Amortized optimization

Brandon Amos
Meta AI NYC, Fundamental AI Research (FAIR)

brandondamos bamos.github.io

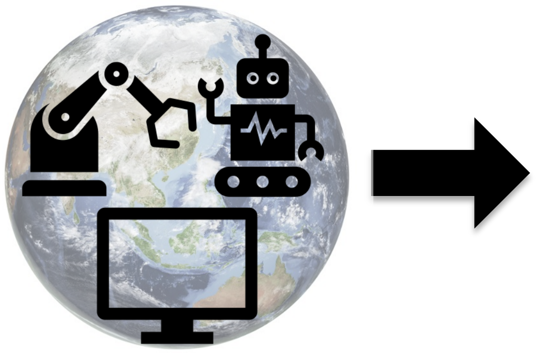
github.com/facebookresearch/amortized-optimization-tutorial
github.com/bamos/presentations

Optimization is crucial technology

Optimization is a **modeling** and **decision-making** paradigm and **encodes reasoning operations**

Finds the **best way to interact** with a **representation of the world**

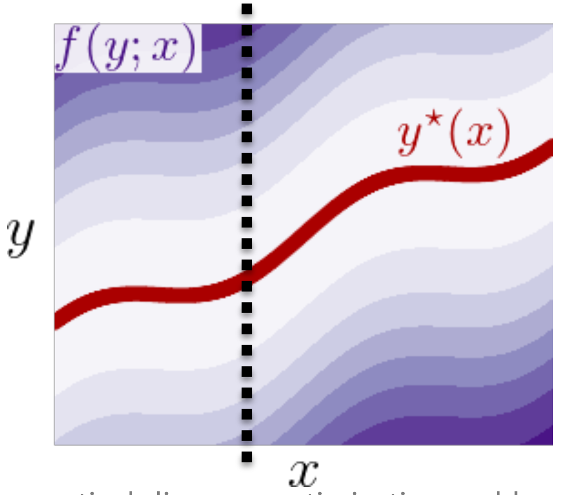
Focus: parametric optimization problems that are **repeatedly solved**



optimal solution   objective   context (or parameterization)

$$y^\star(x) \in \operatorname*{argmin}_{y\in\mathcal{C}(x)} f(y;x)$$

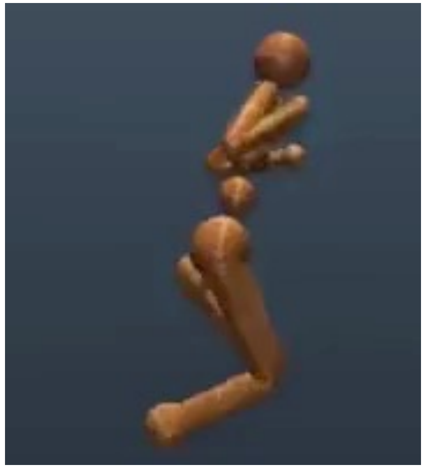
optimization variable   constraints

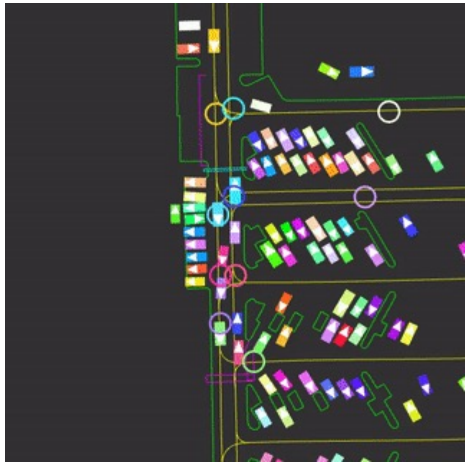


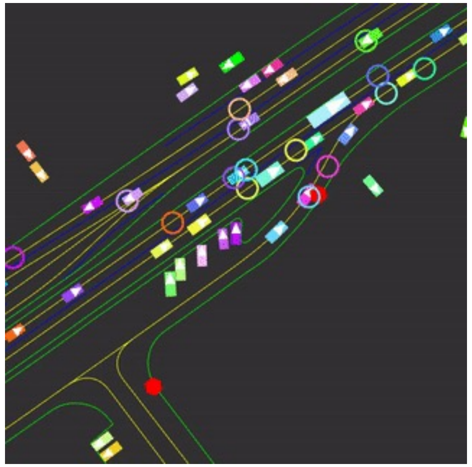
vertical slices are optimization problems

Breakthroughs enabled by optimization include

1. **controlling systems** (robotic, autonomous, mechanical, and multi-agent)



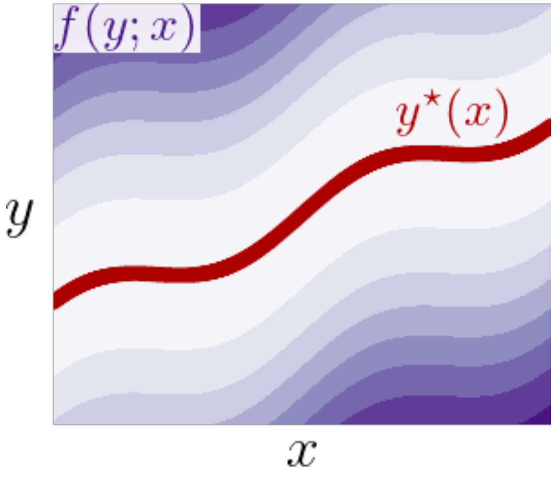




optimal solution objective context (or parameterization)

$$y^\star(x) \in \operatorname*{argmin}_{y \in \mathcal{C}(x)} f(y; x)$$

optimization variable constraints



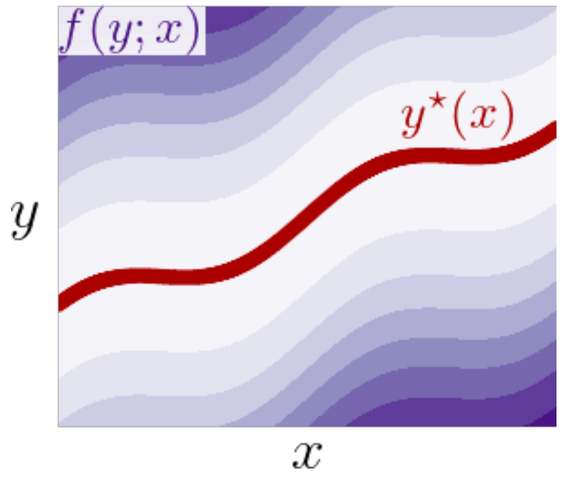
Breakthroughs enabled by optimization include

1. **controlling systems** (robotic, autonomous, mechanical, and multi-agent)
2. **making operational decisions** based on future predictions
3. efficiently **transporting** or **matching** resources, information, and measures
4. **allocating** budgets and portfolios
5. **designing** materials, molecules, and other structures
6. **solving inverse problems** (to infer underlying hidden costs, incentives, geometries, terrains)
7. **parameter learning** of predictive and statistical models

optimal solution　　objective　　context (or parameterization)

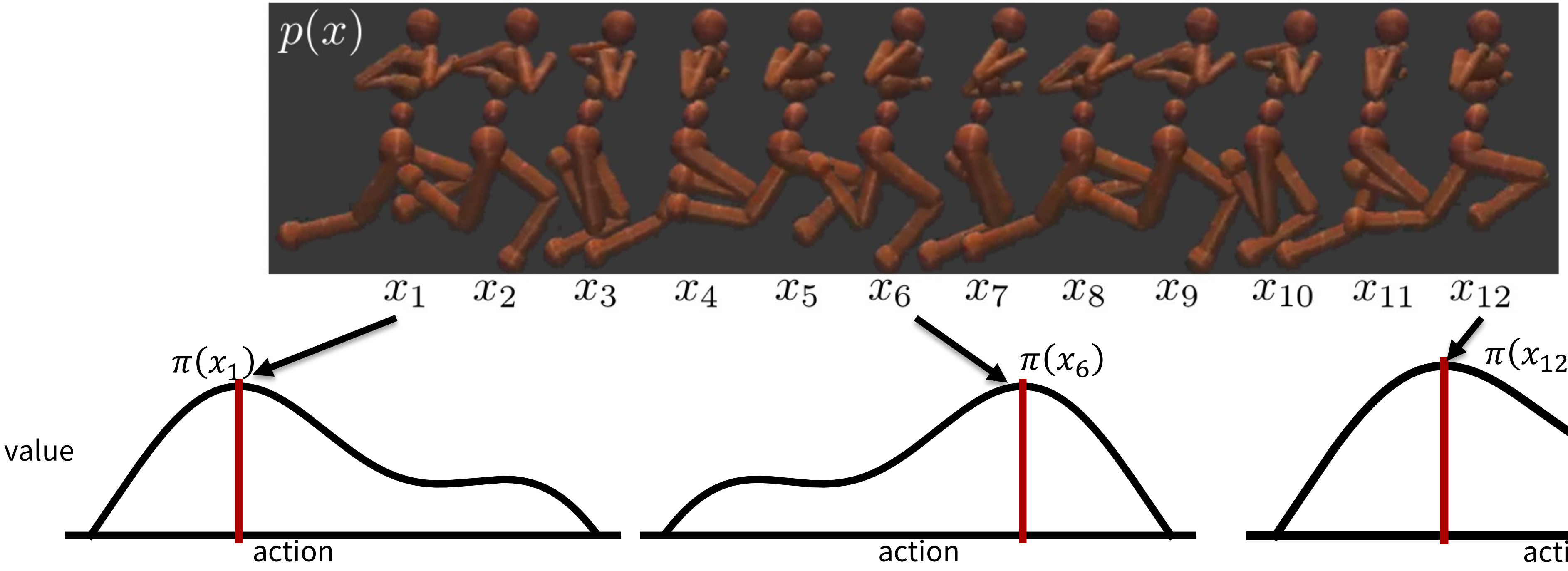
$$y^\star(x) \in \operatorname*{argmin}_{y\in\mathcal{C}(x)} f(y;x)$$

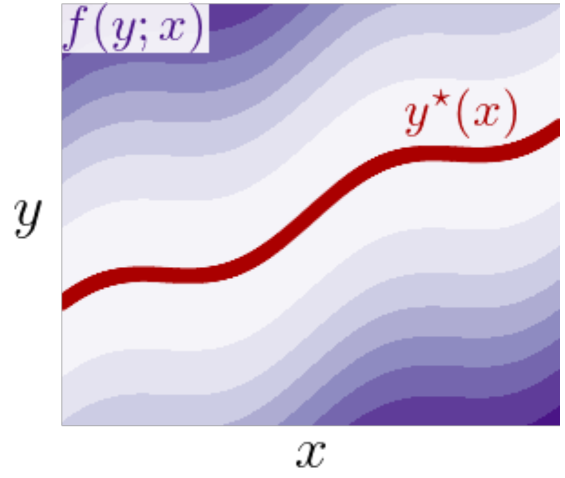
optimization variable　　constraints



Repeatedly solving optimization problems

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.
On the model-based stochastic value gradient for continuous reinforcement learning. Amos et al., L4DC 2021.





$$\pi(x) = \operatorname*{argmax}_u Q(x,u)$$

This talk: amortized optimization

Design decisions

Modeling paradigms for $\hat{y}_\theta$ (fully-amortized and semi-amortized models)
Learning paradigms for $\mathcal{L}$ (objective-based and regression-based)

Applications

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)
Variational inference (VAEs, semi-amortized VAEs)
Meta-learning (HyperNets, MAML)
Sparse coding (PSD, LISTA)
Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, NeuralSCS)
Optimal transport (slicing, conjugation, Meta Optimal Transport)

Amortization: approximate the solution map

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

A **fast amortization model** $\hat{y}_\theta$ can be **25,000 times faster** than solving $y^\star$ from scratch for VAEs

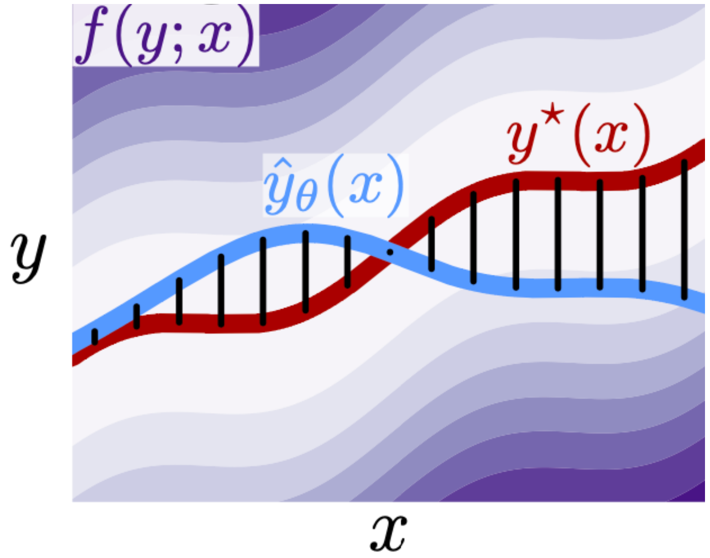
Amortization model $\hat{y}_\theta(x)$ tries to approximate $y^\star(x)$

Example: A neural network mapping from x to the solution

Loss $\mathcal{L}$ measures how well $\hat{y}$ fits $y^\star$ and optimized with $\min_\theta \mathcal{L}(\hat{y}_\theta)$

Regression: $\mathcal{L}(\hat{y}_\theta) \coloneqq \mathbb{E}_{p(x)} \|\hat{y}_\theta(x) - y^\star(x)\|_2^2$

Objective: $\mathcal{L}(\hat{y}_\theta) \coloneqq \mathbb{E}_{p(x)} f(\hat{y}_\theta(x))$



Modeling paradigms for $\hat{y}_\theta$

How to best-predict the solution?

Fully-amortized models: Map from the context x to the solution **without** accessing the objective f
Example: Neural network mapping from x to the solution
Most of our applications will focus on these

Semi-amortized models: Internally access the objective f
Example: Gradient-based meta-learning models such as MAML

$$\hat{y}_\theta^0 \rightarrow \hat{y}_\theta^1 \rightarrow \cdots \rightarrow \hat{y}_\theta^K =: \hat{y}_\theta(x)$$

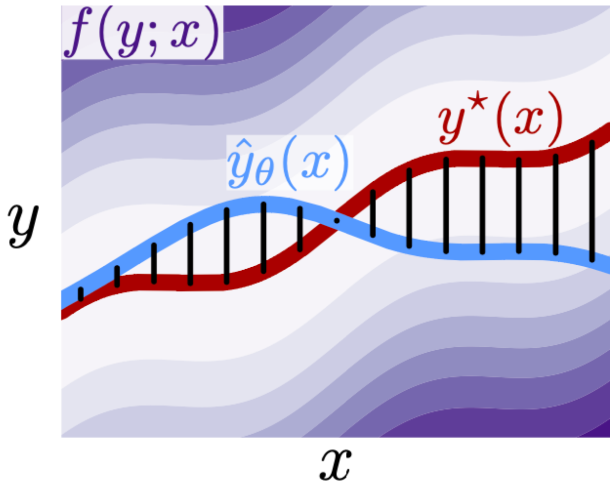
Learning paradigms for $\mathcal{L}$

What should the model $\hat{y}_\theta$ optimize for?

Regression-based

$\mathcal{L}_{\text{reg}}(\hat{y}_\theta) \coloneqq \mathbb{E}_{p(x)} \|\hat{y}_\theta(x) - y^\star(x)\|_2^2$

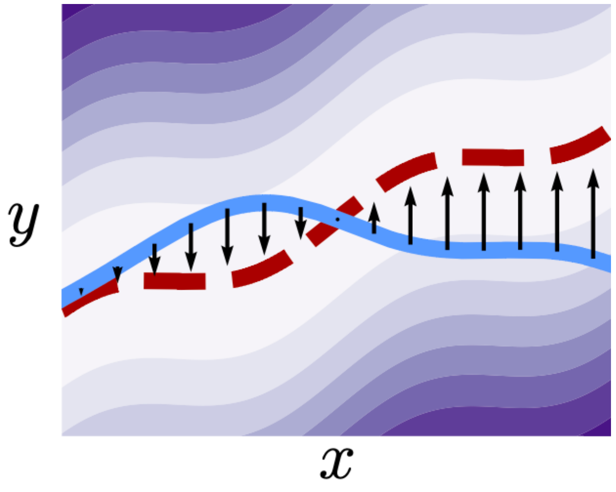
- $-$ Does not consider $f(y;x)$
- $+$ Uses global information with $y^\star(x)$
- $-$ Expensive to compute $y^\star(x)$
- $+$ Does not compute $\nabla_y f(y;x)$
- $-$ Hard to learn non-unique $y^\star(x)$



Objective-based:

$\mathcal{L}_{\text{obj}}(\hat{y}_\theta) \coloneqq \mathbb{E}_{p(x)} f(\hat{y}_\theta(x); x)$

- $+$ Uses objective information of $f(y;x)$
- $-$ Can get stuck in local optima of $f(y;x)$
- $+$ Faster, does not require $y^\star(x)$
- $-$ Often requires computing $\nabla_y f(y;x)$
- $+$ Easily learns non-unique $y^\star(x)$



This talk: amortized optimization

Design decisions

Modeling paradigms for $\hat{y}_\theta$ (fully-amortized and semi-amortized models)
Learning paradigms for $\mathcal{L}$ (objective-based and regression-based)

Applications

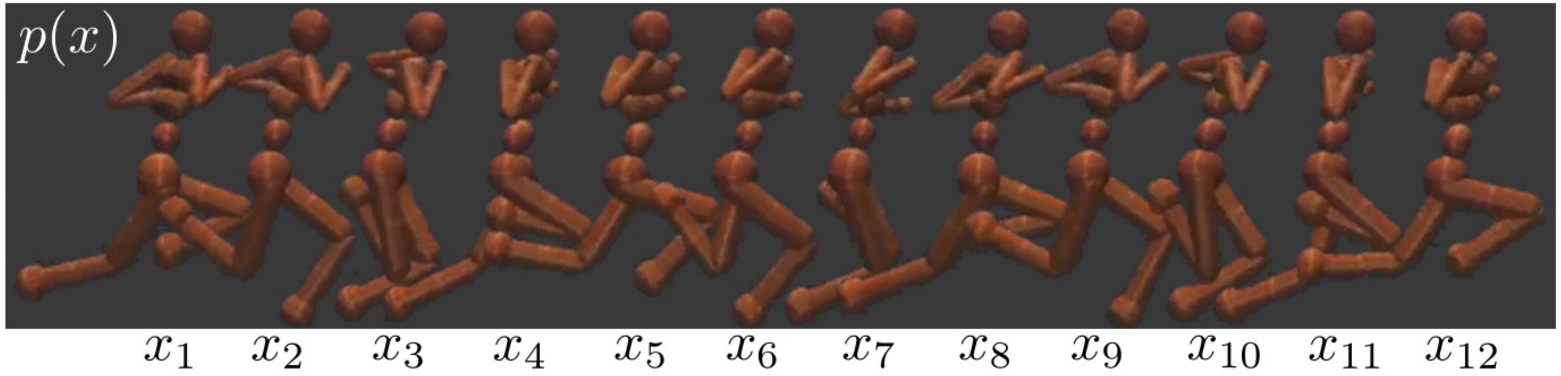
Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)
Variational inference (VAEs, semi-amortized VAEs)
Meta-learning (HyperNets, MAML)
Sparse coding (PSD, LISTA)
Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, NeuralSCS)
Optimal transport (slicing, conjugation, Meta Optimal Transport)

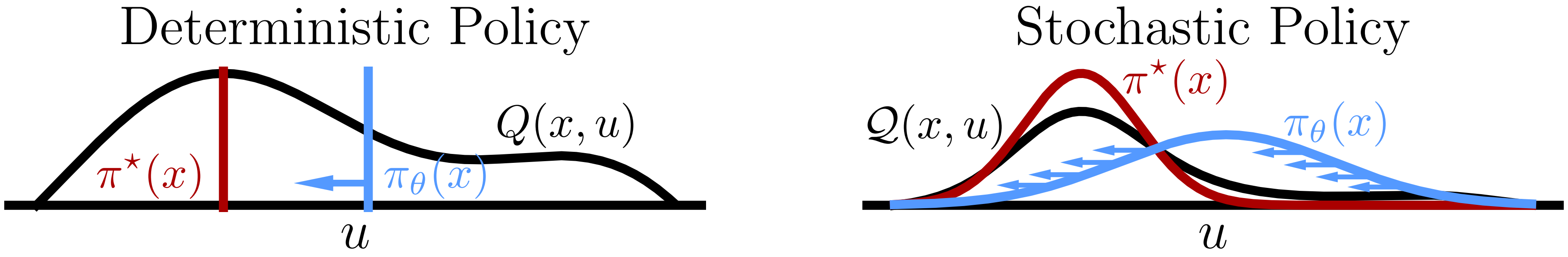
Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

$$\operatorname*{argmax}_{\theta} \mathbb{E}_{p(x)}\, Q(x, \pi_\theta(x))$$

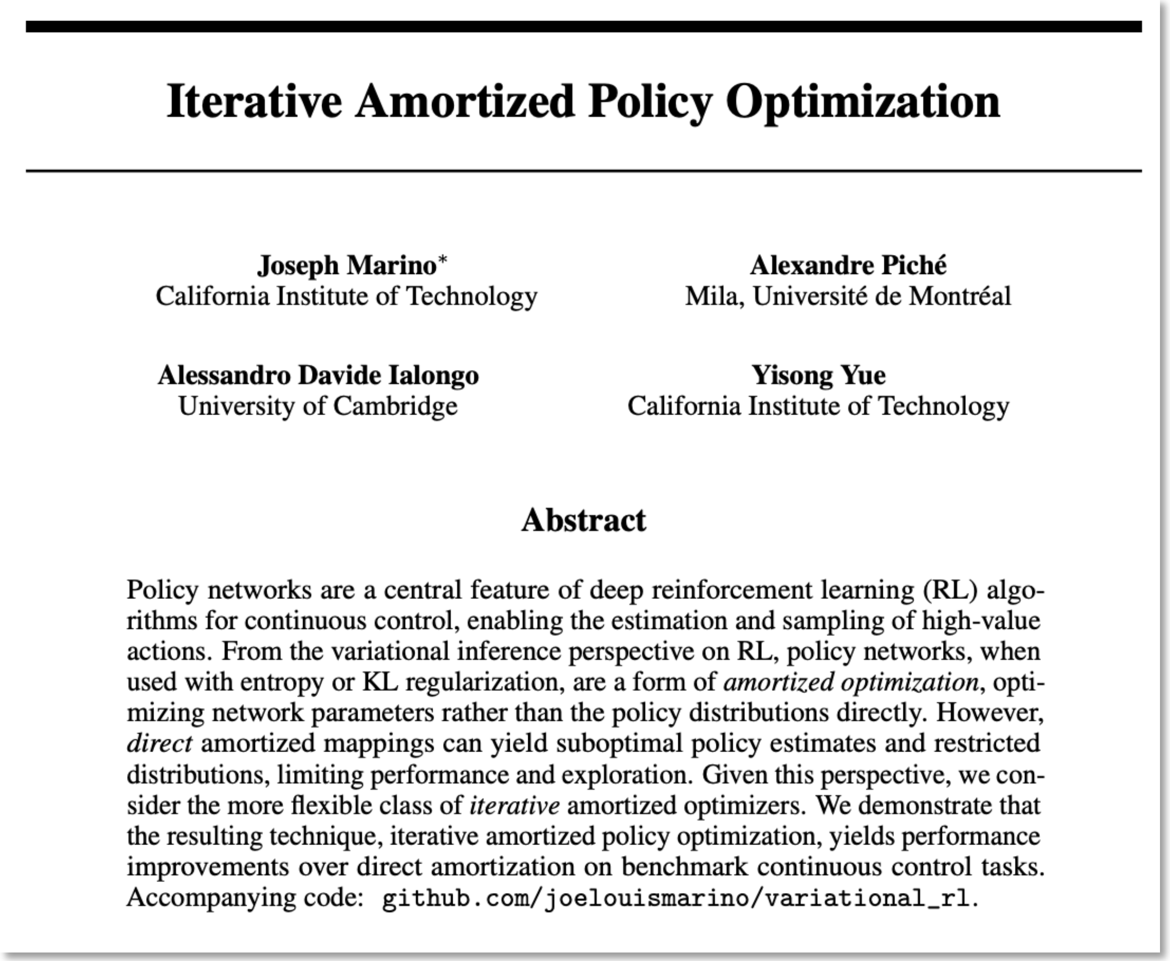




Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)



Iterative Amortized Policy Optimization

Joseph Marino*
California Institute of Technology

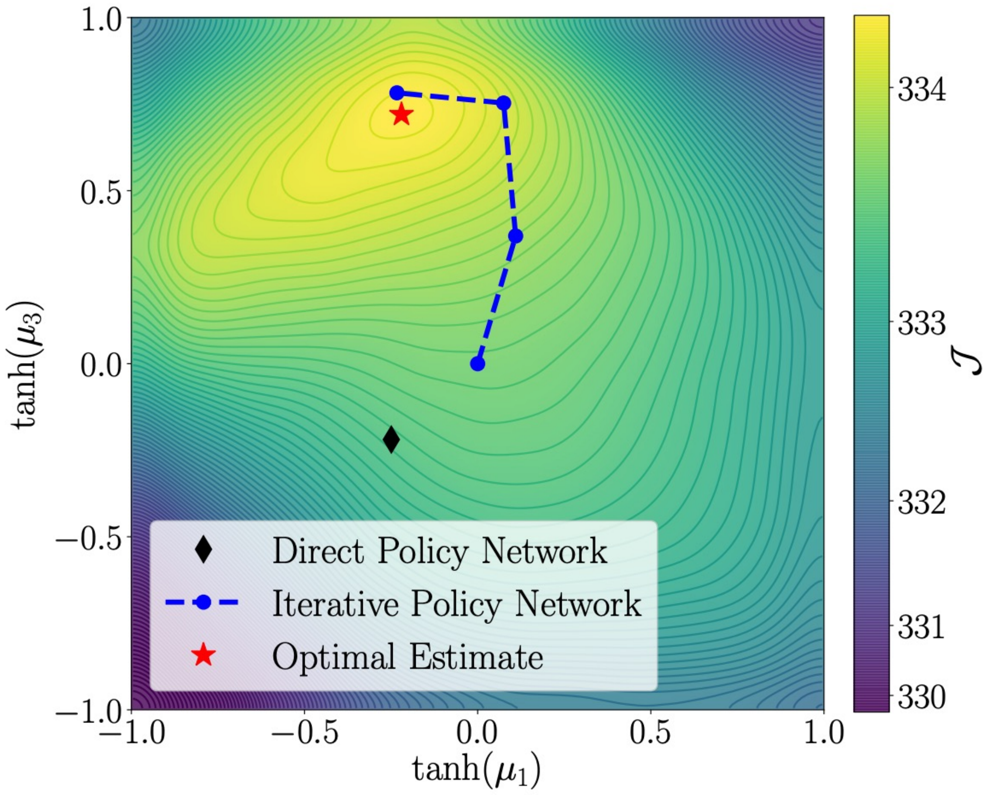
Alexandre Piché
Mila, Université de Montréal

Alessandro Davide Ialongo
University of Cambridge

Yisong Yue
California Institute of Technology

Abstract

Policy networks are a central feature of deep reinforcement learning (RL) algorithms for continuous control, enabling the estimation and sampling of high-value actions. From the variational inference perspective on RL, policy networks, when used with entropy or KL regularization, are a form of *amortized optimization*, optimizing network parameters rather than the policy distributions directly. However, *direct* amortized mappings can yield suboptimal policy estimates and restricted distributions, limiting performance and exploration. Given this perspective, we consider the more flexible class of *iterative* amortized optimizers. We demonstrate that the resulting technique, iterative amortized policy optimization, yields performance improvements over direct amortization on benchmark continuous control tasks. Accompanying code: `github.com/joelouismarino/variational_rl`.



Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Scalable Online Planning via Reinforcement Learning Fine-Tuning

Arnaud Fickinger* Facebook AI Research arnaudfickinger@fb.com

Hengyuan Hu* Facebook AI Research hengyuan@fb.com

Brandon Amos Facebook AI Research bda@fb.com

Stuart Russell UC Berkeley russell@berkeley.edu

Noam Brown Facebook AI Research noambrown@fb.com

Hanabi scores

Variant	Blueprint	SPARTA (Single)	SPARTA (Multi)	RL Search (Single)	RL Search (Multi)
Normal	24.23 ± 0.04 63.20%	24.57 ± 0.03 73.90%	24.61 ± 0.02 75.46%	24.59 ± 0.02 75.05%	**24.62 ± 0.03** **75.93%**
2 Hints	22.99 ± 0.04 17.50%	23.60 ± 0.03 25.85%	23.67 ± 0.03 26.87%	23.61 ± 0.03 27.85%	**23.76 ± 0.04** **31.01%**

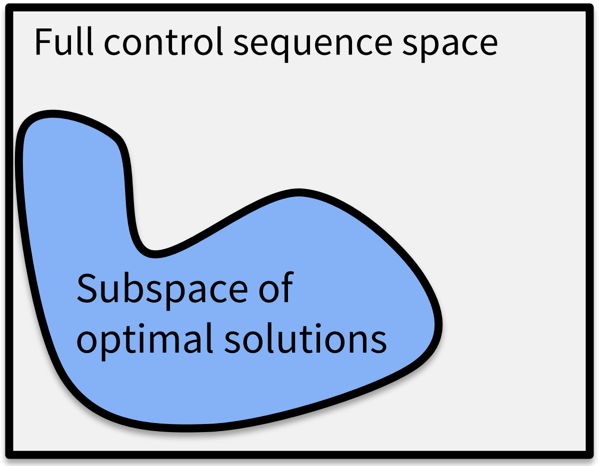
Ms. Pacman scores

Additional Samples	0	3.10^5	4.10^5	8.10^5
RL Fine-Tuning	1880	**3940**	**4580**	**5510**
PPO Training	1880	1900	1900	1920

Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

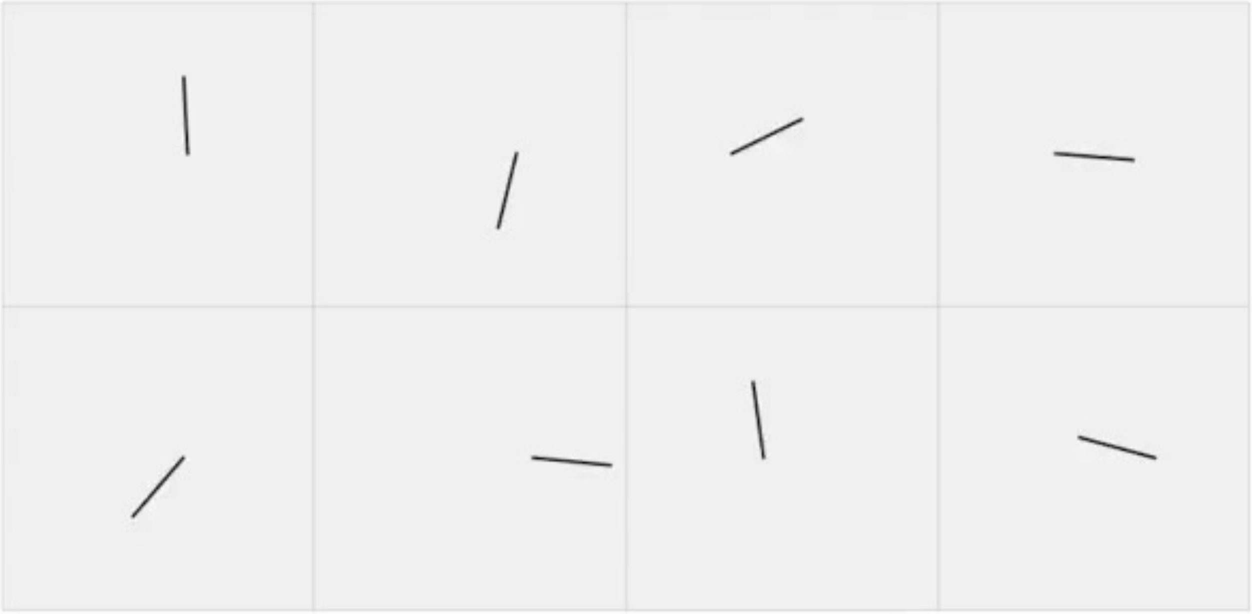
Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)



Amortize by **learning a latent subspace** of optimal solutions
Only search over optimal solutions rather than the entire space

The differentiable cross-entropy method. Amos and Yarats, ICML 2020.

Cartpole videos



Optimal controls over time — force on the cartpole

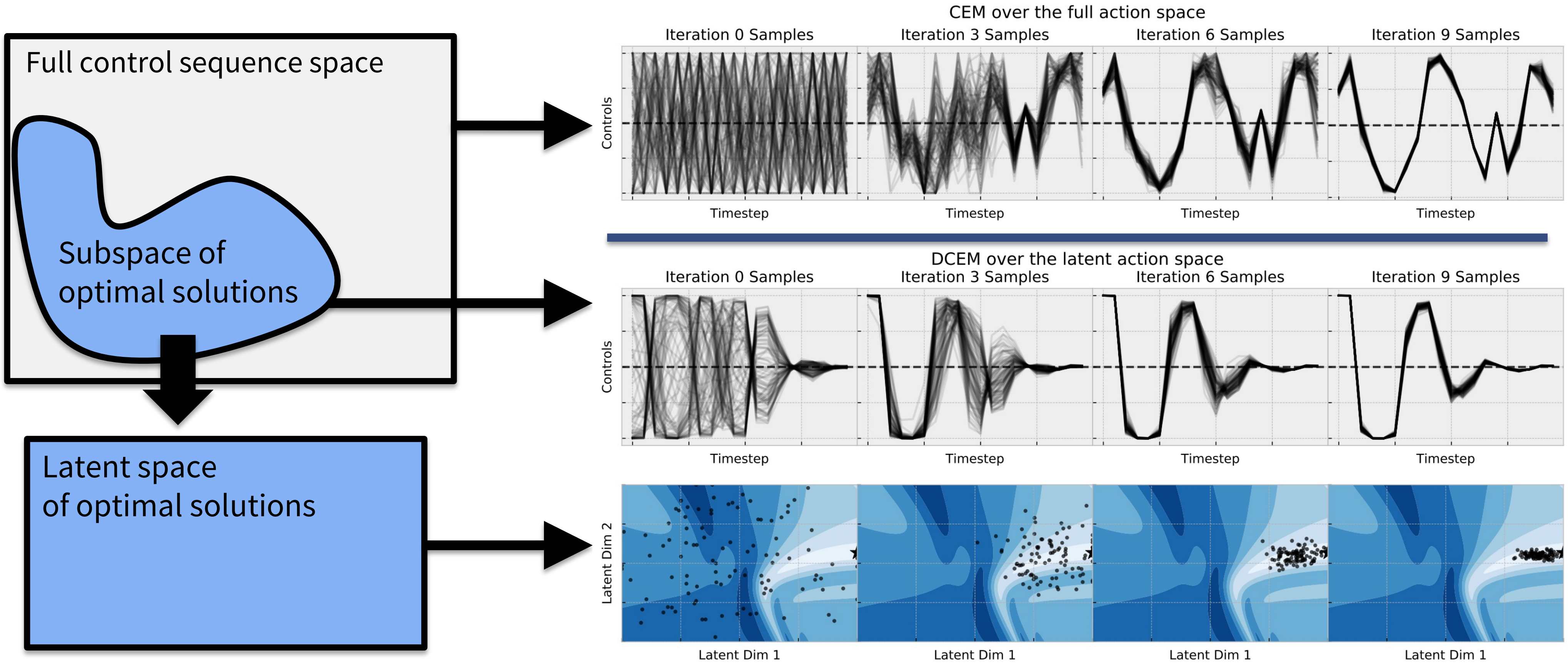
Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Amortize by **learning a latent subspace** of optimal solutions

The differentiable cross-entropy method. Amos and Yarats, ICML 2020.



Applications of amortized optimization

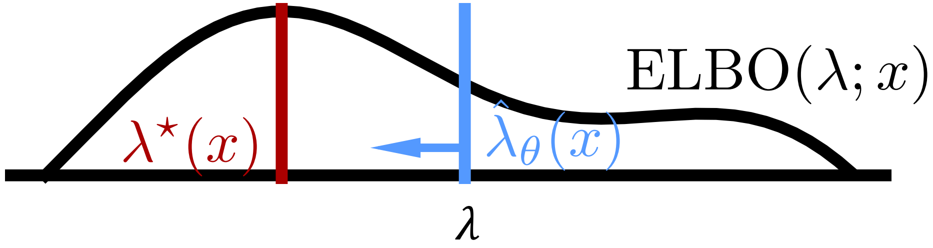
Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Variational inference (VAEs, semi-amortized VAEs)

Given a **VAE** model $p(x) = \log \int_z p(x|z)p(x)$, **encoding** amortizes the optimization problem

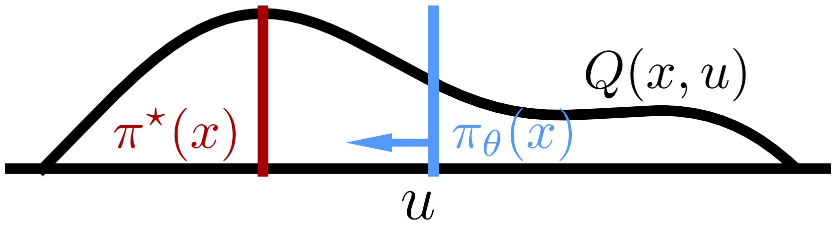
$$\lambda^\star(x) = \operatorname*{argmax}_\lambda \mathrm{ELBO}(\lambda; x) \quad \text{where} \quad \mathrm{ELBO}(\lambda; x) \coloneqq \mathbb{E}_{q(z;\lambda)}[\log p(x|z)] - \mathrm{D}_{\mathrm{KL}}(q(x;\lambda)|p(z)).$$



VAE amortization is conceptually the same as RL

Value gradient amortization in RL

$$\underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{p(x)}\ Q(x, \pi_\theta(x))$$

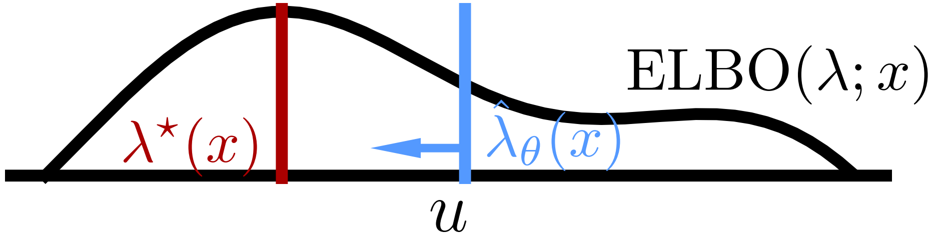


x: states from system

x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}

VAE posterior amortization

$$\underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{p(x)}\ \mathrm{ELBO}(\hat\lambda_\theta(x); x)$$



x: images from dataset

x_1 x_2 x_3

Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

Given a **task** $\mathcal{T}$, amortize the **computation of the optimal parameters** of a model

$$\theta^\star(\mathcal{T}) = \operatorname*{argmax}_\theta \ell_\mathcal{T}(\theta)$$

Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

Sparse coding (PSD, LISTA)

Given a **dictionary** W_d of **basis vectors** and **input** x, a **sparse code** is recovered with

$$y^\star(x) \in \underset{y}{\operatorname{argmin}} \|x - W_d y\|_2^2 + \alpha \|y\|_1$$

Predictive sparse decomposition (PSD) and Learned ISTA (LISTA) **amortize this problem**

Kavukcuoglu, Ranzato, and LeCun, 2010. Gregor and LeCun, 2010.

Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

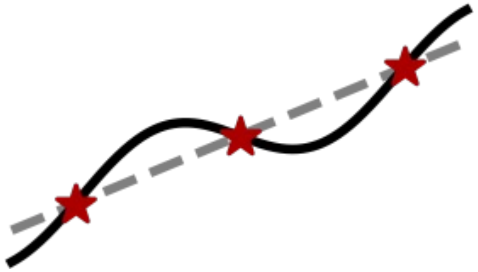
Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

Sparse coding (PSD, LISTA)

Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, NeuralSCS)

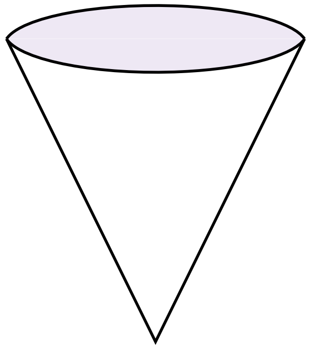
Finding fixed points $y = g(y)$



$$x^\star = \underset{x}{\operatorname{argmin}} \; \frac{1}{2} x^\top Q x + p^\top x \quad \text{subject to} \;\; b - Ax \in \mathcal{K}$$

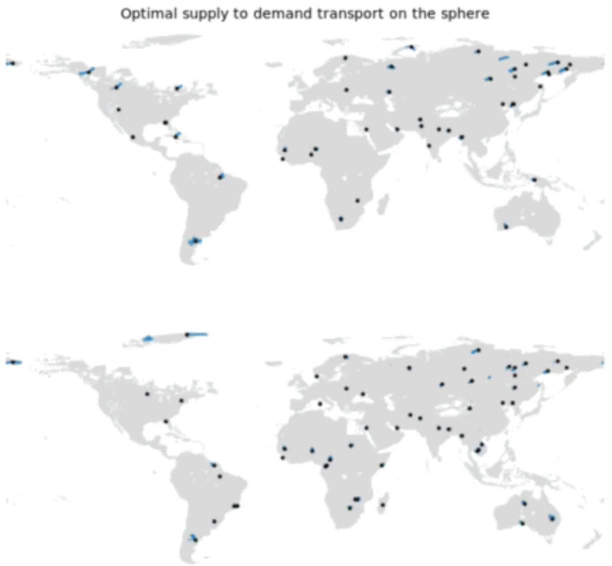
KKT conditions

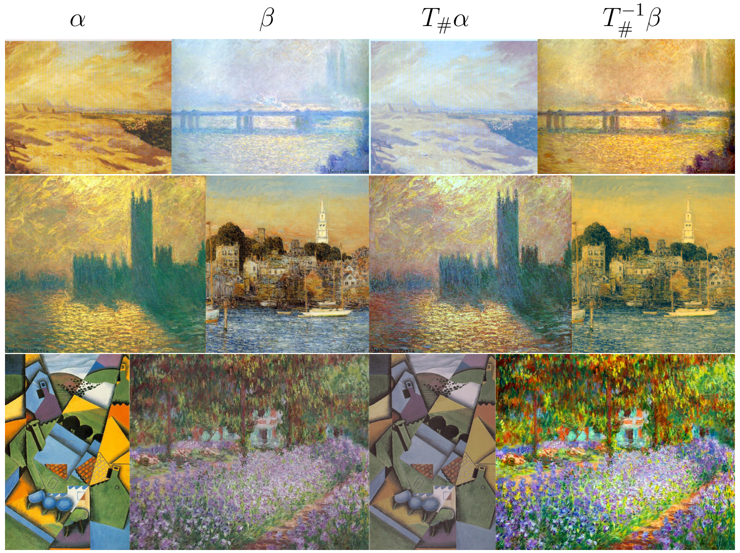
Find $z^\star$ s.t. $\mathcal{R}(z^\star, \theta) = 0$



Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.





Optimal transport (slicing, conjugation, Meta Optimal Transport)

$$T^{\star}(\alpha,\beta) \in \underset{T\in\mathcal{C}(\alpha,\beta)}{\operatorname{argmin}}\ \mathbb{E}_{x\sim\alpha}\|x - T(x)\|_2^2$$

Meta Optimal Transport. Amos et al., ICML 2023.

Applications of amortized optimization

Tutorial on amortized optimization for learning to optimize over continuous domains. Amos, Foundations and Trends in Machine Learning 2023.

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

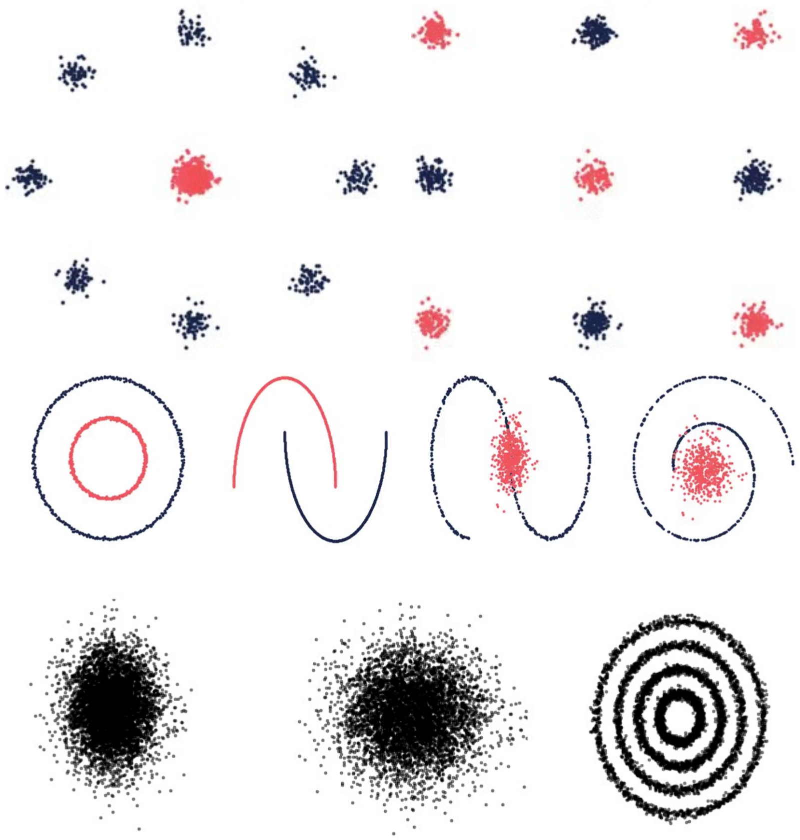
Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

Sparse coding (PSD, LISTA)

Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, N

Optimal transport (slicing, conjugation, Meta Optimal Transport)



$$f^c(y) = -\inf_x f(x) - x^\top y$$

On amortizing convex conjugates for optimal transport. Amos, ICLR 2023

Applications of amortized optimization

Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

Sparse coding (PSD, LISTA)

Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, NeuralSCS)

Optimal transport (slicing, conjugation, Meta Optimal Transport)

Foundations and Trends® in Machine Learning

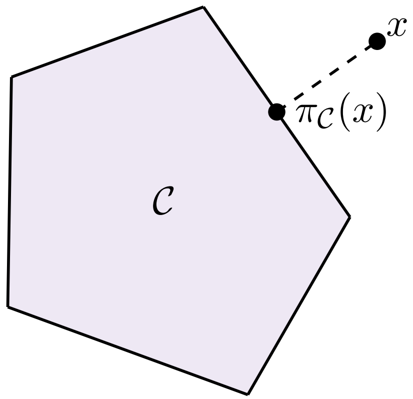
Tutorial on amortized optimization for learning to optimize over continuous domains

Brandon Amos BDA@FB.COM
Facebook AI Research, Meta

Future directions and limitations

Amortized optimization is established and budding with new methods and applications

Possible to expand far beyond **unconstrained continuous Euclidean optimization** settings:
1. **New applications and settings for semi-amortized modeling**
2. **Constrained domains** (e.g., with differentiable projections)
3. **Discrete optimization settings** (e.g., with differentiable discrete optimization)
4. **Non-Euclidean settings** (e.g., with Riemannian optimization)



Potential limitations:
1. Difficult in **out-of-domain settings** when the contexts significantly change
2. Generally difficult to **ensure stability or convergence**
3. Typically **does not solve previously intractable problems**
4. Can be **difficult to obtain high-accuracy solutions** without fine-tuning/semi-amortization

Amortized optimization

Brandon Amos
Meta AI NYC, Fundamental AI Research (FAIR)
brandondamos bamos.github.io

github.com/facebookresearch/amortized-optimization-tutorial
github.com/bamos/presentations

The differentiable cross-entropy method [Amos and Yarats, ICML 2020]
Neural Potts Model [Sercu*, Verkuil*, et al., MLCB 2020]
On the model-based stochastic value gradient [Amos, Stanton, Yarats, Wilson, L4DC 2021]
Online planning via RL fine-tuning [Fickinger*, Hu*, et al., NeurIPS 2021]
Neural fixed-point acceleration [Venkataraman and Amos, ICML AutoML Workshop, 2021]
On amortizing convex conjugates for optimal transport [Amos, ICLR, 2023]
Meta Optimal Transport [Amos, Cohen, Luise, Redko, ICML 2023]
Tutorial on amortized optimization [Amos, Foundations and Trends in ML, 2023]