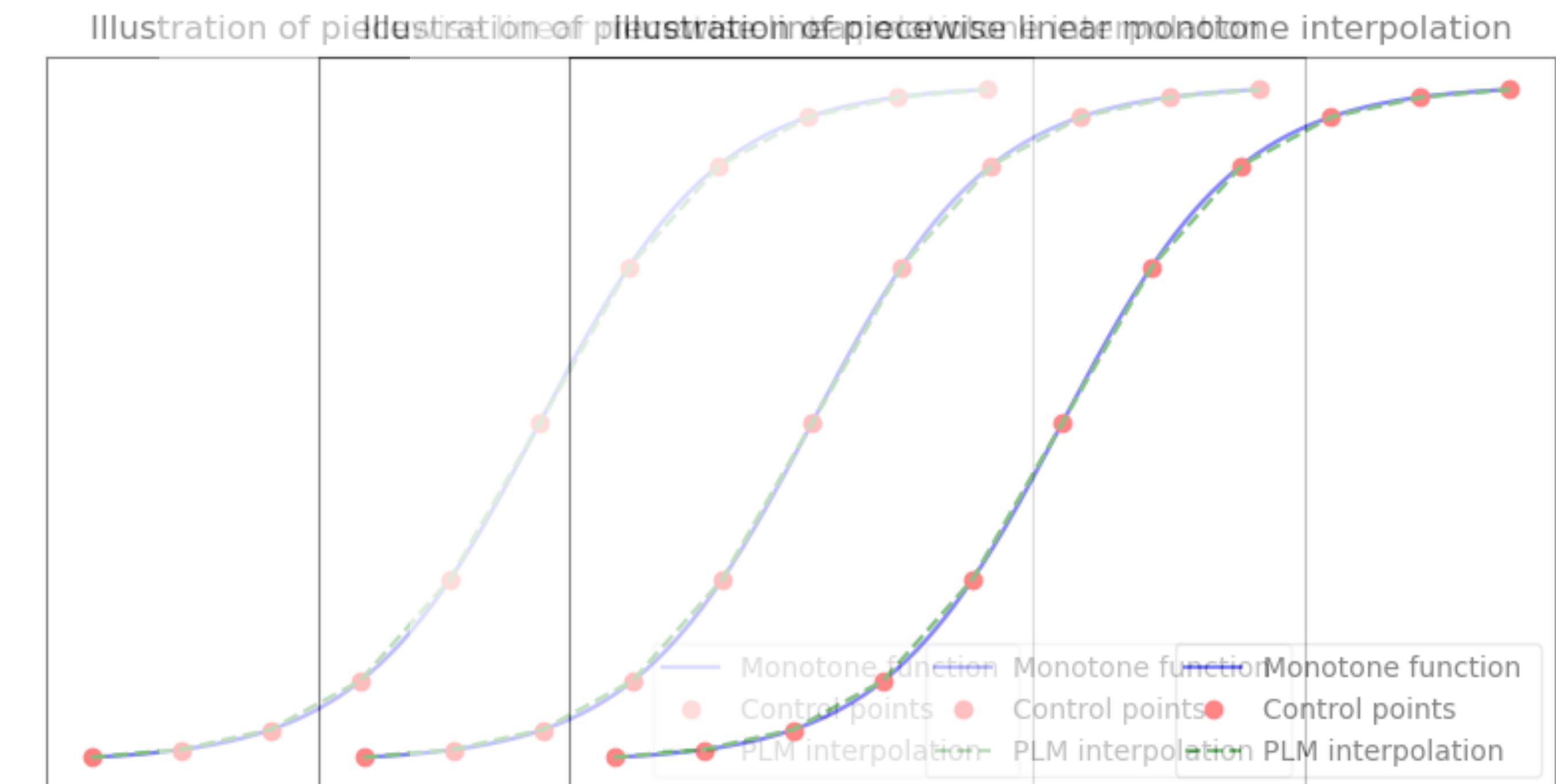
In higher dimensions, there is a natural extension:

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

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

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

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

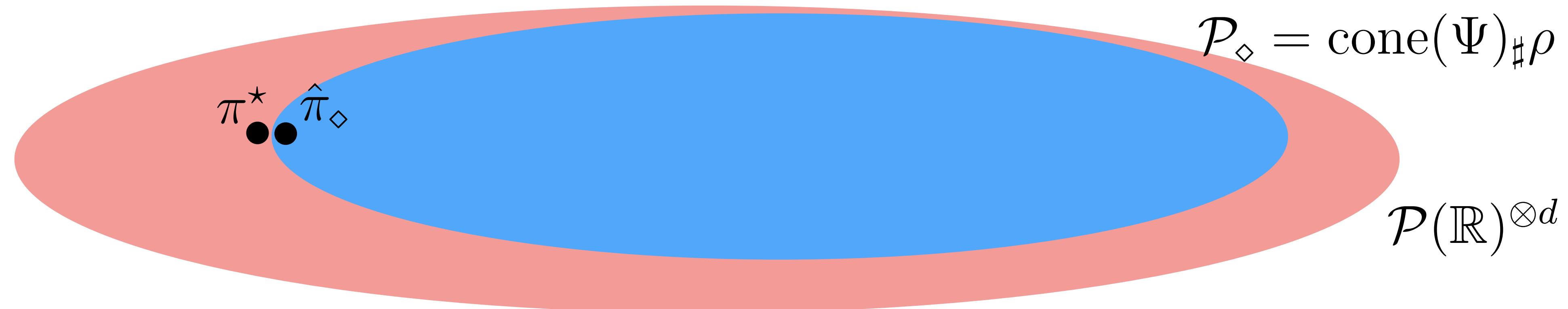


Unfortunately, approximation is not possible

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

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

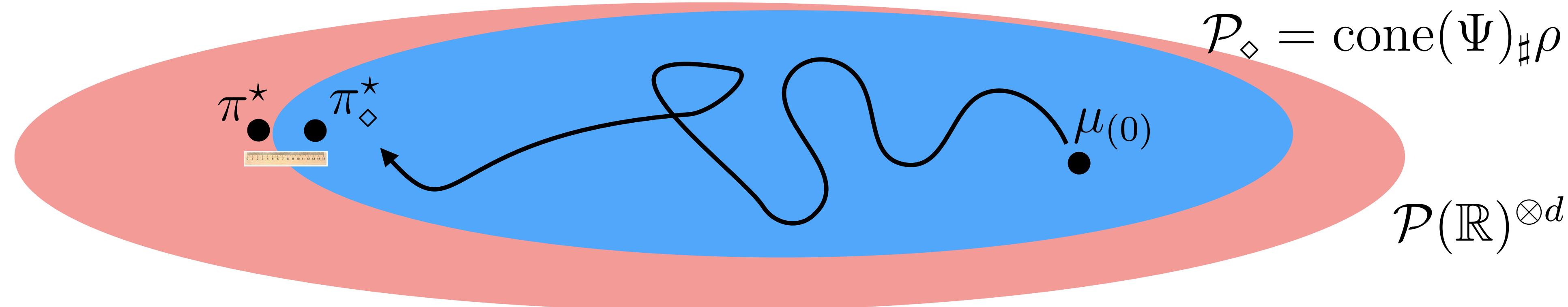
True for J large enough



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

Let's optimize directly over the parameterization

$$\pi^* = \arg \min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \text{KL}(\mu \| \pi) \stackrel{?}{\simeq} \pi_\diamond^* = \arg \min_{\mu \in \mathcal{P}_\diamond} \text{KL}(\mu \| \pi)$$



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

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 $J = \tilde{O}(\kappa^2 d^{1/2}/\varepsilon)$, $\sqrt{\alpha}W_2(\pi_\diamond^\star, \pi^\star) \leq \varepsilon$.*

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

Properties of pushforward cones

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

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

These properties hold for *Polyhedral sets*

Proposing to solve $\pi_{\diamond}^{\star} = \arg \min_{\mu \in \mathcal{P}_{\diamond}} \text{KL}(\mu \| \pi) \iff \lambda_{\diamond}^{\star} = \arg \min_{\lambda \in \mathcal{K}} \text{KL}(\mu_{\lambda} \| \pi)$

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

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

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

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

(convex subset)

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

How to optimize over pushforward cones

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

Gradient flows over polyhedral sets: “ $\nabla_{\mathbb{W}} \text{KL}(\mu_t \| \pi) \Big|_{\mathcal{P}_{\diamond}} = Q^{-1} \nabla_{\lambda} \text{KL}(\mu_{\lambda} \| \pi)$ ”

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

Need smoothness and strong convexity for convergence guarantees

Road to convergence guarantees

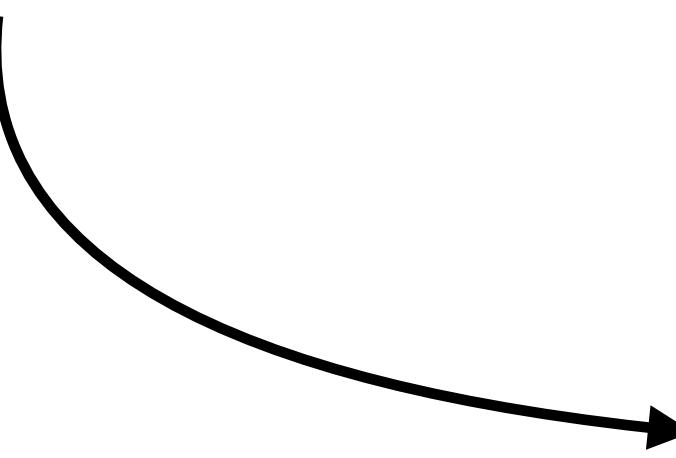
Strong convexity is free (\mathcal{P}_\diamond is geodesically convex, and $\nabla^2 V \succeq \alpha I$)

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

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

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

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

 Choose $\ell = 1/\sqrt{\beta}$

Accelerated gradient descent for VI

$\lambda \mapsto \text{KL}(\mu_\lambda \| \pi)$ 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 $\text{KL}(\cdot \| \pi)$

Set $\eta^{(0)} = \lambda^{(0)}$, $\kappa := \beta(1 + \Upsilon)/\alpha$

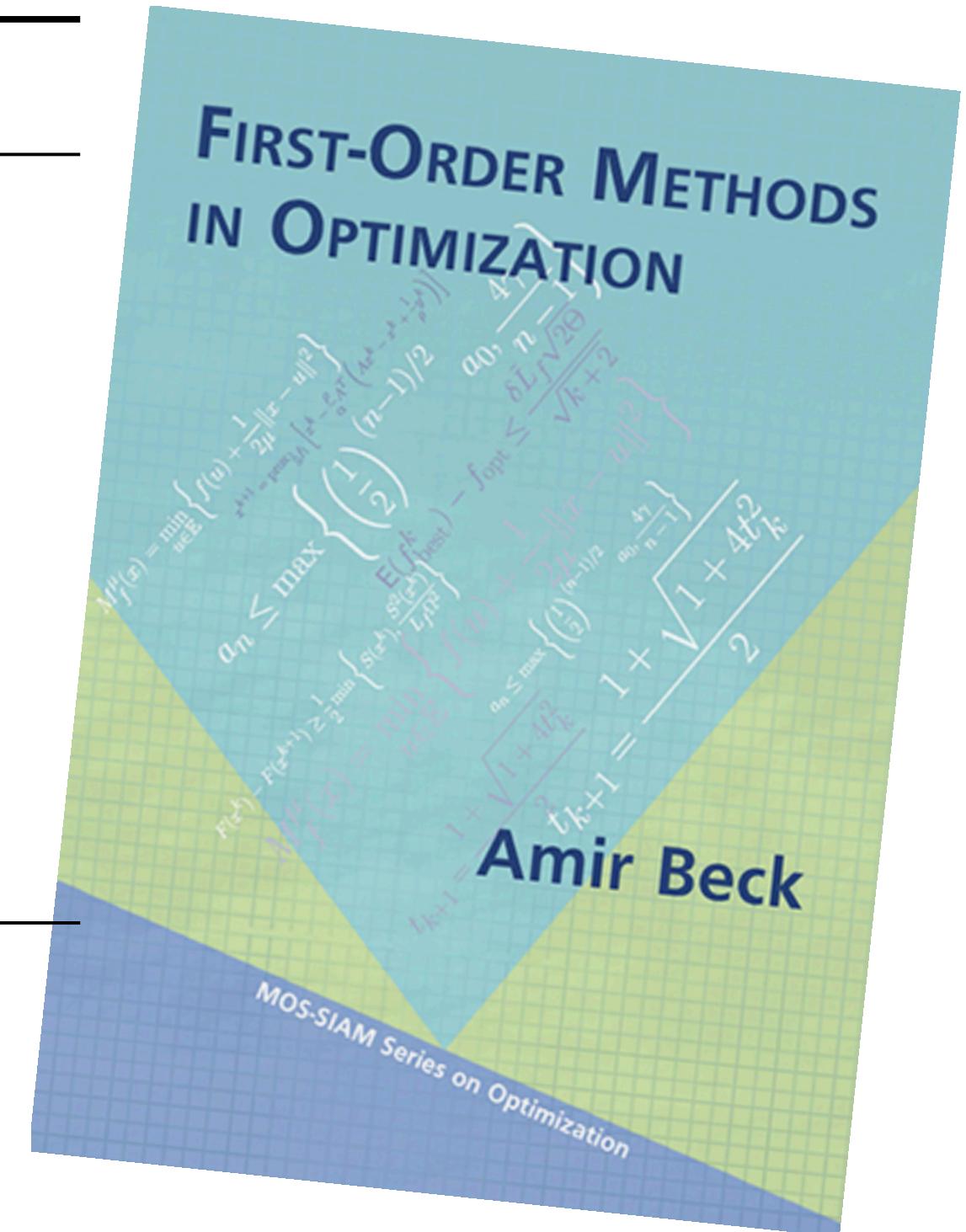
for $t = 0, 1, 2, 3, \dots$ **do**

$\lambda^{(t+1)} \leftarrow \text{proj}_{\mathbb{R}_+^{dJ}, Q}(\eta^{(t)} - \frac{1}{\beta(1+\Upsilon)} Q^{-1} \nabla_\lambda \text{KL}(\mu_{\eta^{(t)}} \| \pi))$

$\eta^{(t+1)} \leftarrow \lambda^{(t+1)} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} (\lambda^{(t+1)} - \lambda^{(t)})$

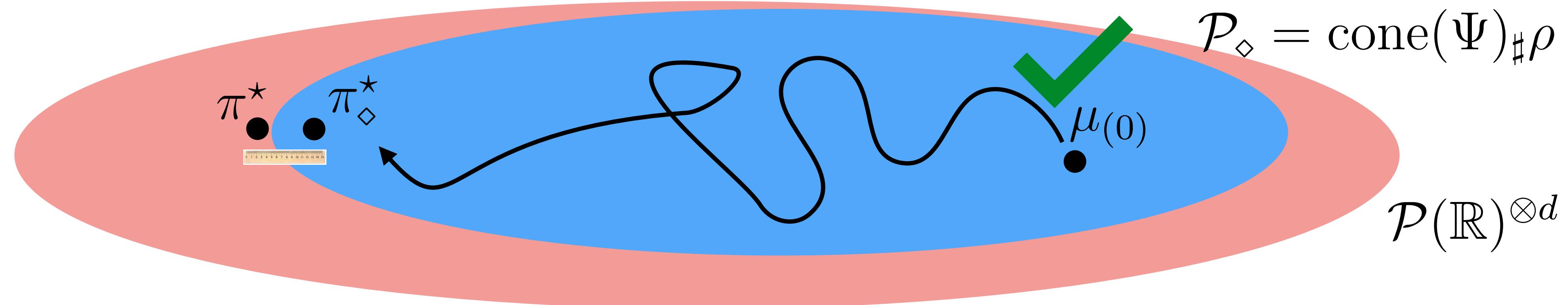
end for

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



But are these minimizers actually close?

$$\pi^* = \arg \min_{\mu \in \mathcal{P}(\mathbb{R})^{\otimes d}} \text{KL}(\mu \| \pi) \stackrel{?}{\simeq} \pi_\diamond^* = \arg \min_{\mu \in \mathcal{P}_\diamond} \text{KL}(\mu \| \pi)$$



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

24

Quick proof sketch of closeness:

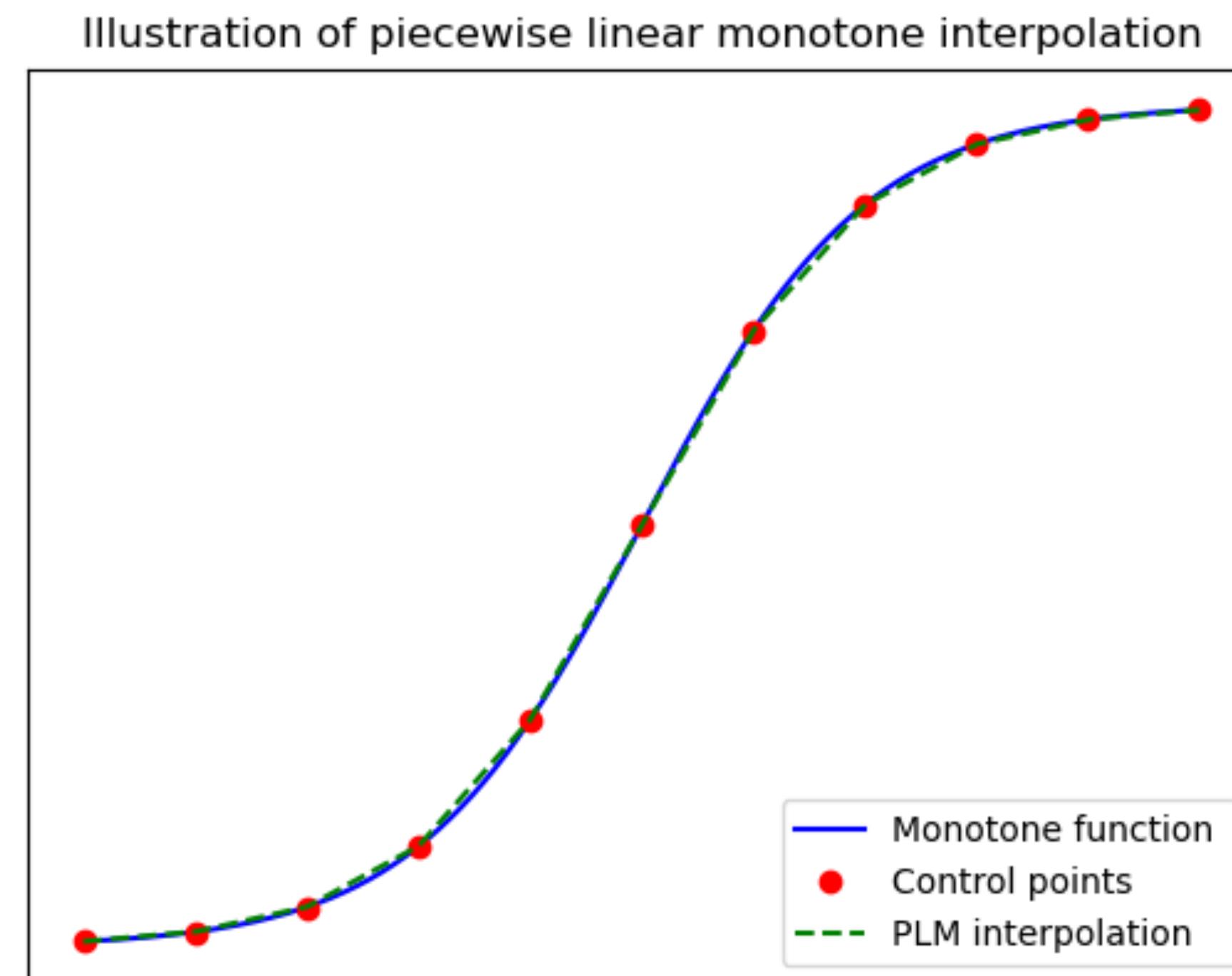
$$W_2^2(\pi_{\diamond}^{\star}, \pi^{\star}) = W_2^2((T_{\diamond}^{\star})_{\sharp}\rho, (T^{\star})_{\sharp}\rho) = \|T_{\diamond}^{\star} - T^{\star}\|_{L^2(\rho)}^2$$

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

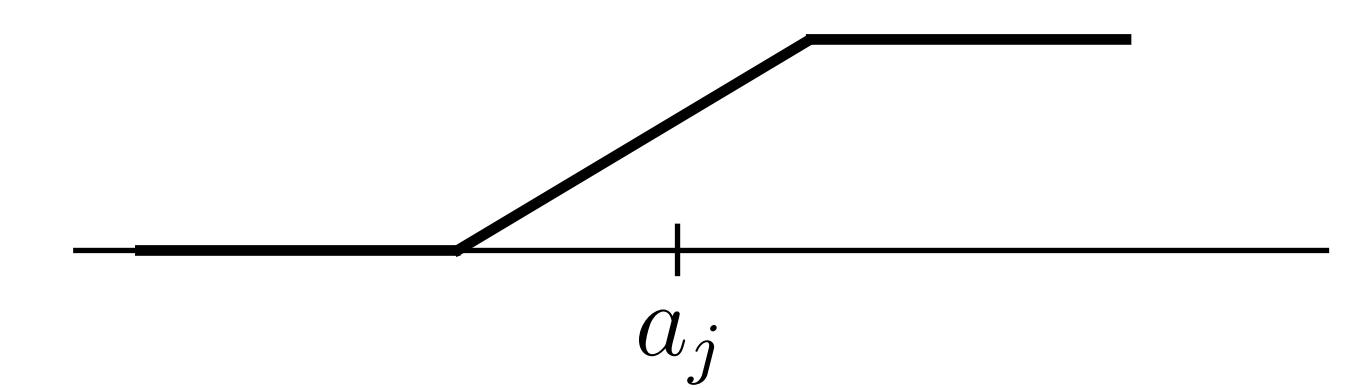
How close is this approximation?

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

where $\psi_j(x) \sim \min\{1, \max\{x, 0\}\}$
and $\lambda \in \mathbb{R}_+^J$



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

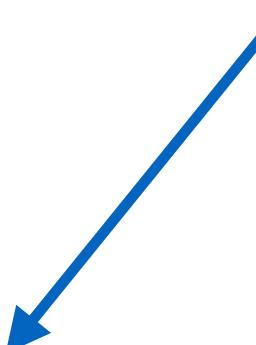


Quick proof sketch of closeness:

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

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

(approximation is close to optimal)



Proof requires new regularity properties of the Monge–Ampère equation

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

(Caffarelli (2000))

Improvement under smoothness?

Improvement under smoothness!

Theorem (Smooother 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 $\sqrt{\alpha}W_2(\pi_\diamond^\star, \pi^\star) \leq \varepsilon$.*

Implementation?

Implementation (yes, we coded it!)

```
Users / apooladian / mean-fieldvi_public / mfv_main.py / MFVI / SPGD
  1 import numpy as np
  2
  3 from optimization_utils import *
  4 from misc_utils import *
  5 from gaussian_utils import rho_gaussian_samples
  6
  7 class MFVI:
  8     def __init__(self, V, grad_V, mesh, trunc, dim):
  9         self.V = V
 10         self.grad_V = grad_V
 11         self.mesh = mesh
 12         self.trunc = trunc
 13         self.dim = dim
 14         self.J = int(2 * self.trunc / self.mesh)
 15
 16         self.lamb_opt, self.v_opt, self.T_opt = None, None, None
 17
 18         self.KL_vals, self.W2_vals = None, None
 19
 20         self.M_1d, _, self.Q, self.Qinv, means, self.gradient_num = build_M_FAST(dim=self.dim, mesh=self.mesh, truncation=self.trunc)
 21     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=False):
 22         self.alpha = alpha
 23         if save_vals:
 24             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,
 25                                         self.grad_V, self.V, num_iter=num_iters, stochastic_samples=batch_size,
 26                                         compute_KL=compute_KL, compute_W2=compute_W2, ground_cov=ground_cov, KL_tol=tol, stopping_cond=stopping_cond)
 27         else:
 28             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.gradient_num, lamb0,
 29                                         self.grad_V, self.V, num_iter=num_iters, stochastic_samples=batch_size,
 30                                         compute_KL=compute_KL, compute_W2=compute_W2, ground_cov=ground_cov, KL_tol=tol, stopping_cond=stopping_cond)
 31
```

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

Example: Bayesian Logistic Regression

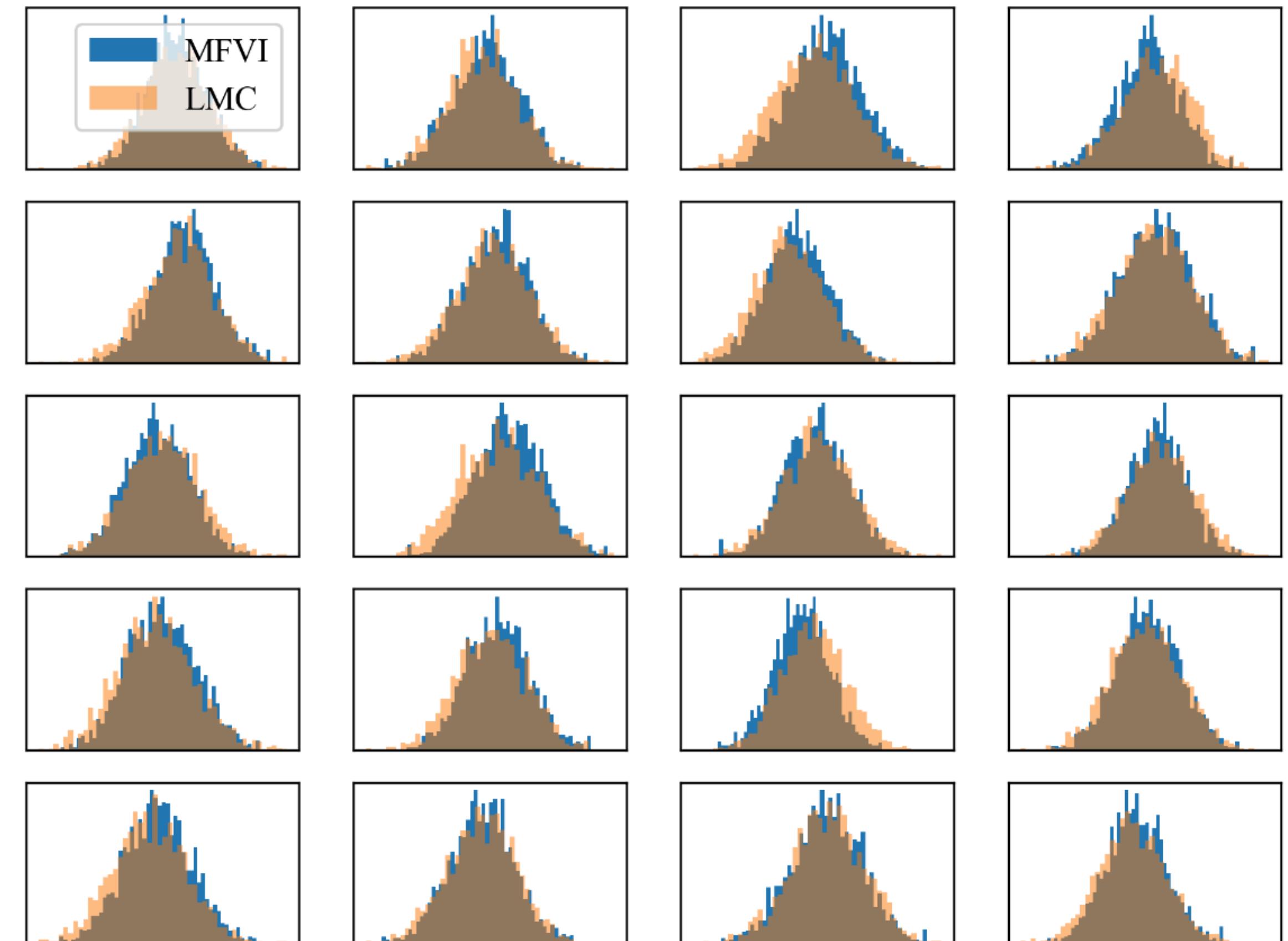
We generate (for random X_i and θ)

$$Y_i | X_i \sim \text{Bern}(\exp(\theta^\top X_i)) ,$$

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

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:

