Amortized optimization for optimal transport

Brandon Amos • Meta (FAIR) NYC

http://github.com/bamos/presentations

What is optimal transport?

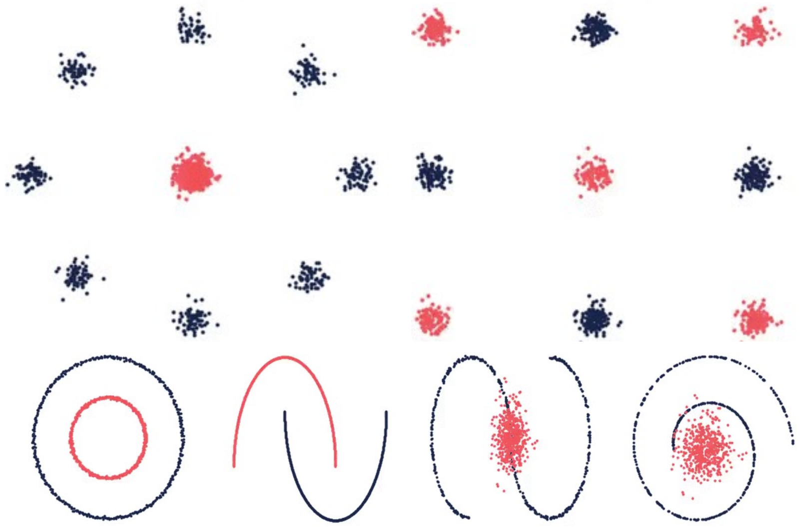
a way of **connecting probability measures**

Optimal transport: old and new. Villani, 2009.
Optimal Transport in Learning, Control, and Dynamical Systems. Bunne and Cuturi, ICML 2023 Tutorial.
Computational Optimal Transport. Peyré and Cuturi, Foundations and Trends in Machine Learning, 2019.
Optimal Transport for Applied Mathematicians. Santambrogio, Birkhäuser, 2015
Optimal Transport in Systems and Control. Chen, Georgiou, and Pavon, Annual Review of Control, Robotics, and Autonomous Systems, 2021.
Optimal mass transport: Signal processing and machine-learning applications. Kolouri et al., 2017.

Monge's problem (squared Euclidean)

$$\inf_{T\in\mathcal{T}(\alpha,\beta)} \int_{\mathcal{X}} \|T(x)-x\|_2^2 \mathrm{d}\alpha(x)$$

find a map connecting α and β that minimally displaces mass



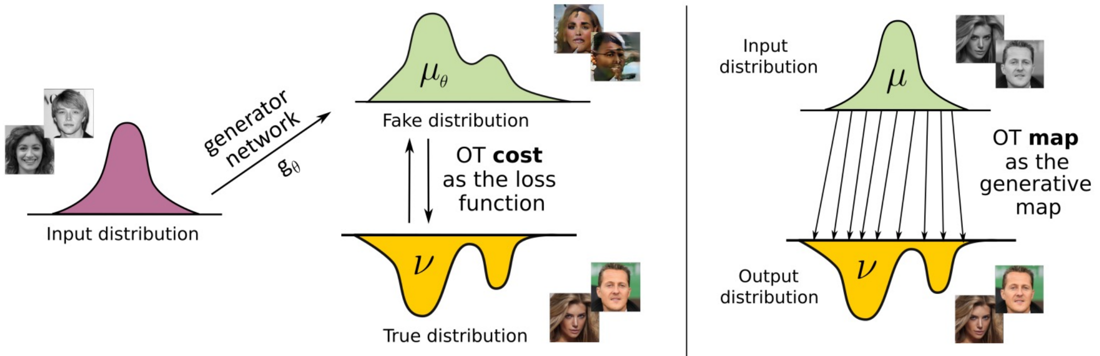
On amortizing convex conjugates for optimal transport. Amos, ICLR 2023.

Why optimal transport? (selected ML-focused highlights)

Defines a **metric on the space of measures**

(metricizes the space of weak convergence)

- *Wasserstein GAN.* Arjovsky, Chintala, Bottou, ICML 2017.
- *Generalized sliced Wasserstein distances.* Kolouri et al., NeurIPS 2019.
- *Sliced wasserstein distance for learning GMMs.* Kolouri et al., CVPR 2018.
- *Convolutional Wasserstein Distances on Geometric Domains*. Solomon et al., ToG 2015.



(a) OT cost as the loss for the generative model. (b) OT map as the generative model.

source: Rout et al.

Couples measures without pairwise data

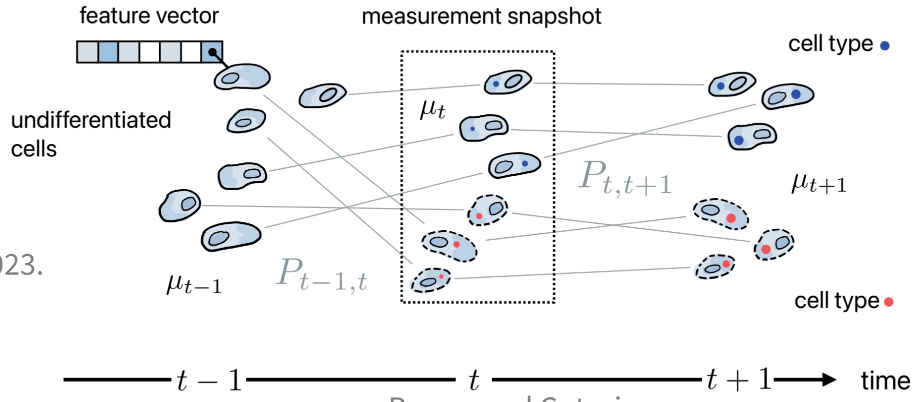
(e.g., for generative modeling, domain adaptation)

- *Generative modeling via OT maps.* Rout, Korotin, Burnaev. ICLR 2022.
- *Neural Optimal Transport.* Korotin et al., ICLR 2023
- *Neural Monge map estimation*. Jiaojiao Fan et al., TMLR 2023.
- *Joint distribution optimal transportation for domain adaptation.* Courty et al., NeurIPS 2017.
- *Geometric Dataset Distances via Optimal Transport.* Alvarez-Melis et al., NeurIPS 2020.

Finds interpolating paths between populations

(e.g., for cell populations or multi-agent systems)

- *Optimal-transport analysis of single-cell gene expression.* Schiebinger et al., Cell 2019.
- *Learning single-cell perturbation responses using neural optimal transport*. Bunne et al., Nature Methods 2023.
- *Likelihood Training of Schrödinger Bridge.* Liu, Horng, Theodorou. ICLR 2022.
- *Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics.* Tong et al., ICML 2020.



source: Bunne and Cuturi

Optimization problems and sub-problems in OT

📚 *GeomLoss*. Feydy et al., AISTATS 2019. 📚 *Python Optimal Transport.* Flamary et al., JMLR 2021. 📚 *Optimal Transport Tools.* Cuturi et al., 2022.

Primal formulations

Monge (squared Euclidean)

$$\inf_{T\in\mathcal{T}(\alpha,\beta)} \int_{\mathcal{X}} \|T(x)-x\|_2^2 \mathrm{d}\alpha(x)$$

Kantorovich

...

Kantorovich dual

$$\hat{\psi}(\alpha,\beta,c) \in \operatorname*{argsup}_{\psi\in L^1(\alpha)} \int_{\mathcal{Y}} \psi^c(y)d\beta(y) - \int_{\mathcal{X}} \psi(x)d\alpha(x)$$

c-transform

$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \ \psi(x) + c(x,y)$$

Costs

Squared Euclidean

$$c(x,y) = \|x-y\|_2^2$$

Lagrangian (e.g., geodesics)

$$c(x,y) = \inf_{\gamma\in\mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t,\dot{\gamma}_t)\mathrm{d}t$$

Easy when convex, finite-dimensional

Linear programming, Sinkhorn (entropic), forward-backward

May be hard otherwise

continuous, high-dimensional

Can ML help /solve/ OT problems? Yes!

by rapidly **predicting approximate solutions**

📚 *Tutorial on amortized optimization.* Amos. FnT in ML, 2023.

Primal formulations

Monge (squared Euclidean)

$$\inf_{T \in \mathcal{T}(\alpha,\beta)} \int_{\mathcal{X}} \|T(x) - x\|_2^2 \mathrm{d}\alpha(x)$$

Kantorovich

...

Kantorovich dual

amortize this

$$\hat{\psi}(\alpha, \beta, c) \in \operatorname*{argsup}_{\psi \in L^1(\alpha)} \int_{\mathcal{Y}} \psi^c(y) d\beta(y) - \int_{\mathcal{X}} \psi(x) d\alpha(x)$$

c-transform

$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \ \psi(x) + c(x, y)$$

amortize that

Costs

Squared Euclidean

$$c(x, y) = \|x - y\|_2^2$$

Lagrangian (e.g., geodesics)

and this

$$c(x, y) = \inf_{\gamma \in \mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t) \mathrm{d}t$$

See also:

📚 *Amortized projection optimization for sliced Wasserstein.* Nguyen et al., NeurIPS 2022.

Why call it *amortized* optimization?

📚 *Tutorial on amortized optimization.* Amos. FnT in ML, 2023.

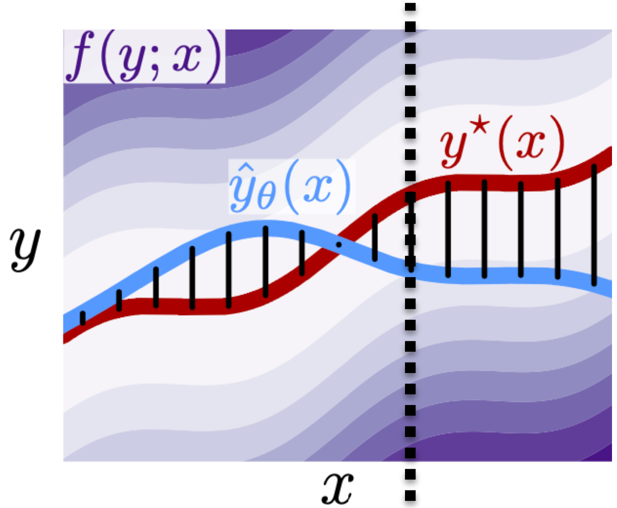
*also referred to as *learned* optimization

to amortize: *to spread out an upfront cost over time*

$$\hat{y}_\theta(x) \approx y^\star(x) \in \underset{y \in \mathcal{Y}(x)}{\operatorname{argmin}} f(y; x)$$

expensive upfront cost

training the model → fast approximate solutions



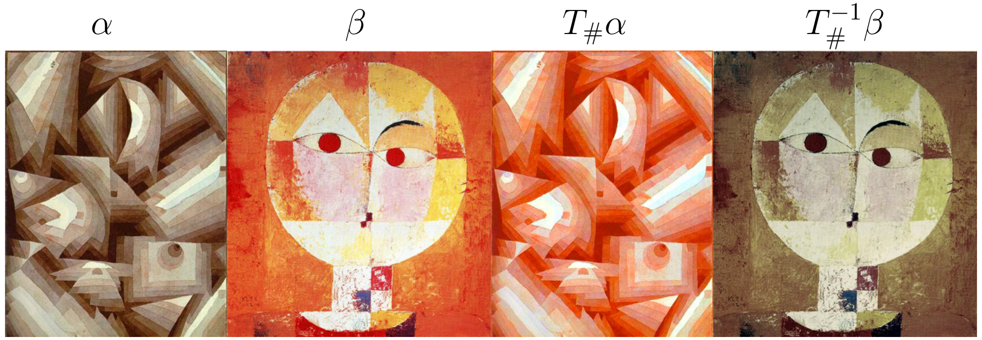
Vertical slices are optimization problems

This talk: amortized optimization for OT

OT problems (Monge maps, dual potentials)

- *Supervised training of conditional Monge maps.* Bunne et al., NeurIPS 2022.
- *Meta Optimal Transport.* Amos et al., ICML 2023.

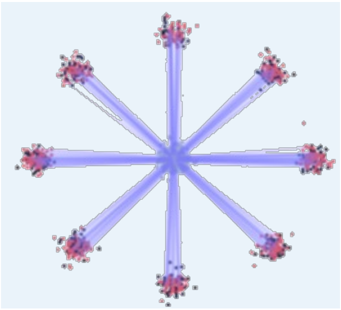
$$\hat{\psi}(\alpha, \beta, c) \in \underset{\psi \in L^1(\alpha)}{\operatorname{argsup}} \int_{\mathcal{Y}} \psi^c(y) d\beta(y) - \int_{\mathcal{X}} \psi(x) d\alpha(x)$$



The c-transform (e.g., the convex conjugate)

- *Optimal transport mapping via input convex neural networks.* Makkuva et al., ICML 2020.
- *Wasserstein-2 Generative Networks.* Korotin et al., ICLR 2021.
- *On amortizing convex conjugates for optimal transport.* Amos, ICLR 2023.

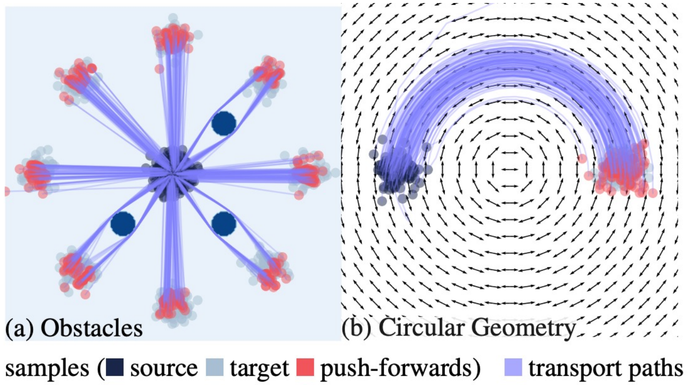
$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \ \psi(x) + c(x, y)$$



Lagrangian costs (e.g., geodesic distances)

- *Deep Generalized Schrödinger Bridge.* Liu et al., NeurIPS 2022.
- *Riemannian metric learning via optimal transport.* Scarvelis and Solomon, ICLR 2023.
- *Neural Lagrangian Schrödinger Bridge.* Koshizuka and Sato, ICLR 2023.
- *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023.
- *A Computational Framework for Solving Wasserstein Lagrangian Flows.* Neklyudov et al., 2023.
- *Generalized Schrödinger Bridge Matching.* Liu et al., 2023.

$$c(x, y) = \inf_{\gamma \in \mathcal{P}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t) \mathrm{d}t$$



Challenge: computing OT maps

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

Monge (primal, Wasserstein-2)

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

we also consider other/discrete OT formulations

Many OT problems are **numerically solved**

Improving OT solvers is active research

Solving multiple OT problems: even harder

Standard solution: independently solve

Meta Optimal Transport

Idea: predict the solution to OT problems with amortized optimization
Simultaneously solve many OT problems, sharing info between instances

Why call it "meta"? Instead of solving a single OT problem, learn how to solve many

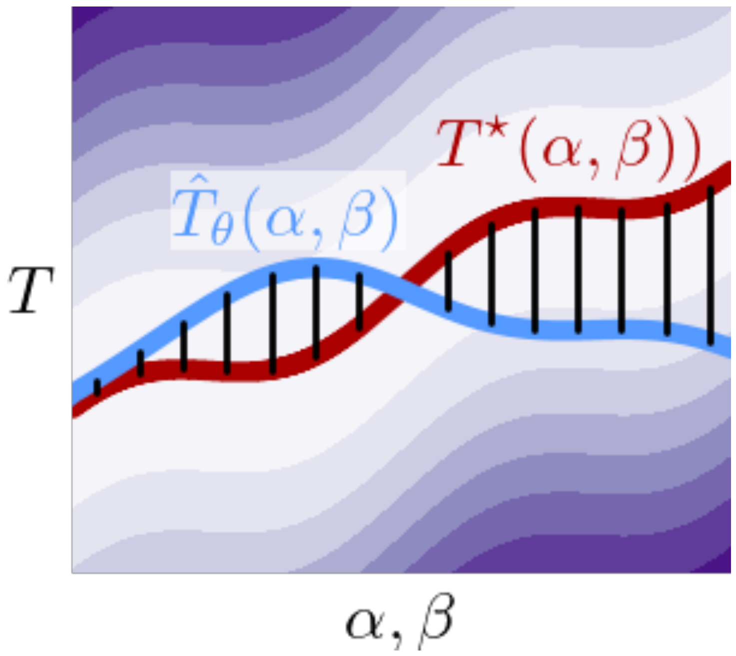
Monge (primal, Wasserstein-2)

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

$$\approx$$

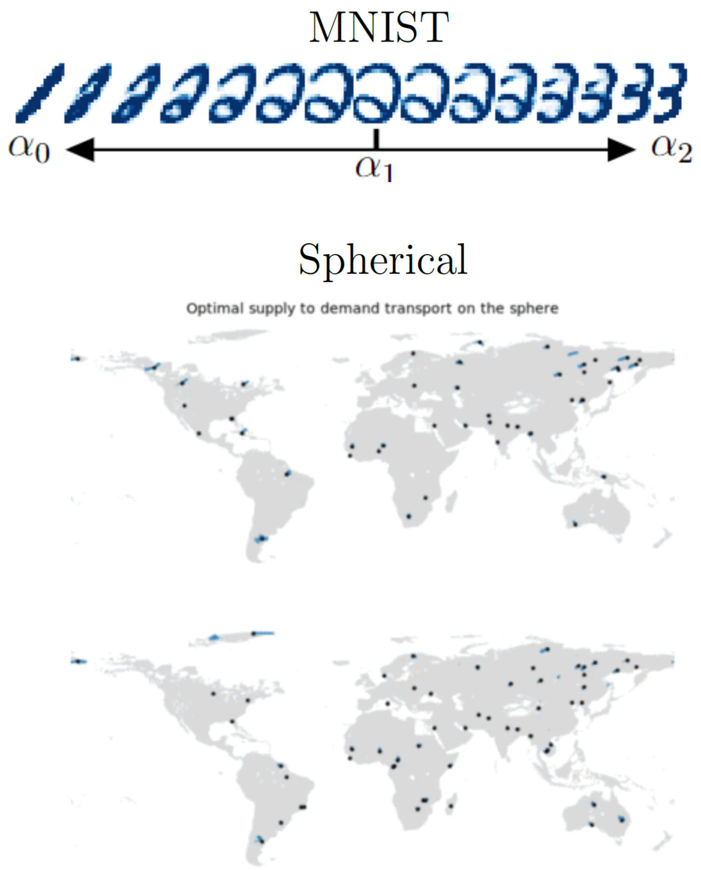
$\hat T_\theta(\alpha,\beta)$ (parameterize dual potential via an MLP)

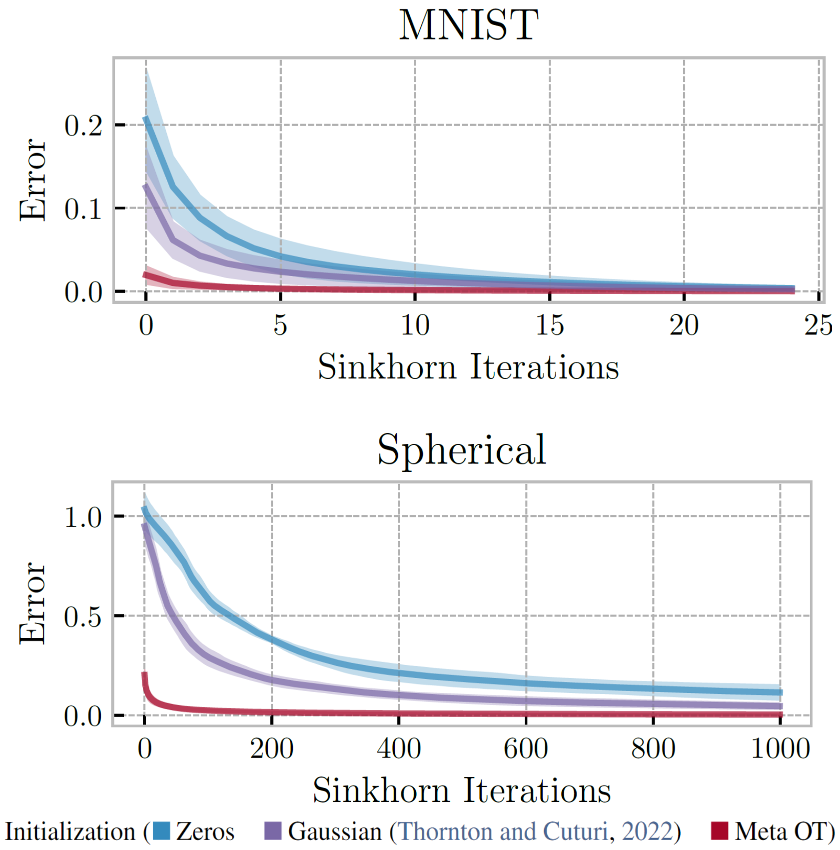
we also consider other/discrete OT formulations



Meta OT for Discrete OT (Sinkhorn)

📚 *Sinkhorn Distances: Lightspeed Computation of Optimal Transport.* Marco Cuturi, NeurIPS 2013





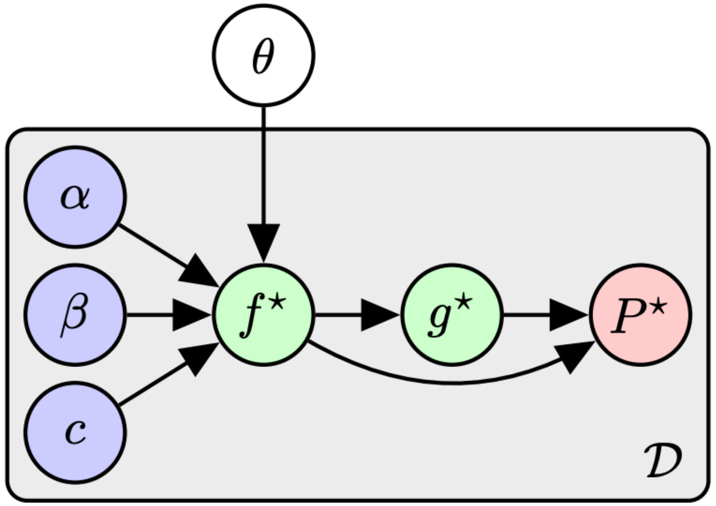


Table 1. Sinkhorn runtime (seconds) to reach a marginal error of 10^{-2}. Meta OT's initial prediction takes $\approx 5 \cdot 10^{-5}$ seconds. We report the mean and std across 10 test instances.

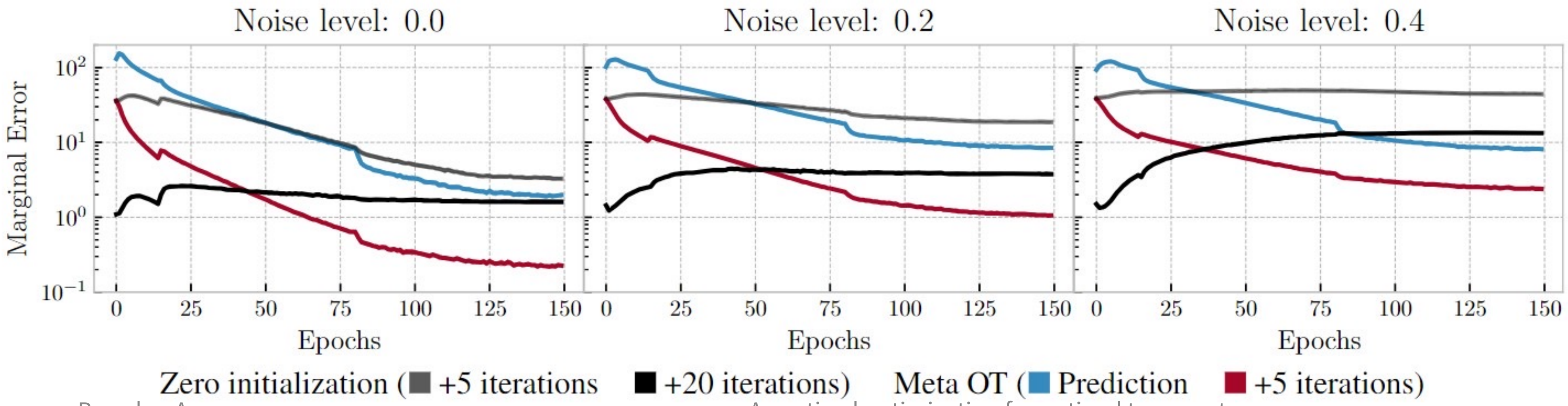
Initialization	MNIST	Spherical
Zeros (t_{zeros})	$4.5 \cdot 10^{-3} \pm 1.5 \cdot 10^{-3}$	0.88 ± 0.13
Gaussian	$4.1 \cdot 10^{-3} \pm 1.2 \cdot 10^{-3}$	$0.56 \pm 9.9 \cdot 10^{-2}$
Meta OT (t_{Meta})	$2.3 \cdot 10^{-3} \pm 9.2 \cdot 10^{-6}$	$7.8 \cdot 10^{-2} \pm 3.4 \cdot 10^{-2}$
Improvement ($t_{\text{zeros}}/t_{\text{Meta}}$)	1.96	11.3

Wasserstein adversarial regularization

Wasserstein adversarial regularization for learning with label noise. Kilian Fatras et al., TPAMI 2021.

Setting: discrete OT for classification with label noise

OT is **repeatedly solved** across minibatches
Use Meta OT to **learn better solutions**



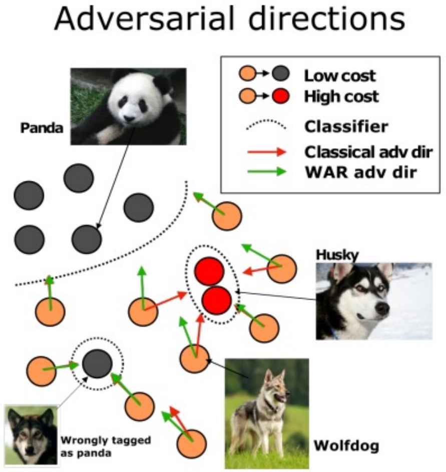
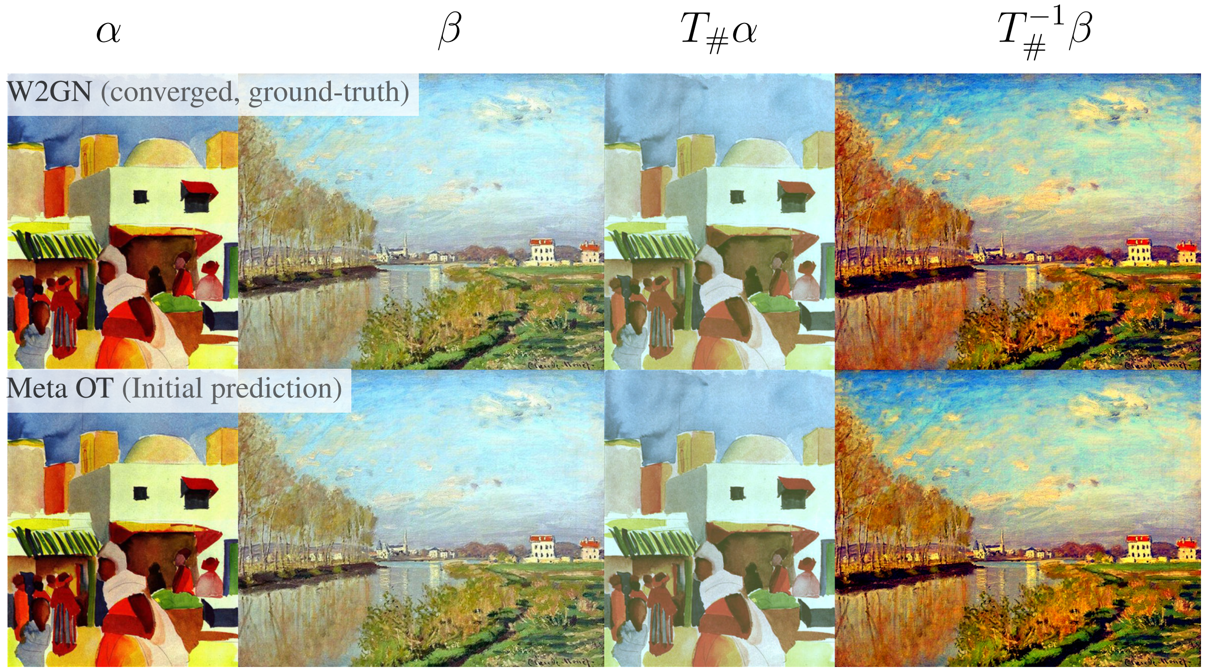


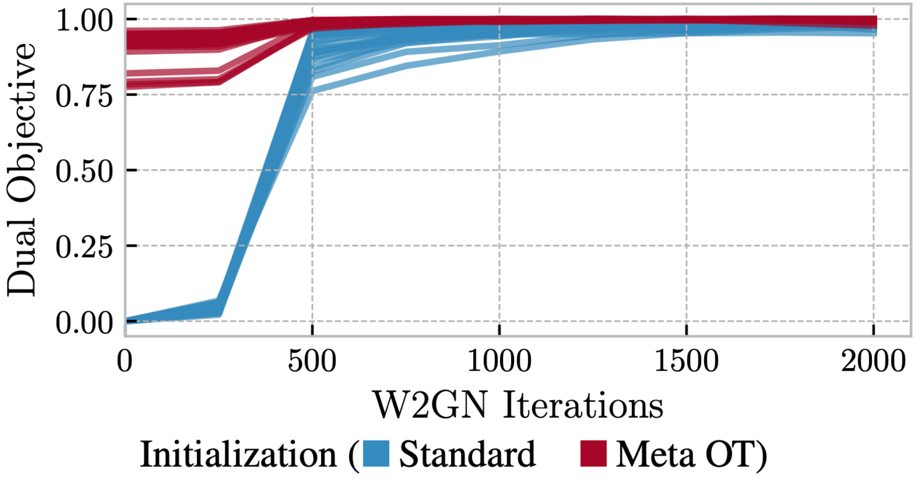
Fig. 1: AR vs. WAR. Given a number of samples, both methods regularize along adversarial directions (arrows in the left panel), leading to updated decision functions (right panel). While both regularizations prevent the classifier to overfit on the noisy labelled sample, AR also tends over-smooth between similar classes (*wolfdog* and *husky*), while WAR preserves them by changing the adversarial direction.

Meta OT in continuous settings (W2GN)

📚 *Wasserstein-2 Generative Networks.* Alexander Korotin et al., ICLR 2021.

RGB color palette transport

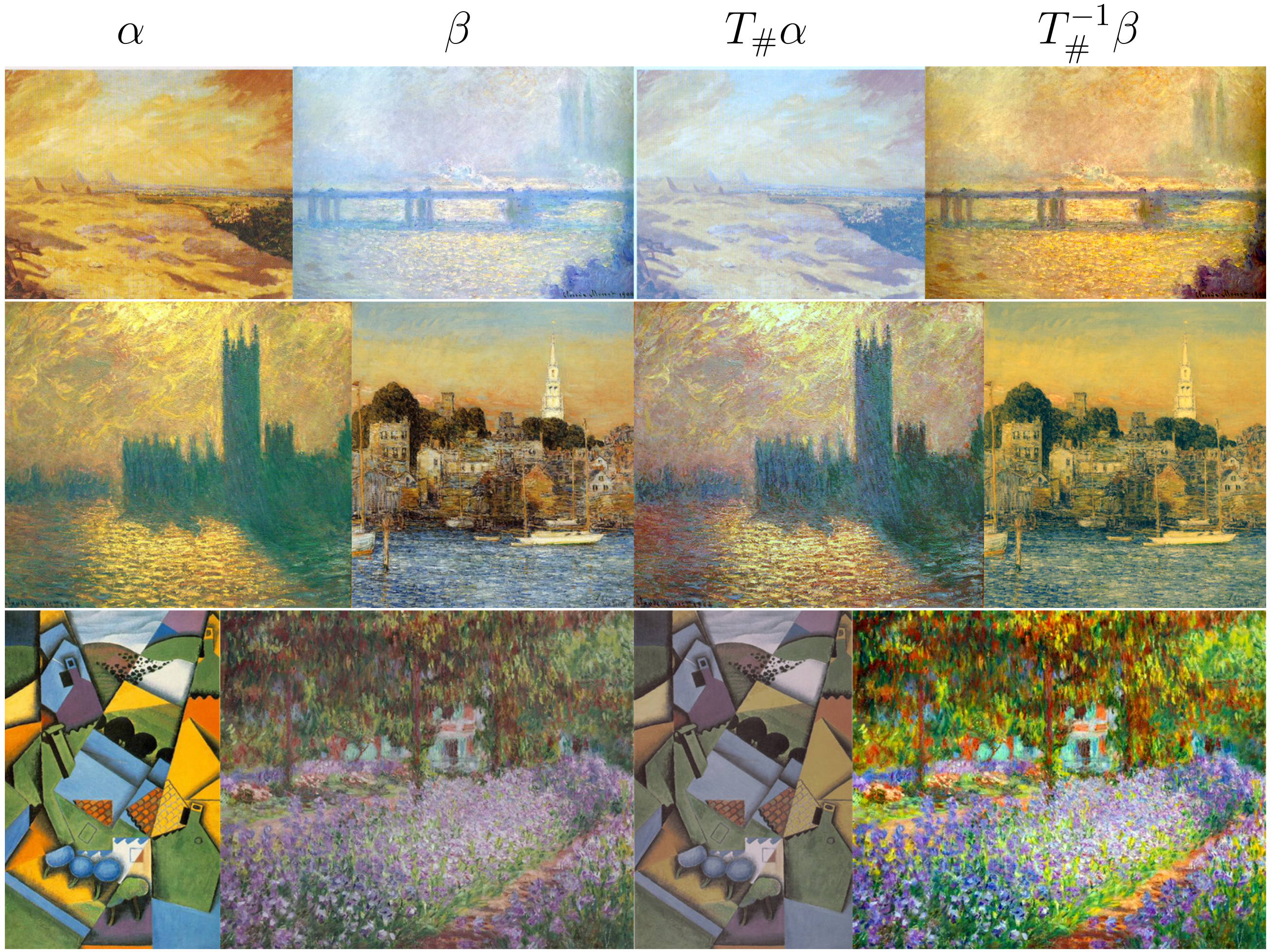


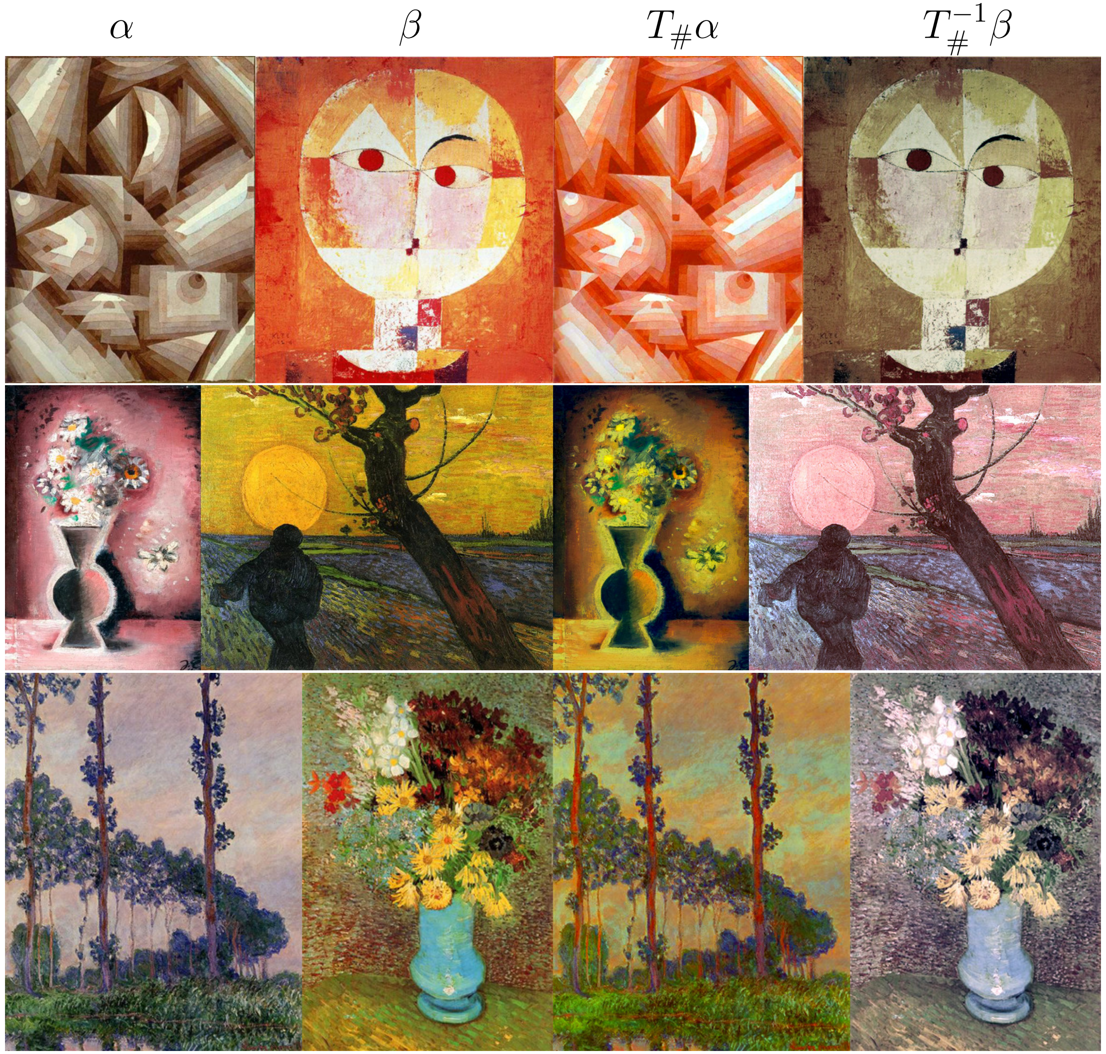


	Iter	Runtime (s)	Dual Value
Meta OT + W2GN	None	$3.5 \cdot 10^{-3} \pm 2.7 \cdot 10^{-4}$	$0.90 \pm 6.08 \cdot 10^{-2}$
	1k	$0.93 \pm 2.27 \cdot 10^{-2}$	$1.0 \pm 2.57 \cdot 10^{-3}$
	2k	$1.84 \pm 3.78 \cdot 10^{-2}$	$1.0 \pm 5.30 \cdot 10^{-3}$
W2GN	1k	$0.90 \pm 1.62 \cdot 10^{-2}$	$0.96 \pm 2.62 \cdot 10^{-2}$
	2k	$1.81 \pm 3.05 \cdot 10^{-2}$	$0.99 \pm 1.14 \cdot 10^{-2}$

More Meta OT color transfer predictions

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





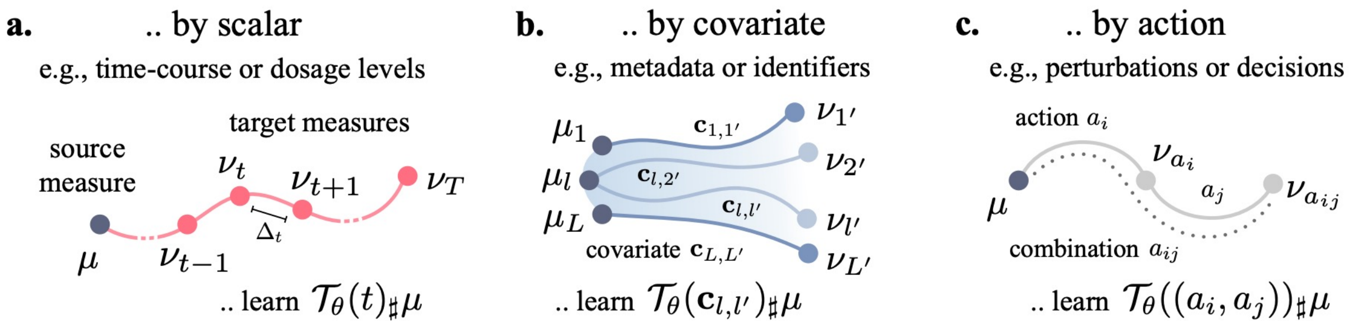
Conditional Monge Maps

📚 *Supervised Training of Conditional Monge Maps.* Bunne, Krause, Cuturi, NeurIPS 2022.

Focus: predicting drug treatments with OT
Idea: condition OT map on patient information

Methodological differences
Conditional Monge Maps ≈ Neural Processes
Predict conditioning inputs of the OT map

Meta OT ≈ Hyper-Networks
Predict parameters of an OT map



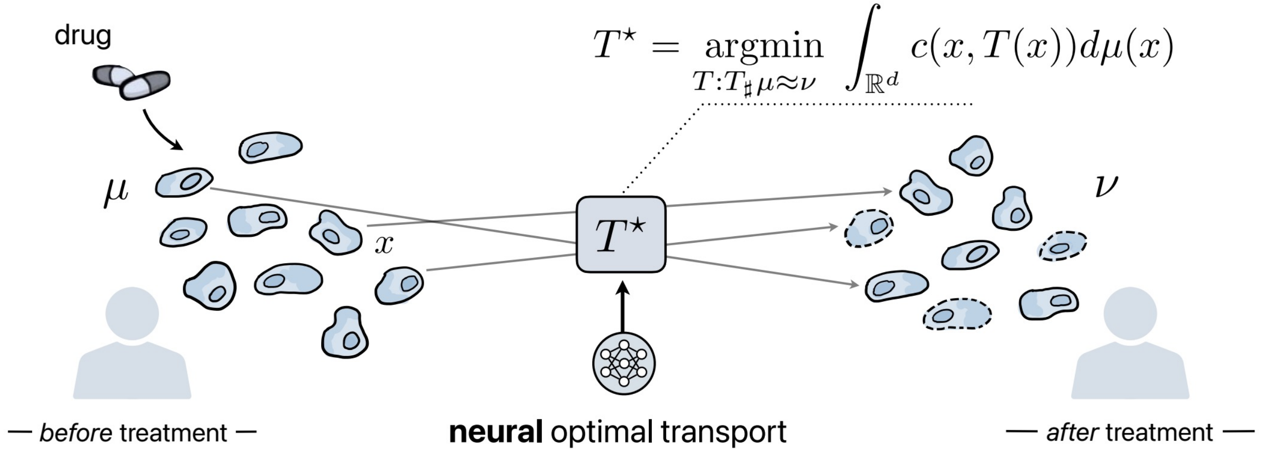


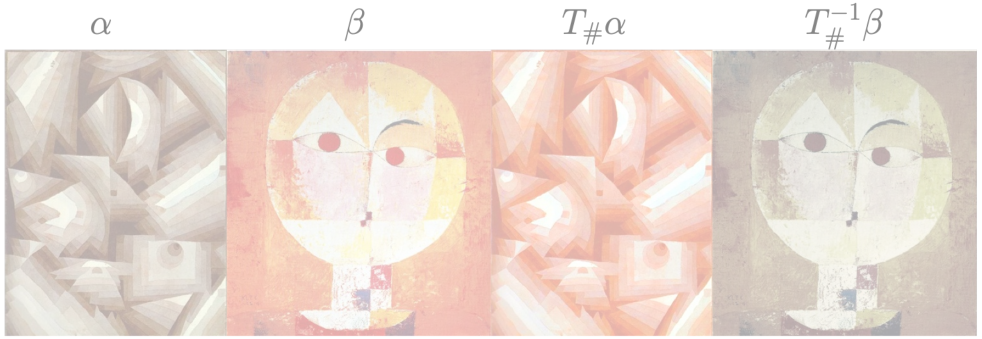
image sources: Bunne and Cuturi

This talk: amortized optimization for OT

OT problems (Monge maps, dual potentials)

- *Supervised training of conditional Monge maps.* Bunne et al., NeurIPS 2022.
- *Meta Optimal Transport.* Amos et al., ICML 2023.

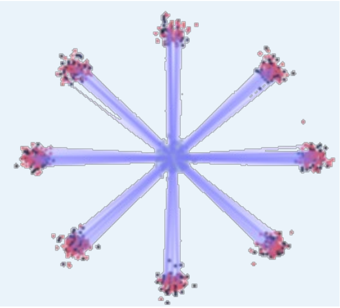
$$\hat{\psi}(\alpha, \beta, c) \in \operatorname*{argsup}_{\psi \in L^1(\alpha)} \int_{\mathcal{Y}} \psi^c(y) d\beta(y) - \int_{\mathcal{X}} \psi(x) d\alpha(x)$$



The c-transform (e.g., the convex conjugate)

- *Optimal transport mapping via input convex neural networks.* Makkuva et al., ICML 2020.
- *Wasserstein-2 Generative Networks.* Korotin et al., ICLR 2021.
- *On amortizing convex conjugates for optimal transport.* Amos, ICLR 2023.

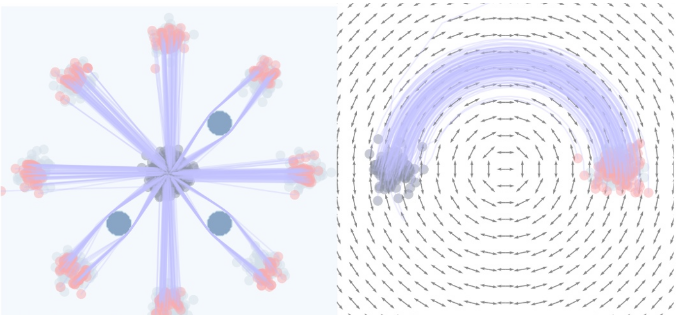
$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \; \psi(x) + c(x, y)$$



Lagrangian costs (e.g., geodesic distances)

- *Deep Generalized Schrödinger Bridge.* Liu et al., NeurIPS 2022.
- *Riemannian metric learning via optimal transport.* Scarvelis and Solomon, ICLR 2023.
- *Neural Lagrangian Schrödinger Bridge.* Koshizuka and Sato, ICLR 2023.
- *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023.
- *A Computational Framework for Solving Wasserstein Lagrangian Flows.* Neklyudov et al., 2023.
- *Generalized Schrödinger Bridge Matching.* Liu et al., 2023.

$$c(x, y) = \inf_{\gamma \in \mathcal{P}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t) \mathrm{d}t$$



(a) Obstacles (b) Circular Geometry

samples (■ source ■ target ■ push-forwards) ■ transport paths

Solving Kantorovich's dual with a neural net

📚 *2-wasserstein approximation via restricted convex potentials*. Taghvaei and Jalali, 2019.
📚 *Three-Player Wasserstein GAN via Amortised Duality*. Nhan Dam et al., IJCAI 2019.
📚 *Optimal transport mapping via input convex neural networks*. Makkuva et al., ICML 2020.
📚 *Wasserstein-2 generative networks.* Korotin et al., ICLR 2020.
📚 *The monge gap.* Uscidda and Cuturi, ICML 2023.
📚 *On amortizing convex conjugates for optimal transport.* Amos, ICLR 2023.

$$\max_\theta \mathcal{V}(\theta) \quad \text{where} \quad \mathcal{V}(\theta) := -\mathop{\mathbb{E}}_{x\sim\alpha}[f_\theta(x)] - \mathop{\mathbb{E}}_{y\sim\beta}[f_\theta^\star(y)]$$

Focus: computing the conjugate

- *2-wasserstein approximation via restricted convex potentials*. Taghvaei and Jalali, 2019.
- *Three-Player Wasserstein GAN via Amortised Duality*. Nhan Dam et al., IJCAI 2019.
- *Optimal transport mapping via input convex neural networks*. Makkuva et al., ICML 2020.
- *Wasserstein-2 generative networks.* Korotin et al., ICLR 2020.
- *The monge gap.* Uscidda and Cuturi, ICML 2023.
- *On amortizing convex conjugates for optimal transport.* Amos, ICLR 2023.

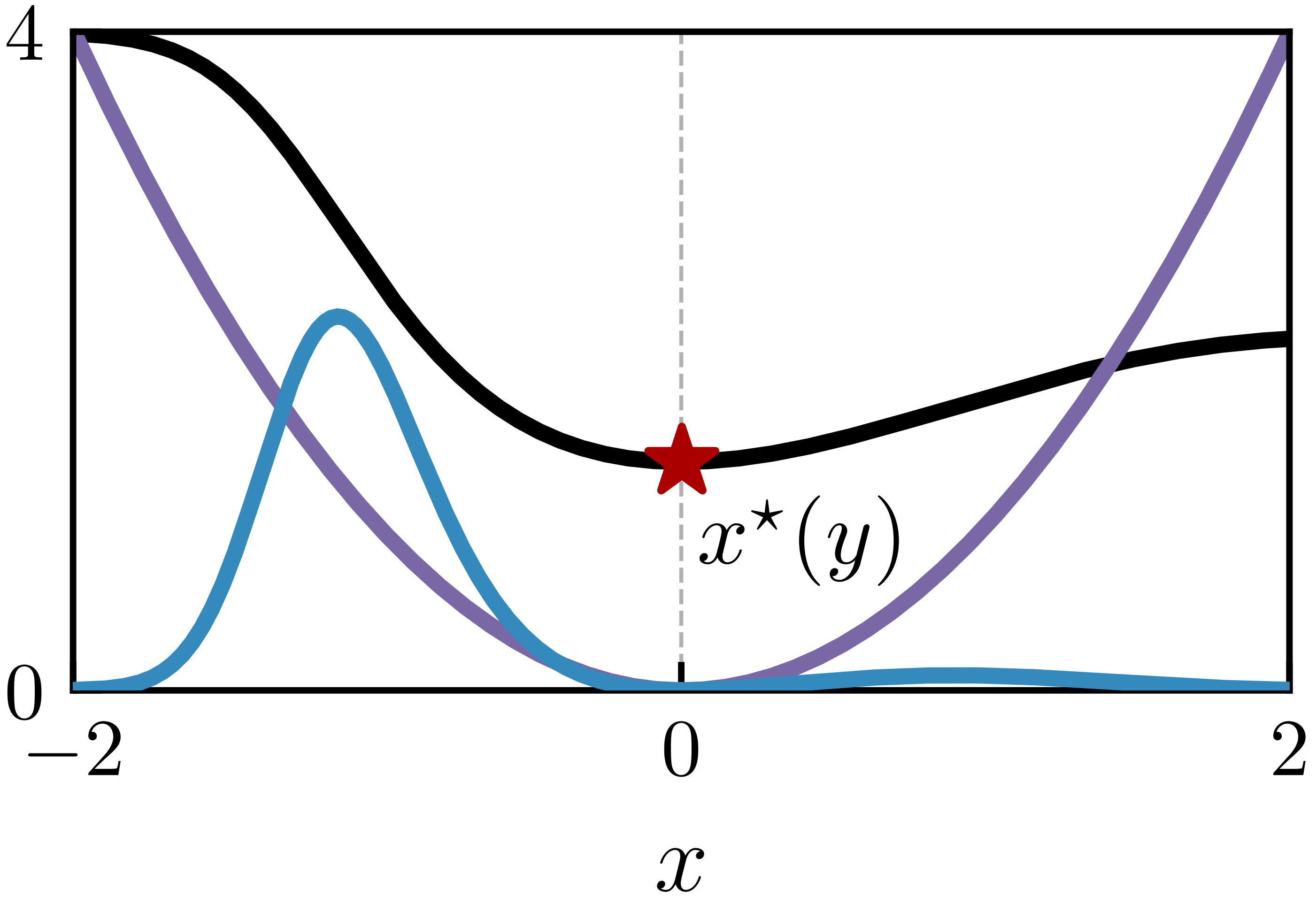
$$\max_\theta \mathcal{V}(\theta) \quad \text{where} \quad \mathcal{V}(\theta) := -\mathop{\mathbb{E}}_{x\sim\alpha}[f_\theta(x)] - \mathop{\mathbb{E}}_{y\sim\beta}[f_\theta^\star(y)]$$

$$f^\star(y) := -\inf_{x\in\mathcal{X}} J_f(x;y) \quad \text{with objective} \quad J_f(x;y) := f(x) - \langle x, y\rangle$$

Amortization: approximate the $\arg\inf$ with (another) neural network

Conjugate amortization loss choices

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



$\|x - x^\star(y)\|_2^2$
Regression

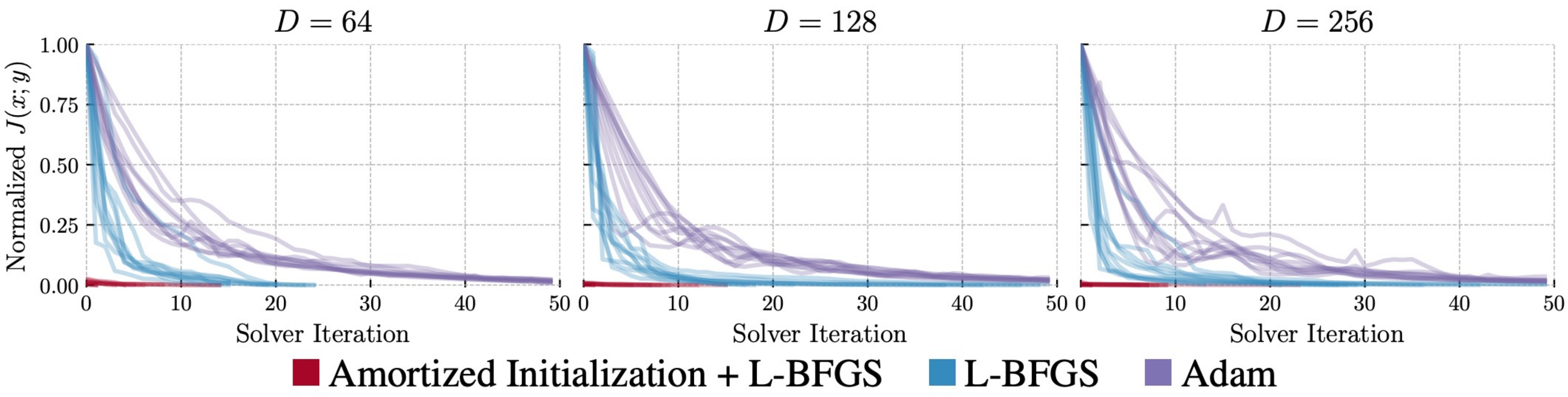
$J_f(x; y)$
Objective

$\propto \|\nabla J_f(x)\|_2^2$
Cycle

Insight: inaccurate predictions are still useful

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

Concern: inaccurate predictions of the conjugate give a biased estimation of the OT objective
Solution: optimality conditions can be checked, prediction can be fine tuned
we know the true conjugate optimization problem, use existing solvers for it



Wasserstein-2 benchmark results

On amortizing convex conjugates for optimal transport. Amos, ICLR 2023.
Do Neural Optimal Transport Solvers Work? Korotin et al., NeurIPS 2021.

Takeaway: amortization choice important, fine-tuning significantly helps

HD benchmarks: Unexplained Variance Percentage (UVP, lower is better)

Baselines from Korotin et al. (2021a)

	Amortization loss	Conjugate solver	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
*[W2]	Cycle	None	0.1	0.7	2.6	3.3	6.0	7.2	2.0	2.7
*[MMv1]	None	Adam	0.2	1.0	1.8	1.4	6.9	8.1	2.2	2.6
*[MMv2]	Objective	None	0.1	0.68	2.2	3.1	5.3	10.1	3.2	2.7
*[MM]	Objective	None	0.1	0.3	0.9	2.2	4.2	3.2	3.1	4.1

Potential model: the input convex neural network described in app. B.3 **Amortization model:** the MLP described in app. B.2

Amortization loss	Conjugate solver	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
Cycle	None	0.28 ±0.09	0.90 ±0.11	2.23 ±0.20	3.03 ±0.06	5.32 ±0.14	8.79 ±0.16	5.66 ±0.45	4.34 ±0.14
Objective	None	0.27 ±0.09	0.78 ±0.12	1.78 ±0.26	2.00 ±0.11	>100	>100	>100	>100
Cycle	L-BFGS	0.26 ±0.09	0.77 ±0.11	1.63 ±0.28	1.15 ±0.14	2.02 ±0.10	4.48 ±0.89	1.65 ±0.10	5.93 ±9.43
Objective	L-BFGS	0.26 ±0.09	0.79 ±0.12	1.63 ±0.30	1.12 ±0.11	1.92 ±0.19	4.40 ±0.79	1.64 ±0.11	2.24 ±0.13
Regression	L-BFGS	0.26 ±0.09	0.78 ±0.12	1.64 ±0.29	1.14 ±0.12	1.93 ±0.20	4.41 ±0.74	1.69 ±0.11	2.21 ±0.15
Cycle	Adam	0.26 ±0.09	0.79 ±0.11	1.62 ±0.29	1.14 ±0.12	1.95 ±0.21	4.55 ±0.62	1.88 ±0.26	>100
Objective	Adam	0.26 ±0.09	0.79 ±0.14	1.62 ±0.31	1.08 ±0.14	1.89 ±0.19	4.23 ±0.76	1.59 ±0.12	1.99 ±0.15
Regression	Adam	0.35 ±0.07	0.81 ±0.12	1.61 ±0.32	1.09 ±0.11	1.85 ±0.20	4.42 ±0.68	1.63 ±0.08	1.99 ±0.16

Potential model: the non-convex neural network (MLP) described in app. B.4 **Amortization model:** the MLP described in app. B.2

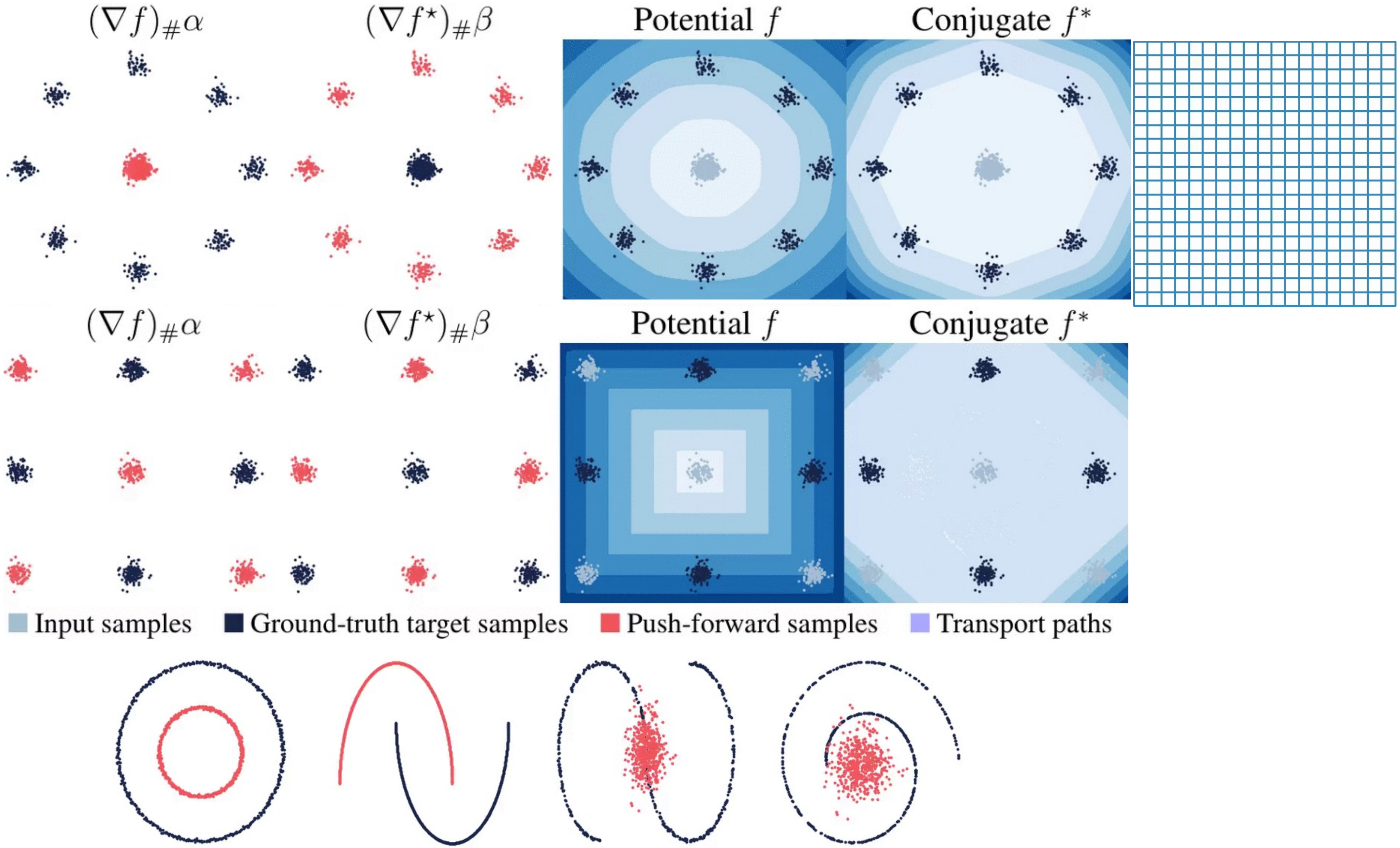
Amortization loss	Conjugate solver	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
Cycle	None	0.05 ±0.00	0.35 ±0.01	1.51 ±0.08	>100	>100	>100	>100	>100
Objective	None	>100	>100	>100	>100	>100	>100	>100	>100
Cycle	L-BFGS	>100	>100	>100	>100	>100	>100	>100	>100
Objective	L-BFGS	0.03 ±0.00	0.22 ±0.01	0.60 ±0.03	0.80 ±0.11	2.09 ±0.31	2.08 ±0.40	0.67 ±0.05	0.59 ±0.04
Regression	L-BFGS	0.03 ±0.00	0.22 ±0.01	0.61 ±0.04	0.77 ±0.10	1.97 ±0.38	2.08 ±0.39	0.67 ±0.05	0.65 ±0.07
Cycle	Adam	0.18 ±0.03	0.69 ±0.56	1.62 ±2.82	>100	>100	>100	>100	>100
Objective	Adam	0.06 ±0.01	0.26 ±0.02	0.63 ±0.07	0.81 ±0.10	1.99 ±0.32	2.21 ±0.32	0.77 ±0.05	0.66 ±0.07
Regression	Adam	0.22 ±0.01	0.28 ±0.02	0.61 ±0.07	0.80 ±0.10	2.07 ±0.38	2.37 ±0.46	0.77 ±0.06	0.75 ±0.09
Improvement factor over prior work		**3.3**	**3.1**	**3.0**	**1.8**	**2.7**	**1.5**	**3.0**	**4.4**

CelebA benchmarks: UVP

	Amortization loss	Conjugate solver	Potential Model	Early Generator	Mid Generator	Late Generator
*[W2]	Cycle	None	ConvICNN64	1.7	0.5	0.25
*[MM]	Objective	None	ResNet	2.2	0.9	0.53
*[MM-R†]	Objective	None	ResNet	1.4	0.4	0.22
	Cycle	None	ConvNet	>100	26.50 ±60.14	0.29 ±0.59
	Objective	None	ConvNet	>100	0.29 ±0.15	0.69 ±0.90
	Cycle	Adam	ConvNet	0.65 ±0.02	0.21 ±0.00	0.11 ±0.04
	Cycle	L-BFGS	ConvNet	0.62 ±0.01	0.20 ±0.00	0.09 ±0.00
	Objective	Adam	ConvNet	0.65 ±0.02	0.21 ±0.00	0.11 ±0.05
	Objective	L-BFGS	ConvNet	0.61 ±0.01	0.20 ±0.00	0.09 ±0.00
	Regression	Adam	ConvNet	0.66 ±0.01	0.21 ±0.00	0.12 ±0.00
	Regression	L-BFGS	ConvNet	0.62 ±0.01	0.20 ±0.00	0.09 ±0.01
	Improvement factor over prior work			**2.3**	**2.0**	**2.4**

† the *reversed* direction from Korotin et al. (2021a), i.e. the potential model is associated with the β measure

Transport between synthetic measures



Learning flows via continuous OT

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

Continuous OT for flows:

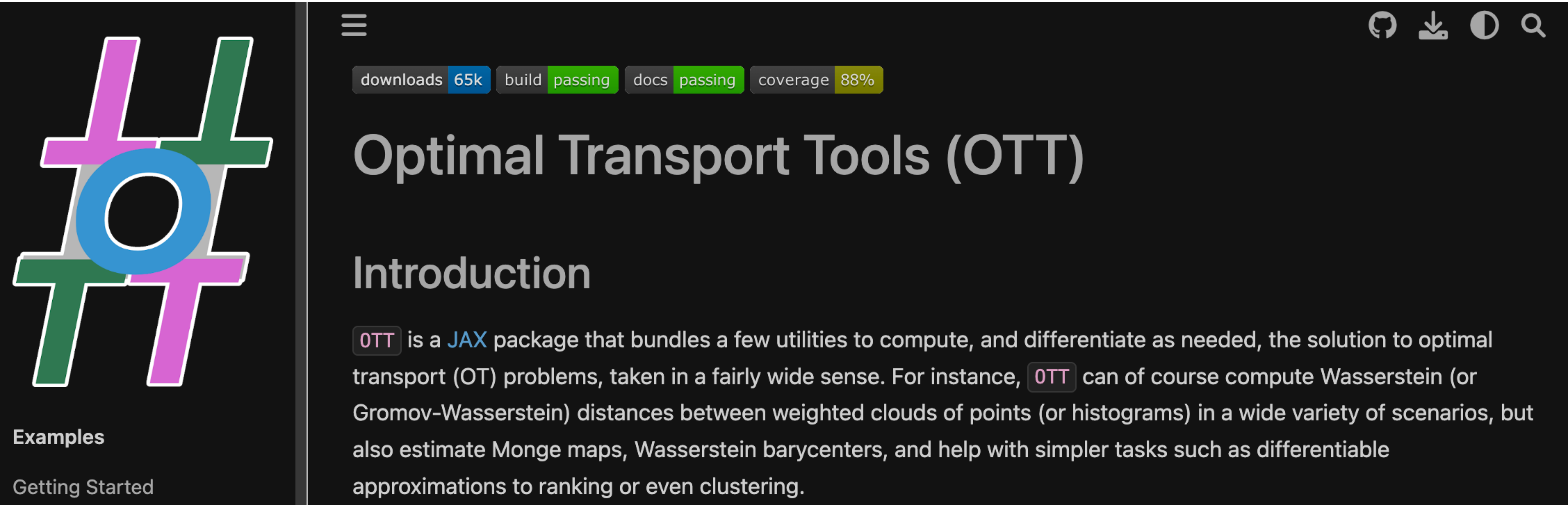
1. Works **only from samples** (no likelihoods needed)
2. No need to explicitly enforce invertibility
3. No need to compute the log-det of the Jacobian

$$p_Y(y) = p_X(f^{-1}(y)) \left| \frac{\partial f^{-1}(y)}{\partial y} \right|$$

Conjugate amortization+fine-tuning in OTT

📚 *Optimal Transport Tools.* Cuturi et al., 2022.

github.com/ott-jax/ott

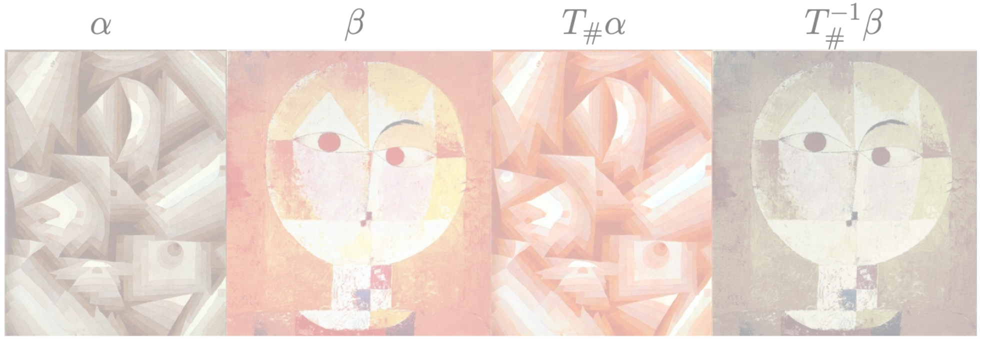


This talk: amortized optimization for OT

OT problems (Monge maps, dual potentials)

- *Supervised training of conditional Monge maps.* Bunne et al., NeurIPS 2022.
- *Meta Optimal Transport.* Amos et al., ICML 2023.

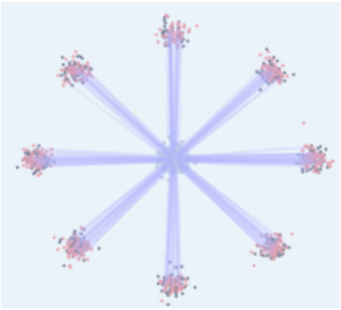
$$\hat{\psi}(\alpha, \beta, c) \in \operatorname*{argsup}_{\psi \in L^1(\alpha)} \int_{\mathcal{Y}} \psi^c(y) d\beta(y) - \int_{\mathcal{X}} \psi(x) d\alpha(x)$$



The c-transform (e.g., the convex conjugate)

- *Optimal transport mapping via input convex neural networks.* Makkuva et al., ICML 2020.
- *Wasserstein-2 Generative Networks.* Korotin et al., ICLR 2021.
- *On amortizing convex conjugates for optimal transport. Amos, ICLR 2023.*

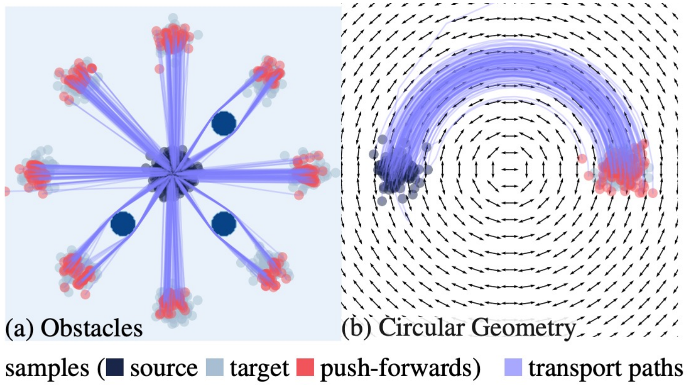
$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \ \psi(x) + c(x, y)$$



Lagrangian costs (e.g., geodesic distances)

- *Deep Generalized Schrödinger Bridge.* Liu et al., NeurIPS 2022.
- *Riemannian metric learning via optimal transport.* Scarvelis and Solomon, ICLR 2023.
- *Neural Lagrangian Schrödinger Bridge.* Koshizuka and Sato, ICLR 2023.
- *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023.
- *A Computational Framework for Solving Wasserstein Lagrangian Flows.* Neklyudov et al., 2023.
- *Generalized Schrödinger Bridge Matching.* Liu et al., 2023.

$$c(x, y) = \inf_{\gamma \in \mathcal{P}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t) \mathrm{d}t$$



From Euclidean to Lagrangian costs

incorporates **physical knowledge** from the world (e.g., obstacles, manifolds)

- 📚 *Deep Generalized Schrödinger Bridge.* Liu et al., NeurIPS 2022.
- 📚 *Riemannian metric learning via optimal transport.* Scarvelis and Solomon, ICLR 2023.
- 📚 *Neural Lagrangian Schrödinger Bridge.* Koshizuka and Sato, ICLR 2023.
- 📚 *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023.
- 📚 *A Computational Framework for Solving Wasserstein Lagrangian Flows*. Neklyudov et al., 2023.
- 📚 *Generalized Schrödinger Bridge Matching.* Liu et al., 2023.

Squared Euclidean

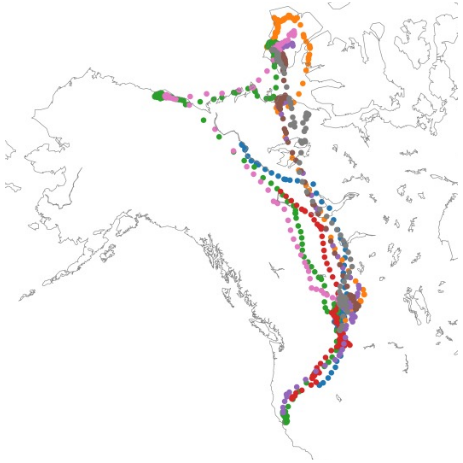
$$c(x,y) = \|x-y\|_2^2$$

→

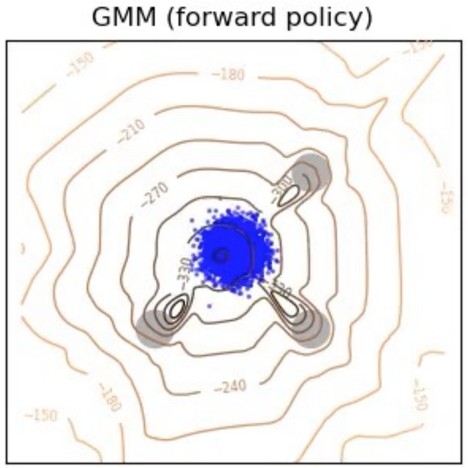
Lagrangian (e.g., geodesics)

$$c(x,y) = \inf_{\gamma \in \mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t)\mathrm{dt}$$

source



source: Scarvelis and Solomon



source: Liu et al., 2023.

Expressivity of Lagrangian costs

📚 *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023. (and many others!)

$$c(x,y) = \inf_{\gamma \in \mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t)\mathrm{d}t$$

Euclidean $c(x,y) = \|x-y\|_2^2$

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2$$

easy, closed-form computation

Potential term (e.g., obstacles)

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2 - U(\gamma_t)$$

Riemannian geodesics

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_{A(\gamma_t)}^2$$

challenging in general, no known closed-form solutions

Our approach: amortize the geodesic path!

enables us to solve the **static OT formulation**

📚 *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023.

$$\tilde{\gamma}_\theta(x,y) \approx \gamma^\star(x,y) = \underset{\gamma \in \mathcal{P}(x,y)}{\operatorname{arginf}} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t)\mathrm{d}t$$

Euclidean $c(x,y) = \|x-y\|_2^2$

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2$$

easy, closed-form computation

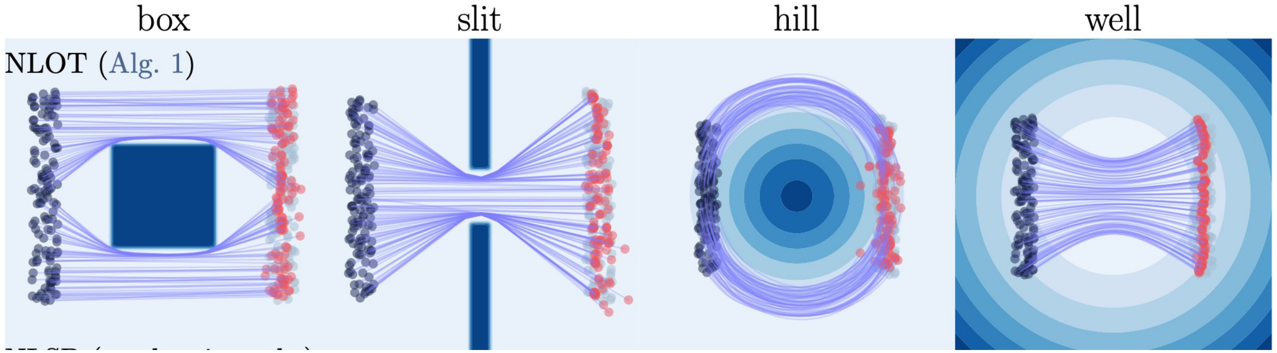
Potential term (e.g., obstacles)

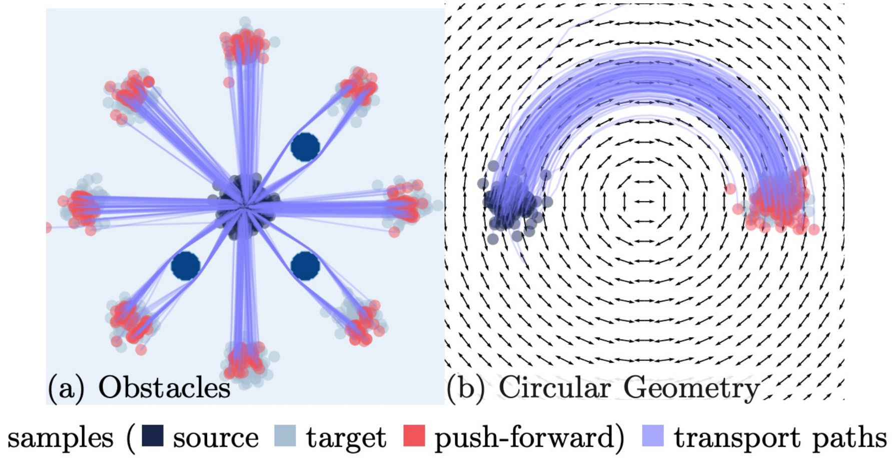
$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2 - U(\gamma_t)$$

Riemannian geodesics

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_{A(\gamma_t)}^2$$

challenging in general, no known closed-form solutions





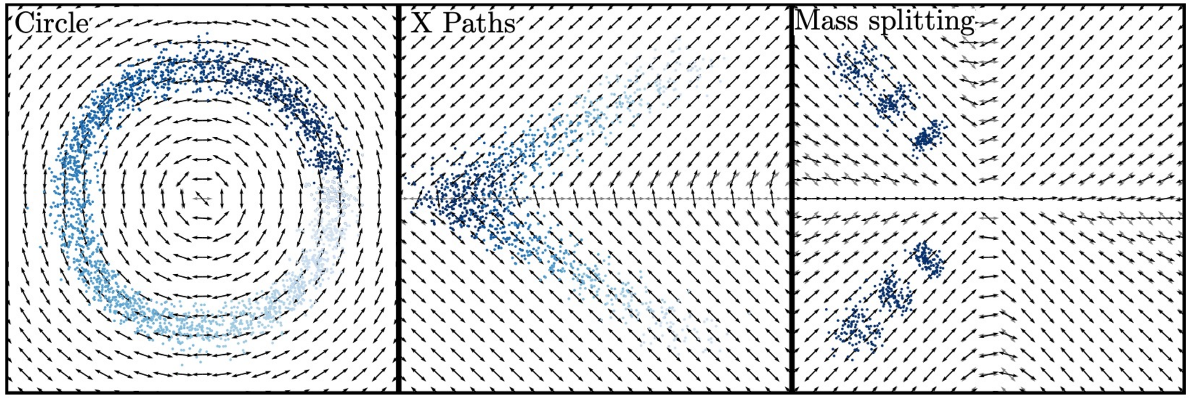
Excels at solving OT and learning metrics

📚 *Neural Optimal Transport with Lagrangian Costs.* Pooladian, Domingo-Enrich, Chen, Amos, 2023. (and many others!)

Table 1: Marginal 2-Wasserstein errors (scaled by 100x) of the push-forward measure on the synthetic settings from Koshizuka and Sato (2022).

	box	slit	hill	well
NLOT (ours)	$\mathbf{1.6 \pm 0.2}$	$\mathbf{1.3 \pm 0.2}$	$\mathbf{1.8 \pm 1.3}$	$\mathbf{1.3 \pm 0.3}$
NLSB (stochastic)	2.4 ± 0.6	$\mathbf{1.3 \pm 0.4}$	$\mathbf{2.0 \pm 0.1}$	2.6 ± 1.6
NLSB (expected)	6.0 ± 0.5	17.6 ± 1.8	4.0 ± 0.9	16.1 ± 3.5

*Results are from training three trials for every method.



smallest eigenvectors of A (■ learned ■ ground-truth)
■ data (lighter colors=later time)

Table 2: Alignment scores $\ell_{\text{align}} \in [0, 1]$ for metric recovery in Fig. 4. (higher is better)

	Circle	Mass Splitting	X Paths
Metric learning with NLOT (ours)	$\mathbf{0.997 \pm 0.002}$	$\mathbf{0.986 \pm 0.001}$	$\mathbf{0.957 \pm 0.001}$
Scarvelis and Solomon (2023)	0.995	0.839	0.916

Amortized optimization beyond OT 🚀

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

Variational inference (amortized VI, 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, Lagrangian costs)

📚 Foundations and Trends® in Machine Learning

Tutorial on amortized optimization

Learning to optimize over continuous spaces

Brandon Amos, *Meta AI*

Amortized optimization for optimal transport

Brandon Amos • Meta (FAIR) NYC

http://github.com/bamos/presentations

Kantorovich dual

amortize this

$$\hat{\psi}(\alpha, \beta, c) \in \underset{\psi \in L^1(\alpha)}{\operatorname{argsup}} \int_{\mathcal{Y}} \psi^c(y) d\beta(y) - \int_{\mathcal{X}} \psi(x) d\alpha(x)$$

***c*-transform**

$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \; \psi(x) + c(x, y)$$

amortize that

Costs

Squared Euclidean

$$c(x, y) = \|x - y\|_2^2$$

Lagrangian (e.g., geodesics)

and this

$$c(x, y) = \inf_{\gamma \in \mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t) \mathrm{d}t$$