On amortizing convex conjugates for optimal transport

Brandon Amos • Meta AI (FAIR) NYC

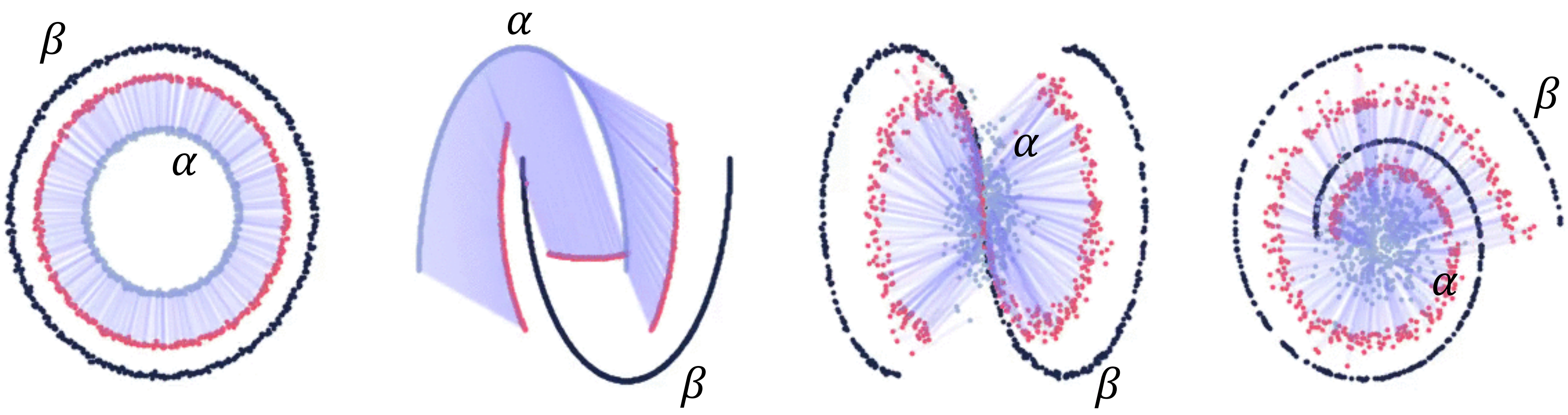
http://github.com/bamos/presentations
http://github.com/facebookresearch/w2ot

Optimal transport connects spaces

Monge (primal)

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

α, β are **measures**
$\mathcal{C}(\alpha,\beta)$ is the set of valid **coupling**
T is a **transport map** from α to β



Duality and continuous OT

Monge (primal)

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

$$\Updownarrow \quad T^\star = \nabla \hat{f}$$ (Brenier's Theorem)

Kantorovich (dual)

$$\hat{f} \in \underset{f\in\mathcal{L}^1(\alpha)}{\operatorname{argmax}} \; -\mathbb{E}_{x\sim\alpha}[f(x)] - \mathbb{E}_{y\sim\beta}[f^\star(y)]$$

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

Solving Kantorovich's dual with a neural net

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

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.

Focus: computing the conjugate

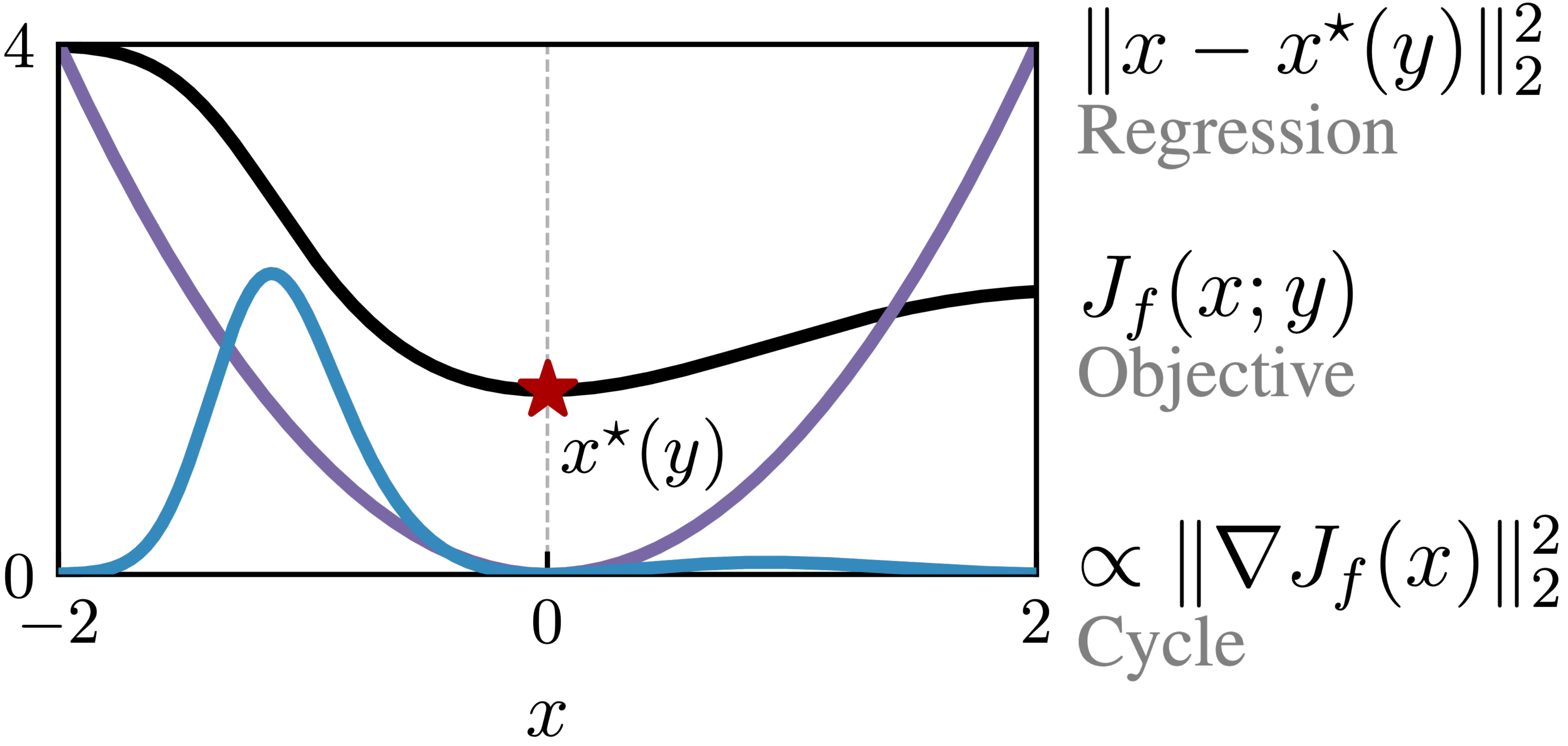
$$\max_\theta \mathcal{V}(\theta) \quad \text{where} \quad \mathcal{V}(\theta) := -\mathop{\mathbb{E}}_{x\sim\alpha}[f_\theta(x)] - \boxed{\mathop{\mathbb{E}}_{y\sim\beta}[f_\theta^\star(y)]}$$

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

Amortization: Approximate the arginf with (another) neural network

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.

Conjugate amortization loss choices



Wasserstein-2 benchmark results

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

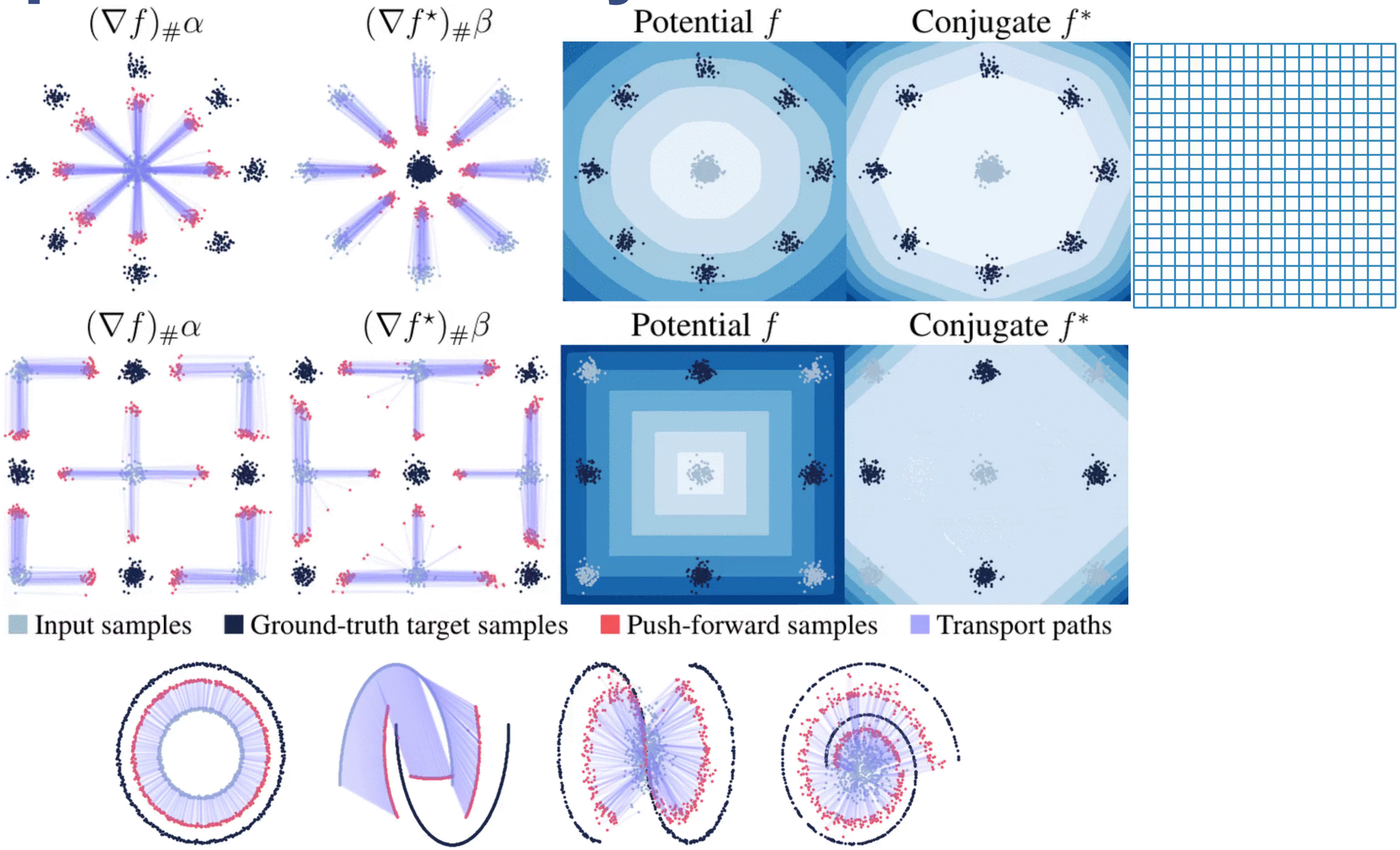
Amortization loss	Conjugate solver	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
Cycle	None	0.05 ±0.00	0.35 ±0.01	1.51 ±0.08	>100	>100	>100	>100	>100
Objective	None	>100	>100	>100	>100	>100	>100	>100	>100
Cycle	L-BFGS	>100	>100	>100	>100	>100	>100	>100	>100
Objective	L-BFGS	0.03 ±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



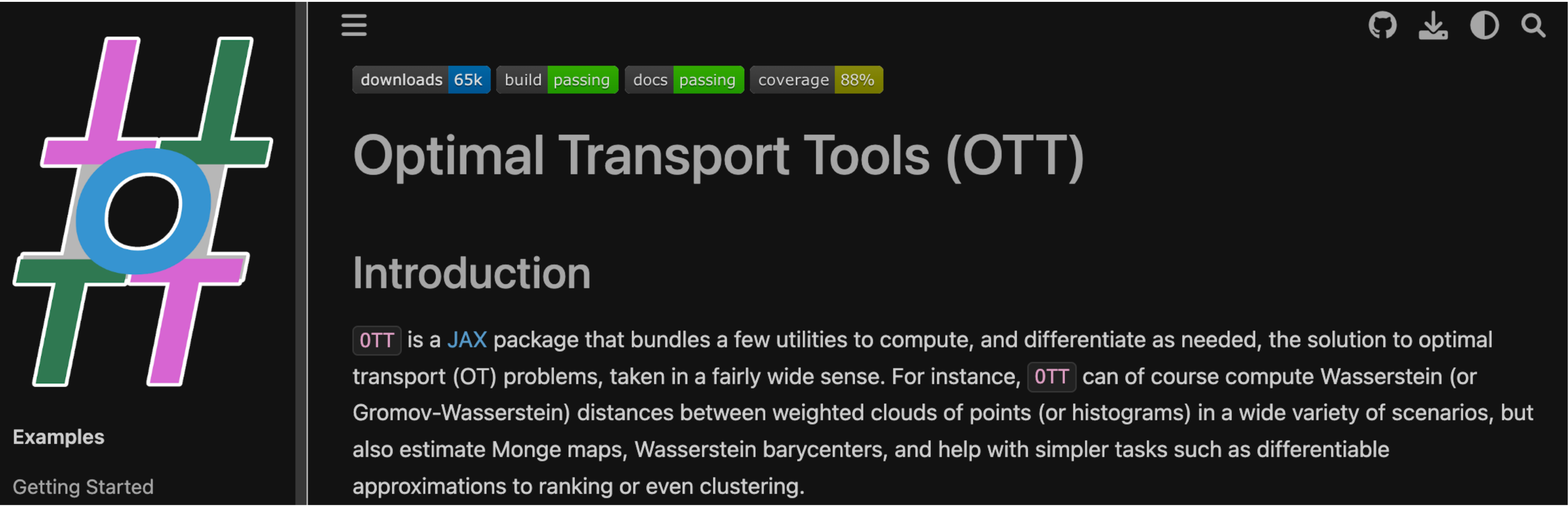
Learning flows via continuous OT

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

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

github.com/ott-jax/ott



downloads 65k | build passing | docs passing | coverage 88%

Optimal Transport Tools (OTT)

Introduction

`OTT` is a JAX package that bundles a few utilities to compute, and differentiate as needed, the solution to optimal transport (OT) problems, taken in a fairly wide sense. For instance, `OTT` can of course compute Wasserstein (or Gromov-Wasserstein) distances between weighted clouds of points (or histograms) in a wide variety of scenarios, but also estimate Monge maps, Wasserstein barycenters, and help with simpler tasks such as differentiable approximations to ranking or even clustering.

Examples

Getting Started

On amortizing convex conjugates for optimal transport

Brandon Amos • Meta AI (FAIR) NYC

http://github.com/bamos/presentations
http://github.com/facebookresearch/w2ot