Differentiable Cost-Parameterized Monge Map Estimators

Samuel Howard [1] **George Deligiannidis** [1] **Patrick Rebeschini** [1] **James Thornton** [1] [2]

Abstract

Within the field of optimal transport (OT), the choice of ground cost is crucial to ensuring that the optimality of a transport map corresponds to being useful in real-world applications. It is therefore desirable to use known information to tailor cost functions and hence learn OT maps which are adapted to the problem at hand. By considering a class of neural ground costs whose Monge maps have a known form, we construct a differentiable Monge map estimator which can be trained to exhibit desirable properties. In doing so, we simultaneously learn both an OT map estimator and a corresponding adapted cost function. Through suitable choices of loss function, our method provides a general approach for incorporating prior information about the *Monge map itself* when learning adapted OT maps and cost functions.

1. Introduction

Optimal transport. Mapping samples between two datasets in a meaningful way is one of the most fundamental tasks across the sciences. Optimal transport (OT) provides a principled framework for such mapping problems, and has enjoyed success in many fields including throughout biology [Schiebinger et al., 2019], and extensively in machine learning with applications in generative modelling [De Bortoli et al., 2021, Genevay et al., 2018], differentiable sorting [Cuturi et al., 2019], clustering [Genevay et al., 2019], resampling [Corenflos et al., 2021] and self supervised learning [Caron et al., 2020].

For two probability measures μ and ν on $\mathbb{R}^d$, the OT problem finds the most efficient way to transport μ to ν relative to a ground cost function $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. The choice of cost function therefore determines the notion of optimality.

Choice of Ground Cost. The squared-Euclidean distance

[1]Department of Statistics, University of Oxford [2]Apple. Correspondence to: Samuel Howard <howard@stats.ox.ac.uk>.
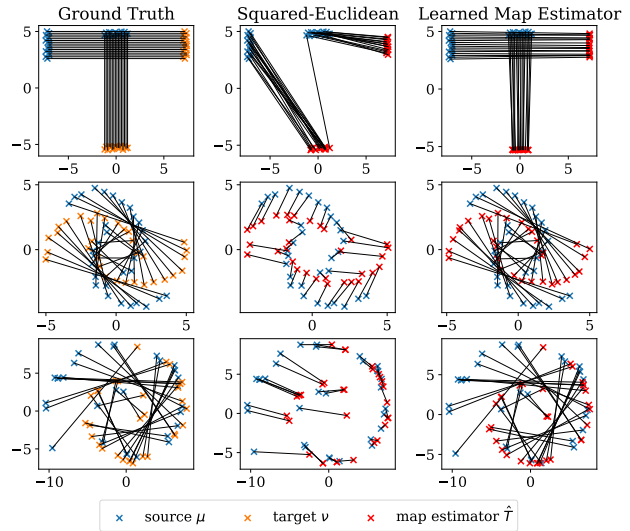
Figure 1. Using the squared-Euclidean cost can result in incorrect map estimators that do not align with known ground-truth mappings. By optimizing a differentiable cost-parameterized Monge map estimator to resemble known information, we can obtain Monge map estimators and corresponding cost functions which are consistent with the ground-truth.

$c(x, y) = \frac{1}{2}\|x - y\|^2$ is the default and most commonly used ground cost in computational OT methods, due to its desirable theoretical properties and ease of implementation. It allows the use of the closed-form transport map from Brenier's theorem [Brenier, 1987], and an elegant connection to convex analysis upon which many methods are based [Makkuva et al., 2020, Korotin et al., 2021a]. However, the squared-Euclidean cost can be an arbitrary choice and may not be suitable for the given problem [Genevay et al., 2018]. Moreover, it can result in erroneous mappings that do not agree with known ground-truth information [Somnath et al., 2023], as demonstrated in Figure 1. Recently, several works have proposed methods to estimate OT maps for more general ground costs [Fan et al., 2023, Asadulaev et al., 2023, Cuturi et al., 2023, Uscidda and Cuturi, 2023], however it remains unclear how such a cost should be chosen.

Cost Learning. Ambiguity surrounding the choice of ground cost has motivated recent interest in learning adapted cost functions by leveraging assumed additional information about the transport map (see Appendix B.2 for a discussion

of related work). A frequently studied setting is inverse OT, which aims to learn cost functions from paired samples assumed to come from a transport coupling. Access to completely paired datasets is however unlikely to arise in practical applications. Instead, we may aim to learn an improved cost function from partial information about the mapping, such as a limited number of paired points or known structure in the map displacements. This motivates a need for general cost learning frameworks that are able to learn from a variety of types of available information.

One general approach for learning cost functions is a bi-level approach entailing solving a regularized OT problem for given cost function with Sinkhorn's algorithm, then differentiating through this solution to optimize the cost (see Appendix B.2.2). In many OT applications however, one wishes to approximate the Monge map $T^\star$ itself, rather than the cost function alone. This can be used to transport out-of-sample points, enabling applications such as generative modelling and prediction [Bunne et al., 2023]. While a learned cost can be plugged-in to an OT map estimator, if an accurate approximation is not obtained then the resulting estimator may not agree with the information used to learn the cost.

Contributions. We introduce an alternative approach for learning adapted cost functions and OT maps by instead *optimizing a cost-parameterized Monge map estimator directly* to be consistent with known information. This approach *simultaneously* learns a map estimator *and* a corresponding cost function, ensuring that the resulting map has the desired properties. We make the following contributions:

- We propose a differentiable, structured Monge map estimator, incorporating flexibly parameterized cost functions c, and provide a multi-level optimization procedure involving the differentiable entropic map estimator [Pooladian and Niles-Weed, 2021, Cuturi et al., 2023] to facilitate gradient based training.

- We first explore costs of the form $c(x, y) = h(x - y)$ for strictly convex h. We parameterize h using an Input Convex Neural Network (ICNN) [Amos et al., 2017] h_θ. As far as we are aware this is the first such use of ICNNs, which are more typically used to parameterize **potentials** [Makkuva et al., 2020] **rather than costs**.

- We then extend our construction to costs $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$ for invertible functions $\Phi_.$, providing additional flexibility whilst retaining a structured Monge map.

- We showcase how our approach can incorporate partially known information, including aligning mapping estimators with known data associations, and encouraging desirable properties in the resulting transport map.

2. Background on Optimal Transport

Monge. The original OT formulation given by Monge [1781] seeks a map $T^\star$ minimizing the total transportation cost amongst maps $T : \mathbb{R}^d \to \mathbb{R}^d$ pushing μ onto ν:

$$\min_{T:T\#\mu=\nu} \int_{\mathbb{R}^d} c(x, T(x))\mathrm{d}\mu(x). \quad (1)$$

Kantorovich. A more general formulation, permitting mass splitting, by Kantorovich [1942] seeks a joint distribution $\pi^\star \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ minimizing the cost over the set of couplings $\Gamma(\mu, \nu)$ between marginals μ and ν:

$$\min_{\pi\in\Gamma(\mu,\nu)} \iint_{\mathbb{R}^d\times\mathbb{R}^d} c(x, y)\mathrm{d}\pi(x, y). \quad (2)$$

The Kantorovich formulation admits the following dual formulation, where $\mathcal{R}_c = \{(f, g) \in L^1(\mu) \times L^1(\nu) : f(x) + g(y) \leq c(x, y)\ \mu \otimes \nu\ \text{a.e.}\}$. The solutions $(f^\star, g^\star)$ to the dual problem are known as the *Kantorovich potentials*,

$$\sup_{(f,g)\in\mathcal{R}_c} \left\{ \int f d\mu + \int g d\nu \right\}. \quad (3)$$

Closed-form Monge maps. The following theorem details sufficient conditions for equivalence of the Monge and Kantorovich problems, and motivates solving for the Kantorovich potentials to obtain the Monge mapping.

Theorem 2.1 (Theorem 1.17, Santambrogio [2015]). *For measures μ and ν on a compact domain $\Omega \subset \mathbb{R}^d$ and a cost of the form $c(x, y) = h(x - y)$ for a strictly convex function h, there exists an optimal plan $\pi^\star$ for the Kantorovich problem. If μ is absolutely continuous and $\partial\Omega$ is negligible, then this optimal plan is unique and of the form $(Id, T^\star)\#\mu$, there exists a Kantorovich potential $f^\star$, and*

$$T^\star(x) = x - (\nabla h)^{-1}(\nabla f^\star(x)). \quad (4)$$

Entropic OT. The entropic OT problem instead smooths the transport plan by adding an entropic penalty term to (2),

$$\pi^\star = \arg\min_{\pi\in\Gamma(\mu,\nu)} \iint_{\mathbb{R}^d\times\mathbb{R}^d} c(x, y)\mathrm{d}\pi(x, y) + \varepsilon KL(\pi|\mu \otimes \nu).$$

This relaxes the constraints on the dual potentials. The solutions $(f_\varepsilon^\star, g_\varepsilon^\star)$ to the dual problem are known as the *entropic potentials*,

$$\sup_{\substack{f\in\mathcal{C}(\mathcal{X}) \\ g\in\mathcal{C}(\mathcal{Y})}} \left\{ \int f\mathrm{d}\mu + \int g\mathrm{d}\nu + R(f, g) \right\}, \quad (5)$$

where $R(f, g) := -\varepsilon \iint e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} \mathrm{d}\mu(x)\mathrm{d}\nu(y)$.

Entropic OT approaches and hence approximates OT in the limit as $\varepsilon \searrow 0$, but the entropic solution enjoys the key benefit of differentiability with respect to the inputs. It also enables efficient computation for discrete measures $\hat\mu, \hat\nu$ using Sinkhorn's algorithm [Cuturi, 2013] to solve for the Sinkhorn potentials $(\mathbf{f}_\varepsilon^\star, \mathbf{g}_\varepsilon^\star)$ (see Appendix B.1.2).

3. A Differentiable Monge Map Estimator

We consider learning costs of the form $c(x, y) = h_\theta(x - y)$ for a strictly convex h_θ, for which the Monge map has the form in (4). This enables direct optimization according to conditions on the map itself, ensuring that the resulting mapping is consistent with known prior information.

Such costs allow the use of methods such as the entropic mapping estimator [Pooladian and Niles-Weed, 2021, Cuturi et al., 2023] and c-rectified flow [Liu, 2022], ensuring reliable estimation of the corresponding OT map. In contrast, learning OT map estimators for arbitrary costs requires a trade-off between distribution fitting and optimality (see Appendix B.1.1), and if the resulting mapping is inaccurate it may not be consistent with the information from which the cost was learned. Moreover, additional known information about an OT map will not uniquely determine a cost function. By considering convex costs, we provide an inductive bias towards simpler and more interpretable costs.

3.1. Differentiable, Cost-Parameterized Transport Maps

We parameterize h_θ using an Input Convex Neural Network (ICNN) [Amos et al., 2017]. We also enforce α-strong convexity and (optionally) symmetry of h_θ (see Appendix A.1). We use the entropic potential $\hat{f}_\theta^\varepsilon$ [Pooladian and Niles-Weed, 2021, Cuturi et al., 2023] for the discrete empirical measures $\hat{\mu}, \hat{\nu}$ as a differentiable proxy for the Kantorovich potential $f^\star$. This is constructed from the Sinkhorn potential $\mathbf{g}_\varepsilon^\star$ solving the discrete OT problem between $\hat{\mu}, \hat{\nu}$:

$$\hat{f}_\theta^\varepsilon(x) = -\varepsilon \log \sum_j \exp\left(\frac{(\mathbf{g}_\varepsilon^\star)_j - h_\theta(x - y_j)}{\varepsilon} \right). \quad (6)$$

The resulting estimator is $T_\theta^\varepsilon(x) = x - (\nabla h_\theta)^{-1}(\nabla \hat{f}_\theta^\varepsilon(x))$.

Differentiability. We can differentiate through the output of Sinkhorn's algorithm using implicit differentiation [Luise et al., 2018], or by unrolling the iterates [Adams and Zemel, 2011, Flamary et al., 2018]. We also note the following well-known relation, enabling differentiation through $(\nabla h_\theta)^{-1}$.

Proposition 3.1. *For a smooth, strictly convex function h with domain $\mathbb{R}^d$, we have*

$$(\nabla h)^{-1}(x) = \arg\min_z \left\{ h(z) - \langle z, x \rangle \right\}. \quad (7)$$

We use the CELU activation function [Barron, 2017] to ensure smoothness of our ICNN parameterization h_θ. As we enforce α-strong convexity of h, strict convexity is also satisfied, and the minimization in (7) can be solved efficiently using numerical methods. We can then use implicit differentiation (see Chapter 10, Blondel and Roulet [2024]) to differentiate through this inner minimization with respect to the cost parameters. We implement this using the JAXopt libary [Blondel et al., 2022].

ICNNs have previously been used in OT methods to parameterize Kantorovich potentials [Makkuva et al., 2020, Korotin et al., 2021a]. To our knowledge, using an ICNN to parameterize the cost function is novel, as is the construction of a Monge map estimator that is end-to-end differentiable with respect to its cost function parameters.

Choice of loss function. The training procedure is described in Algorithm 1 in Appendix A. By choosing an appropriate loss function $\mathcal{L}(\theta)$, we can incorporate desired explicit biases into the learned cost through map estimator T_θ^ε. For example, given a known subset of paired points $(x_i, y_i)_{i=1}^N$, a natural choice of training loss is $\mathcal{L}(\theta) := \frac{1}{N} \sum_{i=1}^N \|T_\theta^\varepsilon(x_i) - y_i\|_2^2$ to encourage the learned map to respect the known pairs. We provide other choices of loss functions suitable for a range of problems in Appendix A.2. We can also construct the reverse map estimator $\left(T_\theta^\varepsilon(x)\right)^{-1}$. We find it beneficial to train using both mappings jointly; see Appendix A.2.3 for details.

Warmstarting the inner optimizations. Each training iteration entails two inner optimization problems, (1) the discrete entropy-regularized OT problem using Sinkhorn, and (2) the evaluation of $(\nabla h)^{-1}$. We warmstart each optimization from the corresponding solutions at the previous iteration, which significantly improves training speed [Amos, 2022].

3.2. Augmenting with Diffeomorphisms

In the presence of extensive information about the map, the choice of strictly convex costs $c(x, y) = h(x - y)$ can be overly restrictive, as there may not be a convex cost that allows the Monge map to be consistent with the known information. We therefore extend our framework by first transforming the marginal measures using diffeomorphisms Φ_μ, Φ_ν, then applying our method to $\tilde{\mu} = \Phi_\mu \# \mu$ and $\tilde{\nu} = \Phi_\nu \# \nu$. The following extension of Theorem 2.1, proved in Appendix E, shows that this amounts to learning an OT map with cost $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$.

Theorem 3.2. *Under the conditions of Theorem 2.1 with a cost of the form $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$ for strictly convex h and diffeomorphisms Φ_μ, Φ_ν, the optimal plan is unique, of the form $(Id, T^\star)\#\mu$, where $T^\star$ can be written:*

$$T^\star(x) = \Phi_\nu^{-1}\left[\Phi_\mu(x) - (\nabla h)^{-1} \circ \nabla f^\star \circ \Phi_\mu(x) \right]. \quad (8)$$

We parameterize Φ_μ, Φ_ν using normalizing flows [Rezende and Mohamed, 2015], and train end-to-end along with the procedure in Algorithm 1. This increases the expressivity of the class of cost functions we consider. As we would prefer to learn simpler costs, we can take $\Phi_\mu = \Phi_\nu$ which preserves symmetry when h is symmetric. Theorem 3.2 resembles similar recent results which use learnable invertible matrices [Klein et al., 2024, Prop 3.] and fixed mirror-maps [Rankin and Wong, 2023].
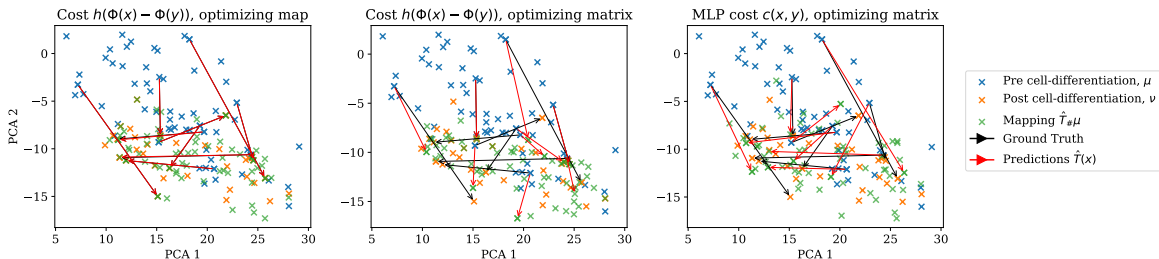
Figure 2. (left) By optimizing the map directly, we learn a Monge map estimator which agrees with the Live-seq trajectories. *(middle)* Optimizing according to the coupling matrix with a cost $h(\Phi(x) - \Phi(y))$ transports some points correctly, but struggles to learn a good cost overall (see Table 1). *(right)* For a general MLP cost, it is difficult to obtain a good mapping estimator.

4. Experiments

Inverse OT. In Figure 1 we verify that our method can indeed learn valid cost functions in the Inverse OT setting using synthetic $2d$ distributions. We are able to learn a consistent map estimator for the T-shape dataset using only a symmetric convex cost $c(x, y) = h(x - y)$. For the moon and spiral datasets, we require costs of form $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$. Solving the discrete OT problem on the training datasets using the learned cost recovers the correct pairs, demonstrating that the learned costs are indeed valid.

Limited labelled pairs. Optimal transport is commonly used to infer single-cell transcriptome trajectories between unaligned datasets, under the assumption that a cell population has moved 'efficiently' between the observed timesteps [Schiebinger et al., 2019, Bunne et al., 2023]. The recent Live-seq profiling technique [Chen et al., 2022] avoids the destruction of measured cells, enabling a number of individual cell trajectories to be traced. These trajectories appear not to agree with those predicted by OT methods that use the squared-Euclidean cost. Nevertheless, the assumption that cells move 'efficiently' is reasonable, raising the question of whether a different choice of ground-cost is more suitable.

To learn an expressive but interpretable cost, we consider learning symmetric costs of the form $h(\Phi(x) - \Phi(y))$. We fit the Monge map estimator to the first 10 principal components of the Live-seq data consisting of cells pre- and post cell-differentiation, so that it agrees with the known trajectories. We plot the resulting mapping estimators in Figure 2. In Table 1 we report the total incorrectly transported mass in the resulting coupling (denoted I.M.; see Appendix C), as a measure of the validity of the learned cost. To assess the adapted OT map, we also report the RMSE of the predictions for the known cell trajectories, and the Sinkhorn divergence $S_\varepsilon(\hat{T}\#\mu, \nu)$. Experimental details are provided in Appendix C. We compare against costs learned by optimizing the Sinkhorn coupling matrix to minimize the incorrect mass reported in Table 1. We first use the same parameterization $h(\Phi(x) - \Phi(y))$ and obtain a map using the entropic mapping estimator. We also use an unstructured MLP cost $c(x, y)$ [Liu et al., 2020] and learn a

Table 1. Results from the Live-seq experiment, over 10 initialization seeds. Lower is better. Unsurprisingly, the MLP cost is able to place most mass along the correct pairings. The resulting cost is however unstructured and difficult to interpret, and it is difficult to approximate the corresponding OT map. For costs $h(\Phi(x) - \Phi(y))$, the resulting entropic mappings show an improved fit to the target, but optimizing the map learns better costs and has significantly better alignment with the ground truth.

| | OPTIMIZING MAP | OPTIMIZING MATRIX | |
	$h(\Phi(x) - \Phi(y))$	$h(\Phi(x) - \Phi(y))$	MLP $c(x, y)$
I.M.	0.095 ± 0.035	0.153 ± 0.012	0.051 ± 0.000
RMSE	2.02 ± 2.21	16.9 ± 1.6	6.91 ± 2.52
$S_\varepsilon(\hat{T}\#\mu, \nu)$	7.73 ± 3.02	4.94 ± 0.85	19.8 ± 2.8

map estimator using the Monge Gap regularizer [Uscidda and Cuturi, 2023]. Optimizing the map directly consistently obtained both an interpretable cost function and a good transport estimator agreeing with the observed trajectories.

Given the scarcity of known Live-seq trajectories, we cannot test out-of-sample performance of the resulting map estimator. We therefore perform the same procedure on synthetic data with different numbers of labelled pairs in Appendix D.1, reporting results on training and newly-sampled data.

Additional Experiments. We provide additional experiments in Appendix D, including the ability to induce desirable properties that hold on the map displacements themselves, such as being low-rank or k-directional.

5. Conclusion

We have introduced a Monge map estimator parameterized by a learnable cost function, which is differentiable with respect to its parameters. This enables learning adapted cost functions and OT map estimators through gradient based training so that the map is consistent with known prior information. We have demonstrated the ability to learn costs by aligning with paired data samples, and by inducing desirable properties on the Monge map estimator itself. Future directions include: conducting further comparisons of optimizing the entropic map compared to the coupling matrix, and investigating the use of adapted costs and OT maps in downstream tasks.

Acknowledgements

References

Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011.

Brandon Amos. Tutorial on amortized optimization. *Found. Trends Mach. Learn.*, 16:592–732, 2022.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Francisco Andrade, Gabriel Peyré, and Clarice Poon. Sparsistency for inverse optimal transport. In *The Twelfth International Conference on Learning Representations*, 2024.

Arip Asadulaev, Alexander Korotin, Vage Egiazarian, Petr Mokrov, and Evgeny Burnaev. Neural optimal transport with general cost functionals, 2023.

Jonathan T. Barron. Continuously differentiable exponential linear units, 2017.

Mathieu Blondel and Vincent Roulet. The elements of differentiable programming, 2024.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-Lopez, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, 2022.

Y. Brenier. Decomposition polaire et rearrangement monotone des champs de vecteurs. *C. R. Acad. Sci. Paris Ser. I Math.*, 305, 1987.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, 1993.

Charlotte Bunne, Stefan G. Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. volume 20 of *Nature Methods*, pages 1759–1768, 2023.

Guillaume Carlier, Arnaud Dupuy, Alfred Galichon, and Yifei Sun. SISTA: Learning optimal transport costs under sparsity constraints. *Communications on Pure and Applied Mathematics*, 76, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33: 9912–9924, 2020.

Wanze Chen, Orane Guillaume-Gentil, Pernille Yde Rainer, Christoph G. Gäbelein, Wouter Saelens, Vincent Gardeux, Amanda Klaeger, Riccardo Dainese, Magda Zachara, Tomaso Zambelli, Julia A. Vorholt, and Bart Deplancke. Live-seq enables temporal transcriptomic recording of single cells. volume 608 of *Nature*, pages 733–740, 2022.

Wei-Ting Chiu, Pei Wang, and Patrick Shafto. Discrete probabilistic inverse optimal transport. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pages 2100–2111. PMLR, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, 2013.

Marco Cuturi and David Avis. Ground metric learning. *J. Mach. Learn. Res.*, 15(1), 2014.

Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in neural information processing systems*, 32, 2019.

Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal Transport Tools (OTT): A JAX toolbox for all things Wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

Marco Cuturi, Michal Klein, and Pierre Ablin. Monge, Bregman and Occam: Interpretable optimal transport in high-dimensions with feature-sparse maps. In *International Conference on Machine Learning*, 2023.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34, 2021.

Arnaud Dupuy and Alfred Galichon. Personality traits and the marriage market. *Journal of Political Economy*, 122, 2014.

Arnaud Dupuy, Alfred Galichon, and Yifei Sun. Estimating matching affinity matrices under low-rank constraints. *Information and Inference: A Journal of the IMA*, 8(4), 2019.

Jiaojiao Fan, Shu Liu, Shaojun Ma, Hao-Min Zhou, and Yongxin Chen. Neural Monge map estimation and its applications. *Transactions on Machine Learning Research*, 2023.

Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018.

Alfred Galichon and Bernard Salanié. Cupid's Invisible Hand: Social Surplus and Identification in Matching Models. *The Review of Economic Studies*, 89(5), 2021.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis R. Bach. Stochastic optimization for large-scale optimal transport. In *Neural Information Processing Systems*, 2016.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

Aude Genevay, Gabriel Dulac-Arnold, and Jean-Philippe Vert. Differentiable deep clustering with cluster size constraints. *arXiv preprint arXiv:1910.09036*, 2019.

Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi, and Gabriel Peyré. Ground metric learning on graphs. *Journal of Mathematical Imaging and Vision*, 63 (1), 2020.

Piotr Indyk, Jiří Matoušek Matoušek, and Anastasios Sidiropoulo. Low-distortion embeddings of finite metric spaces. *Handbook of discrete and computational geometry*, pages 211–231, 2017.

L.V. Kantorovich. *On translation of mass*. Proceedings of the USSR Academy of Sciences, 37, 1942.

Michal Klein, Aram-Alexandre Pooladian, Pierre Ablin, Eugène Ndiaye, Jonathan Niles-Weed, and Marco Cuturi. Learning elastic costs to shape Monge displacements, 2024.

Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2021a.

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous Wasserstein-2 benchmark. 2021b.

Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. *J. Mach. Learn. Res.*, 20(1), 2019.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022.

Ruishan Liu, Akshay Balsubramani, and James Zou. Learning transport cost from subset correspondence. In *International Conference on Learning Representations*, 2020.

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with Wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

Shaojun Ma, Haodong Sun, Xiaojing Ye, Hongyuan Zha, and Haomin Zhou. Learning cost functions for optimal transport, 2021.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences*, pages 666–704, 1781.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017.

François-Pierre Paty and Marco Cuturi. Regularized optimal transport is ground cost adversarial. In *International Conference on Machine Learning*, 2020.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11 (5-6):355–602, 2019.

Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An inductive bias for distances: Neural nets that respect the triangle inequality. In *International Conference on Learning Representations*, 2020.

Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps, 2021.

Aram-Alexandre Pooladian, Carles Domingo-Enrich, Ricky T. Q. Chen, and Brandon Amos. Neural optimal transport with Lagrangian costs. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.

Cale Rankin and Ting-Kam Leonard Wong. Bregman-Wasserstein divergence: geometry and applications. *arXiv preprint arXiv:2302.05833*, 2023.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, 1970.

F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Springer International Publishing, 2015.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4), 2019.

Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-Scale Optimal Transport and Mapping Estimation. In *International Conference on Learning Representations*, 2018.

Liangliang Shi, Gu Zhang, Haoyu Zhen, Jintao Fan, and Junchi Yan. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion Schrödinger bridges. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

Andrew M. Stuart and Marie-Therese Wolfram. Inverse optimal transport. *SIAM J. Appl. Math.*, 80, 2019.

Amirhossein Taghvaei and Amin Jalali. 2-Wasserstein approximation via restricted convex potentials with application to improved training for gans, 2019.

Théo Uscidda and Marco Cuturi. The Monge gap: A regularizer to learn all transport maps. In *International Conference on Machine Learning*, 2023.

Chris Waites. Jax-flows: Normalizing flows in jax, 2020.

Fan Wang and Leonidas J. Guibas. Supervised earth mover's distance learning and its computer vision applications. In *Computer Vision – ECCV 2012*, pages 442–455. Springer Berlin Heidelberg, 2012.

A. Method

Algorithm 1 Differentiable cost-parameterized entropic mapping estimator

---

Input: Empirical measures $\hat{\mu} = \sum_i \delta_{x_i}, \hat{\nu} = \sum_j \delta_{y_j}$,
Parameterized ICNN cost h_θ ;
Entropy regularization scaling value $\tilde{\varepsilon}$;
Loss function $\mathcal{L}(\theta)$;
Regularizing function $\mathcal{R}(\theta)$;
while not converged **do**

    1.   Apply Sinkhorn's algorithm between the empirical measures $\hat{\mu}, \hat{\nu}$ with entropy regularization value ε (the cost matrix mean scaled by $\tilde{\varepsilon}$), using the current parameterized function h_θ,

    2.   Use Sinkhorn output $\mathbf{g}_\varepsilon^\star$ to construct the entropic potential estimator

$$\hat{f}_\theta^\varepsilon(x) = -\varepsilon \log \sum_j \exp \left(\frac{(\mathbf{g}_\varepsilon^\star)_j - h_\theta(x - y_j)}{\varepsilon} \right), \tag{9}$$

    3.   Construct the entropic mapping estimator

$$T_\theta^\varepsilon(x) = x - (\nabla h_\theta)^{-1} \circ (\nabla \hat{f}_\theta^\varepsilon)(x), \tag{10}$$

    4.   Update θ by gradient descent according to a loss function $\mathcal{L}(\theta) + \mathcal{R}(\theta)$ by differentiating through Sinkhorn and $(\nabla h)^{-1}$ using implicit differentiation.

end while

We implement the differentiation through Sinkhorn using the OTT-JAX library [Cuturi et al., 2022]. We use the L-BFGS solver [Liu and Nocedal, 1989] from the JAXopt library [Blondel et al., 2022] to evaluate and implicitly differentiate through the inner minimization when evaluating $(\nabla h_\theta)^{-1}$.

A.1. Cost function parameterization

A.1.1. CONVEX FUNCTION h

Input Convex Neural Networks. ICNNs [Amos et al., 2017] are a class of neural networks $f_\theta : \mathbb{R}^d \to \mathbb{R}$ with an architecture that ensures that the mapping $x \mapsto f_\theta(x)$ is convex. An ICNN consists of k feedforward layers, where for each layer $i = 0, ..., k - 1$ the activations are given by

$$z_{i+1} = \sigma_i(W_i^{(z)} z_i + W_i^{(x)} x + b_i), \qquad f_\theta(x) = z_k. \tag{11}$$

The network ensures convexity by consisting only of non-negative sums and compositions of convex non-decreasing functions with convex functions. It therefore requires the nonlinear activation functions to be convex and non-decreasing, and the weight matrices $W_i^{(z)}$ to be non-negative (no such requirements are required for the passthrough matrices $W_i^{(x)}$). The first layer does not require an additional passthrough layer, so we have $W_0^{(z)} = 0$.

ICNNs have previously been utilised in computational optimal transport methods to parameterize Kantorovich potentials, which via a reparameterization are known to be convex in the case of the squared-Euclidean cost [Makkuva et al., 2020, Korotin et al., 2021a]. Although mathematically elegant, Korotin et al. [2021b] note that constraining the potentials to be ICNNs can in fact result in worse performance versus vanilla MLPs. We remark that in our case, the use of ICNNs instead of MLPs to parameterize the cost is crucial to our methodology, as it ensures that the inner optimization in (7) is convex and can be solved numerically.

Symmetry. The additional information used to learn an adapted Monge map and cost will not uniquely determine the cost function. This is especially the case when there is little additional information. We therefore wish to favour learning simpler and more interpretable costs. Translation-invariant convex costs $c(x, y) = h(x - y)$ provide a good inductive bias towards such costs. We may also wish for the cost to be symmetric; this can be optionally enforced by parameterizing as $h(z) = \tilde{h}_\theta(z) + \tilde{h}_\theta(-z)$ for an ICNN $\tilde{h}_\theta$.

α**-strong convexity.** The closed-form of the Monge map in (4) requires strict convexity of h. We enforce this by adding a quadratic term $\alpha\|x - y\|_2^2$ to the cost function. This also ensures that the inner minimization (7) is strongly convex and thus the unique solution can be solved for efficiently with numerical methods.

Parameterizing the convex conjugate. We remark that after having learned an adapted transport map, evaluating the learned mapping at a point x requires a minimization problem to evaluate $(\nabla h)^{-1}$. This can be avoided by instead parameterizing the convex conjugate h^* rather than h, and using the relation $(\nabla h)^{-1} = \nabla h^*$. While avoiding the inner minimization when evaluating the map, this instead requires an inner minimization to evaluate each entry of the cost matrix $\mathbf{C}_{i,j} = c(x_i, y_j)$, which is then fed into Sinkhorn at each training iteration. The result is that significantly more evaluations of $(\nabla h)^{-1}$ are required during training, though with the benefit that the resulting learned map can be evaluated directly. The choice to parameterize h or h^* should therefore depend on whether fast evaluation of the learned mapping is preferred over training speed.

A.1.2. DIFFEOMORPHISMS Φ_μ, Φ_ν

In Section 3.2, we extend our framework to incorporate cost functions of the form $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$. In our experiments, we use the MADE flow architecture [Papamakarios et al., 2017] to parameterize Φ_μ, Φ_ν, implemented using the Jax-Flows library [Waites, 2020].

Relation to metric learning. When taking $\Phi_\mu = \Phi_\nu$ the cost $c(x, y) = h(\Phi(x) - \Phi(y))$ resembles a Siamese network [Bromley et al., 1993], a common approach in the metric learning literature. Siamese networks seek to learn encoders Φ so that the resulting function $d(x, y) = \frac{1}{2}\|\Phi(x) - \Phi(y)\|_2^2$ is an improved notion of distance for the data. However, Indyk et al. [2017] note that such compositions can be insufficiently expressive to precisely model certain metrics. Motivated by this, Pitis et al. [2020] replace the squared-Euclidean distance with a modified ICNN, providing a more expressive class of neural distances.

Our motivation is similar. Note that we could use an ICNN ψ to directly parameterize a Monge map for a cost $c(x, y) = \frac{1}{2}\|\Phi_\mu(x) - \Phi_\nu(y)\|_2^2$ according to the composition $\Phi_\nu^{-1} \circ \nabla\psi \circ \Phi_\mu$ (via Brenier's theorem and a change of variable as in Theorem 3.2). However, by incorporating an ICNN we add flexibility in the cost parameterization, allowing us to favour simpler and more interpretable learned costs.

A.2. Choice of Loss Function

To illustrate the generality of our method, we provide here potential choices of loss function suitable for different applications. They encourage the map estimator to be consistent with known information or desired behaviour. We remark that these suggestions are only examples of possible loss functions that can be used, and a practitioner is free to choose any differentiable loss function suitable for their needs. In particular, our method is most suitable when behaviour is desired to hold on the map estimator itself, as this is the object being optimized.

A.2.1. LABELLED DATAPOINTS

Paired datapoints. Consider assuming access to a subset $\{(x_i, y_i)\}_{i=1}^N$ of $\hat{\mu}, \hat{\nu}$ consisting of known pairings from the optimal mapping, so $y_i = T^\star(x_i)$. We can optimize our map to agree with the observed pairings by penalizing the L_2 distance between the predictions and the known targets,

$$\mathcal{L}(\theta) := \frac{1}{N}\sum_{i=1}^N \|T_\theta^\varepsilon(x_i) - y_i\|_2^2.$$

In the case where all samples in $\hat{\mu}, \hat{\nu}$ are paired, this recovers the Inverse OT setting. Note that during training, we only need to evaluate T_θ^ε at the paired data points.

Subset-to-subset correspondence. Liu et al. [2020] consider a setting in which certain source subsets are known to map to certain target subsets. If our empirical source and target distributions are $\mu = \bigcup_i \mu_i, \nu = \bigcup_i \nu_i$ with μ_i known to map to ν_i, then the loss function can be chosen as

$$\mathcal{L}(\theta) = \sum_i S_{\tilde{\varepsilon}}(T_\theta^\varepsilon \# \hat{\mu}_i, \hat{\nu}_i),$$

where $S_{\tilde{\varepsilon}}$ denotes the Sinkhorn divergence [Genevay et al., 2018] with regularization parameter $\tilde{\varepsilon}$.

A.2.2. PROPERTIES OF MAP DISPLACEMENTS

Low-rank displacements. To encourage the map estimator T_θ^ε to transport along a lower p-dimensional subspace, we can penalize the magnitude of the trailing singular values $\sigma_i^{(\theta)}$ of the displacement matrix $(T_\theta^\varepsilon(x_i) - x_i)_i$,

$$\mathcal{L}(\theta) = \sum_{i>p} \|\sigma_i^{(\theta)}\|^2.$$

k**-directional displacements.** We can encourage the displacements to lie primarily along at most k distinct directions. We can parameterize k directions $v^\phi = (v_1^\phi, ..., v_k^\phi)$, then maximize the cumulative smoothed minimum of the cosine distance $d_{\cos}(u, v) = 1 - \frac{u \cdot v}{\|u\|\|v\|}$,

$$\mathcal{L}(\theta, \phi) = -\sum_i \text{LogSumExp}_j \Big[-d_{\cos}\big(T_\theta^\varepsilon(x_i) - x_i, v_j^\phi\big) \Big].$$

A.2.3. REVERSE MAP ESTIMATORS

We have presented our method as optimizing according to a loss on the forward mapping estimator $\mathcal{L}(\theta) = \mathcal{L}(T_\theta^\varepsilon)$. However, we can also construct the reverse entropic mapping estimator as

$$\hat{g}_\theta^\varepsilon(y) = -\varepsilon \log \sum_i \exp \left(\frac{(\mathbf{f}_\varepsilon^\star)_i - h_\theta(x_i - y)}{\varepsilon} \right), \qquad (T_\theta^\varepsilon)^{-1}(y) = y - (\nabla \tilde{h}_\theta)^{-1} \circ (\nabla \hat{g}_\theta^\varepsilon)(y),$$

where $\tilde{h}_\theta(z) = h_\theta(-z)$. The reverse map estimator should also be consistent with the known information, so we optimize according to the sum of the losses of the forward and backwards mapping estimators,

$$\mathcal{L}(\theta) = \frac{1}{2}\mathcal{L}(T_\theta^\varepsilon) + \frac{1}{2}\mathcal{L}\big((T_\theta^\varepsilon)^{-1}\big).$$

A.3. Regularization

To ensure stable training, we add a regularization term $\mathcal{R}(\theta)$ to the loss used during training.

Cost matrix. During optimization, the procedure may learn a cost function which places very large absolute values on certain displacements in order to encourage the desired behaviour. Although the OT map is invariant to scaling of the cost function, if such values become extreme this can result in numerical instability. To prevent this, we add a regularization term penalizing the softmax of the largest absolute value in the cost matrix $\mathbf{C}_\theta$,

$$\lambda_{max} \text{LogSumExp}_{i,j} \big(|c_\theta(x_i, y_j)|\big)^2.$$

The value of λ_{max} can be taken to be very small, so that it has minimal effect on optimization apart from preventing extreme values.

Flows. Recall that often we prefer to learn symmetric cost functions. However, when there is a large amount of available information about the transport map, it might not be possible to fit such information with a symmetric cost. In such cases, we can use two separate flows, and we can encourage the cost towards symmetricity by adding a regularizing function so that $\Phi_\mu \approx \Phi_\nu$,

$$\sum_i \|\Phi_\mu(x_i) - \Phi_\nu(x_i)\|^2 + \sum_j \|\Phi_\mu(y_j) - \Phi_\nu(y_j)\|^2.$$

As we wish to prefer simpler cost functions, we can control the complexity of the flow by regularizing using the Dirichlet energy,

$$\frac{1}{2}\sum_i \|\nabla\Phi_\mu(x_i)\|_2^2 + \frac{1}{2}\sum_j \|\nabla\Phi_\nu(y_j)\|_2^2.$$

B. Related Work

B.1. Monge Map Estimation

Often in applications it is desired to construct an estimator for the Monge map itself. This motivates our approach of optimizing a Monge map estimator directly, to ensure that the resulting mapping agrees with known information. We here provide an overview of methods to construct a Monge map estimator.

B.1.1. NEURAL OT MAP ESTIMATION

Many approaches utilise neural networks to parameterize the Kantorovich potentials, or alternatively the transport map itself.

Squared-Euclidean cost. In the case of the squared-Euclidean cost $c(x, y) = \frac{1}{2}\|x - y\|^2$, the closed form expression for the Monge map given in Theorem 2.1 reduces to the celebrated Brenier's theorem, $T^\star(x) = x - \nabla f^\star(x)$ [Brenier, 1987]. In this case, it is common to reparameterize the potentials by letting $\psi = \frac{1}{2}\|x\|^2 - f(x)$ and $\varphi(y) = \frac{1}{2}\|y\|^2 - g(y)$. The Kantorovich dual problem (3) becomes

$$\inf_{(\psi, \varphi) \in \tilde{\Phi}} \left\{ \int \psi d\mu + \int \varphi d\nu \right\}, \tag{12}$$

where $\tilde{\Phi} = \{(\psi, \varphi) \in L^1(\mu) \times L^1(\nu) : \psi(x) + \varphi(y) \geq \langle x, y \rangle \ \mu \otimes \nu \text{ a.e.}\}$. There exists an optimal potential $\psi^\star$ that is convex [Santambrogio, 2015], and the Monge map is given by $T^\star(x) = \nabla \psi^\star(x)$.

The dual objective (12) and the link to convex analysis form the basis for many successful computational techniques, though as a result they are restricted to only the squared-Euclidean cost. Taghvaei and Jalali [2019], Makkuva et al. [2020], Korotin et al. [2021a] parameterize the dual potentials in (12) using ICNNs. Taghvaei and Jalali [2019] solve for the convex conjugate ψ^* as an inner minimization step, and subsequent work [Makkuva et al., 2020] instead parameterize both potentials as ICNNs and optimize a minimax objective. Korotin et al. [2021a] remove the minimax objective by adding a cycle-consistency regularizer to encourage the potentials to be convex-conjugate up to a constant. While the use of ICNNs to parameterize the potentials is appealing from a mathematical standpoint, results in Korotin et al. [2021b] suggest that it may hinder optimization.

Although the squared-Euclidean cost has desirable theoretical properties, it may not be an appropriate choice in applications. Somnath et al. [2023] comment that methods approximating the squared-Euclidean OT map can result in erroneous matchings when ground truth couplings are known (demonstrated in Figure 1), and Genevay et al. [2018] have shown it to have poor discriminative properties for image data. This motivates learning an improved cost function more suitable for the problem at hand.

General Cost Functions. Recently, several alternative methods have been proposed that are suitable for general cost functions. Fan et al. [2023] and Asadulaev et al. [2023] optimize a saddle point formulation of the OT problem, directly parameterizing the transport map and a dual potential using neural networks. Uscidda and Cuturi [2023] take an alternative approach, instead parameterizing the mapping with a neural network and training so that it fits to the source measure, with a *Monge Gap* regularizer that encourages the map to be optimal with respect to the chosen cost. Such methods present a trade-off between fitting to the distribution, and optimality with respect to the chosen cost, both of which need to be minimized if the resulting map is to be an accurate approximation to the OT map.

B.1.2. ENTROPIC MAP ESTIMATION

Entropic OT. The entropy regularized OT problem adds a entropic penalty term to the primal objective,

$$\pi_\varepsilon^\star = \arg\inf_{\pi \in \Gamma(\mu, \nu)} \iint_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y) \mathrm{d}\pi(x, y) + \varepsilon KL(\pi | \mu \otimes \nu). \tag{13}$$

Note that only the support of the reference measure $\mu \otimes \nu$ affects the optimization problem [Peyré and Cuturi, 2019]. The entropy regularization 'blurs' the solution, enabling differentiability of both the entropy-regularized OT cost and plan with respect to the input measures and cost function. As a result, optimal transport has enjoyed success in machine learning applications for which differentiability is required.

The dual formulation of the unregularized Kantorovich problem (3) has strict constraints on the potentials, leading to a difficult optimization problem. The entropic-regularized dual formulation relaxes these constraints, making it more amenable

to optimization,

$$\sup_{\substack{f \in \mathcal{C}(\mathcal{X}) \\ g \in \mathcal{C}(\mathcal{Y})}} \left\{ \int f \mathrm{d}\mu + \int g \mathrm{d}\nu - \varepsilon \iint e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} \mathrm{d}\mu(x)\mathrm{d}\nu(y) \right\}.$$

Stochastic dual optimization. Genevay et al. [2016] optimize this dual objective with stochastic gradient descent using samples from the measures. In the case of continuous measures, they parameterize the potentials as kernel expansions in a Reproducing Kernel Hilbert Space. Seguy et al. [2018] instead propose using neural networks to parameterize the potentials, and also construct a deterministic map using a neural network to approximate the resulting barycentric projection. In the case of the squared-Euclidean cost, this barycentric projection corresponds to the Monge map.

Sinkhorn. In practice, we often have access to discrete empirical approximations to underlying continuous distributions. In the discrete case with measures $\tilde{\mu} = \sum_i a_i \delta_{x_i}, \tilde{\nu} = \sum_j b_j \delta_{y_j}$ and cost matrix $\mathbf{C}_{i,j} = c(x_i, y_j)$, the entropy-regularized OT problem can be written as

$$\Pi_{\varepsilon}^{\star} = \min_{\Pi \in U(\mathbf{a},\mathbf{b})} \langle \Pi, \mathbf{C} \rangle - \varepsilon H(\Pi), \qquad H(\Pi) = -\sum_{i,j} \Pi_{i,j}(\log \Pi_{i,j} - 1). \tag{14}$$

The Sinkhorn algorithm [Cuturi, 2013] provides an efficient way to compute the dual variables of the discrete entropy-regularized OT problem, and is also well-suited for parallelization on GPUs to solve multiple OT problems simultaneously. From an arbitary initialization $\mathbf{g}^{(0)}$ and using the kernel matrix $\mathbf{K}_{i,j} = \exp\left(\frac{-\mathbf{C}_{i,j}}{\varepsilon}\right)$, the Sinkhorn iterates are defined as

$$\mathbf{f}^{(\ell+1)} = \varepsilon \log \mathbf{a} - \varepsilon \log\left(\mathbf{K}e^{\mathbf{g}^{(\ell)}/\varepsilon}\right), \qquad \mathbf{g}^{(\ell+1)} = \varepsilon \log \mathbf{b} - \varepsilon \log\left(\mathbf{K}^{\top}e^{\mathbf{f}^{(\ell+1)}/\varepsilon}\right). \tag{15}$$

These iterates converge to the Sinkhorn potentials $(\mathbf{f}_{\varepsilon}^{\star}, \mathbf{g}_{\varepsilon}^{\star})$. The optimal transport plan $\Pi_{\varepsilon}^{\star}$ can then be recovered from the potentials as $\Pi_{\varepsilon}^{\star} = \mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v})$, where $\mathbf{u} = \exp(\mathbf{f}_{\varepsilon}^{\star}/\varepsilon), \mathbf{v} = \exp(\mathbf{g}_{\varepsilon}^{\star}/\varepsilon)$.

Entropic mapping estimator. Recall that for a cost function $c(x,y) = h(x-y)$ for strictly convex h, the Monge map has form

$$T^{\star}(x) = x - (\nabla h)^{-1}(\nabla f^{\star}(x)),$$

where $f^{\star}$ is the Kantorovich potential. The Kantorovich potentials can be chosen to satisfy the following h-conjugacy property [Santambrogio, 2015], which is unfortunately a difficult property to enforce,

$$f^{\star}(x) = \min_y \left\{ h(x-y) - g^{\star}(y) \right\} \qquad g^{\star}(y) = \min_x \left\{ h(x-y) - f^{\star}(x) \right\}.$$

In the entropically regularized case, the entropic potentials $f_{\varepsilon}^{\star}, g_{\varepsilon}^{\star}$ instead satisfy the following relation, which is a softmin relaxation of the above h-conjugacy property,

$$f_{\varepsilon}^{\star}(x) = -\varepsilon \log \int e^{\frac{g_{\varepsilon}^{\star}(y)-h(x-y)}{\varepsilon}} d\nu(y), \qquad g_{\varepsilon}^{\star}(y) = -\varepsilon \log \int e^{\frac{f_{\varepsilon}^{\star}(x)-h(x-y)}{\varepsilon}} d\mu(x).$$

This suggests using the entropic potentials in place of the true Kantorovich potential. In practice, we only have access to discrete empirical measures $\hat{\mu}, \hat{\nu}$ when constructing a Monge map estimator. The entropic mapping estimator therefore solves the discrete entropic OT problem between $\hat{\mu}, \hat{\nu}$ using Sinkhorn, and uses the output to construct the entropic potential $\hat{f}_{\varepsilon}$ which is used in place of f^{*} in the Monge map expression. The resulting entropic map estimator is

$$T_h^{\varepsilon}(x) = x - (\nabla h)^{-1}(\nabla \hat{f}_{\varepsilon}(x)).$$

This estimator was proposed in Pooladian and Niles-Weed [2021] for the squared-Euclidean cost, along with finite-sample guarantees on its performance. It was extended to general convex functions h in Cuturi et al. [2023], and Klein et al. [2024] demonstrate the versatility and good performance of the resulting estimator.

As it is constructed from the Sinkhorn iterations, the entropic mapping estimator can be computed efficiently in comparison to the alternative approaches outlined above. As we can differentiate through the Sinkhorn iterations, it is therefore suitable for our aim of constructing a differentiable Monge map estimator. Our additional contributions lie in the use of ICNNs to parameterize h, and in the observation that the evaluation of $(\nabla h)^{-1}$ can be differentiated through using implicit differentiation, enabling end-to-end differentiability of the estimator with respect to the cost function parameters.

B.2. Cost Learning

Methods to learn an adapted cost functions have attracted attention since the problem was introduced by Cuturi and Avis [2014], but remain relatively unexplored in comparison to standard OT. Existing methods in the literature consider a variety of different problem settings.

B.2.1. INVERSE OPTIMAL TRANSPORT

Inverse optimal transport (iOT) aims to learn a cost function given paired samples (x_i, y_i) from the optimal or entropic coupling.

Optimizing a dual objective. Existing iOT approaches typically assume access to samples from the entropic coupling $(x_i, y_i) \sim \pi_\varepsilon$, which form an empirical joint measure $\hat{\pi}_n = \frac{1}{n} \sum_i \delta_{(x_i, y_i)}$. They then perform maximum likelihood estimation for a parameterized cost c_θ by minimizing the negative log-likelihood given by the convex function

$$\ell(\theta) = -\langle c_\theta, \hat{\pi}_n \rangle + \sup_{\pi \in \mathcal{U}(a,b)} \big\{ \langle c_\theta, \pi \rangle - \varepsilon H(\pi) \big\}.$$

The cost is usually parameterized as a linear combination of convex basis functions, and a convex regularizer $R(\theta)$ is added to encourage the parameter to be sparse [Carlier et al., 2020, Andrade et al., 2024] or low-rank [Dupuy et al., 2019]. To avoid a bilevel optimization procedure, the problem is reformulated as minimizing the dual objective

$$\mathcal{J}(\theta, f, g) = \int c_\theta(x, y) - f(x) - g(y) d\hat{\pi}_n(x, y) + \varepsilon \int \exp\big(\frac{f(x) + g(y) - c(x, y)}{\varepsilon}\big) d\mu(x) d\nu(y) + \lambda R(\theta). \quad (16)$$

Ma et al. [2021] optimize a similar dual objective, and parameterize both the cost and the Kantorovich potentials using MLPs. The numerical algorithm proposed in Andrade et al. [2024] uses the known form of the optimal potential g to instead optimize a semi-dual formulation of (16), which results in a better-conditioned optimization problem.

Discrete inverse OT has been considered from the perspective of matching problems [Li et al., 2019], contrastive learning [Shi et al., 2023], and economics [Dupuy and Galichon, 2014, Galichon and Salanié, 2021], as well as from a Bayesian perspective in Stuart and Wolfram [2019]. Rather than learning a specific cost matrix, Chiu et al. [2022] provide a theoretical analysis of the set of possible cost matrices.

The inverse OT setting assumes access to paired samples from an transport plan, which is an unlikely scenario in practice. It is more likely that we have partial information about data associations, such as a smaller subset of paired points. Alternatively, we may leverage known information about the transport map specific to the problem at hand as an inductive bias. The above inverse OT methods are unable to utilise such partial information.

B.2.2. DIFFERENTIATING THROUGH SINKHORN

An alternative approach for learning cost functions is to optimize according a loss that is a function of the entropy-regularized coupling matrix, which can be obtained using Sinkhorn's algorithm. The cost parameter is then updated through the bilevel optimization problem by unrolling the Sinkhorn iterations, or using implicit differentiation. Differentiating through Sinkhorn provides a more flexible cost-learning approach in comparison to inverse OT, as a specific loss function can be chosen to encourage the desired behaviour in the coupling matrix.

Liu et al. [2020] use this procedure to learn a cost function from known subset correspondences. they use Sinkhorn's algorithm to solve for the entropic coupling matrix according to the current parameterized cost function. The loss is then the sum of the squares of the entries in the resulting matrix corresponding to incorrect mappings between the known subset assignments. In the extreme case, this recovers inverse OT and the loss function is the sum of the squares of the off-diagonal elements.

Klein et al. [2024] differentiate through Sinkhorn to instead learn cost functions based on structural assumptions on the displacements. They minimize a loss $\mathcal{L}(\theta) = \langle \pi_\theta^\varepsilon, M(\theta) \rangle$, where the matrix entries $M(\theta)_{ij} = \tau_\theta(x_i - y_j)$ are a parameterized convex regularization function evaluated on the displacements. As such, they aim to learn a cost so that the resulting coupling places low mass on values with large regularization value. In particular, they consider regularizing functions $\tau_A^\perp(z) = \frac{1}{2} \|A^\perp z\|_2^2$ promoting displacements in the span of the orthonormal matrix A, which they aim to learn.

As discussed in Appendix B.1, it is often the case that we wish to obtain an estimator for the Monge map itself. Learned costs can be plugged in to Monge map estimators, though it is only recently that such solvers have been developed that

13

can handle general ground-cost functions (see Appendix B.1.1). Such estimators can be difficult to interpret as they use a neural network rather than a closed-form mapping, and if they fail to learn accurate approximations to the OT map then the resulting mapping may not display the desired properties. Moreover, it could be desired for properties to hold in the displacements of the mapping estimator itself, rather than for the displacements in the coupling matrix which are fixed according to the empirical distributions. In contrast to the aforementioned approaches, we optimize a *map estimator itself*. This avoids the need for a two-step procedure and instead optimizes the object of interest directly, ensuring that the resulting mapping displays the desired behaviour.

B.2.3. ALTERNATIVE APPROACHES TO COST LEARNING

Learning improved cost functions has been considered from the supervised metric-learning perspective, aiming to learn a ground cost between histograms that agrees with labelled 'similarity' coefficients between pairings [Cuturi and Avis, 2014, Wang and Guibas, 2012]. Heitz et al. [2020] and Pooladian et al. [2023] learn a cost from observations of a density that is assumed to be evolving optimally. Heitz et al. [2020] consider discrete measures supported on graphs, in which the cost is given by a geodesic on the graph parameterized by weights on each edge, whereas Pooladian et al. [2023] learn a Riemannian metric on the underlying continuous space. Genevay et al. [2018], Paty and Cuturi [2020] consider an adversarial setting in which the cost is chosen to maximize its discriminative power.

C. Experimental details

C.1. Inverse OT

We train according to the Inverse OT loss function given in Appendix A.2.1 using 128 pairs sampled from 2-dimensional T-shape, moon, and spiral distributions. We plot the ground-truth pairings and learned predictions for 32 newly-sampled test points in Figure 1, along with the squared-Euclidean entropic map estimator as a comparison. In all experiments, we parameterize h to be symmetric and 0.01-strongly convex, and use an ICNN with hidden layers of size $[32, 32]$. We train for 500 iterations. For the moon and spiral datasets we use a relative epsilon value of 0.01; for the T-shape dataset we decay the relative epsilon from 0.05 to 0.002. We are able to learn a mapping estimator resembling the T-shape data using a simple symmetric cost of the form $c(x, y) = h(x - y)$. For the moon and spiral datasets, we require costs of the form $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$.

C.2. Aligning OT maps to Live-seq trajectories

OT for cell profiling. Tracing individual cell transcriptome trajectories is an important problem in biological applications, and aims to identify and predict responses of cells to biological processes or external treatments. Most profiling techniques result in the destruction of the cell. The resulting data therefore consist only of individual 'snapshots' of the overall cell population without alignment. In order to infer how individual cells have developed between these snapshots, optimal transport methods can be used as an inductive bias to align the observed distributions, as it is assumed that the cell population has moved 'efficiently' in the time between observations [Schiebinger et al., 2019, Bunne et al., 2023]. Such methods typically use the squared-Euclidean cost, but it is unclear whether this is a suitable notion of distance for gene-expression data.

Live-seq. Recently, Chen et al. [2022] propose the Live-seq profiling technique, which conducts genetic profiling using only a small extract of cytoplasm from the cell. This does not require the destruction of the cell and has minimal effect on its development, therefore enabling individual cell trajectories to be observed. As only cells which provide viable samples at both timesteps can be traced, the number of individual cell trajectories that are observed is far smaller than the number of observations in each cell population. Chen et al. [2022] observe the trajectories of 12 individual cells before and after treatment, along with many more unpaired samples. It is therefore desirable to use this observed data to learn an improved cost function and corresponding OT map, which can then be used in downstream tasks such as trajectory prediction or lineage tracing. Inverse OT methods could be used to learn from only the paired samples, but they are unable to make use of the unpaired majority in the cell populations.

We restrict our attention to the adipose stem and progenitor cell (ASPC) populations before and after cell-differentiation, as these contain the majority of observed cell trajectories. Our source and target distributions therefore consist of 121 and 72 cell measurements respectively, including 9 trajectory pairings tracing the same cell.

Experiment details. We train a Monge map estimator for a symmetric cost of form $h(\Phi(x) - \Phi(y))$ to agree with these

observed trajectories, using the first 10 principal components. We use a relative epsilon value of 0.01, enforce 0.01-strong convexity of h, and use an ICNN with hidden dimensions $[64, 64, 64]$. We train both the ICNNs and flows using the Adam optimizer with learning rates 3e-3 and 1e-3 respectively, and train for 1000 iterations.

For comparison, we consider optimizing the coupling matrix rather than the mapping. To ensure a fair comparison, we first consider using the same cost parameterization $h(\Phi(x) - \Phi(y))$, and we use the entropic mapping estimator to obtain a Monge map estimator. We also compare to learning a general unstructured cost $c(x, y)$ parameterized by an MLP. To obtain a map estimator for this cost, we use the Monge gap regularizer [Uscidda and Cuturi, 2023]. When optimizing the coupling matrix, we maximize the total 'correct mass' according to the known trajectories. That is, we use the loss function

$$\mathcal{L}(\theta) = -\langle M, \pi_\theta^\varepsilon \rangle, \qquad M_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is known to map to } y_j \\ 0 & \text{otherwise.} \end{cases}$$

In Figure 2, we compare the predicted trajectories of the resulting mapping estimators to see whether they are indeed consistent with the known ground-truth trajectories. In Table 1 we report the RMSE of the predictions of the known points, and also the Sinkhorn divergence between the predicted points and the target distribution, averaged over 5 initializations. We also report the total incorrectly transported mass in the resulting entropy-regularized coupling matrices, defined as

$$\langle M, \pi_\theta^\varepsilon \rangle, \qquad M_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is known to map to } y_{j'} \text{ for } j \neq j', \text{ or } x_{i'} \text{ is known to map to } y_j \text{ for } i \neq i' \\ 0 & \text{otherwise.} \end{cases}$$

For a good learned cost, this value should be low as the entropic transport plan will transport a large amount of the mass for a point x_i to its known endpoint y_j. Note that as the coupling matrix has the correct marginals, minimizing this quantity is equivalent to the objective used when optimizing according to the matrix.

Results. By optimizing the map directly, we are able to learn a mapping that aligns with the known pairings (Figure 2), which should be expected given that this was the objective used during training. The incorrectly transported mass in Table 1 is also low, indicating that we have also learned a good cost function which places mass along the known trajectory pairings. In contrast, the maps obtained by optimizing the matrix do not align well with the ground-truth. For costs $h(\Phi(x) - \Phi(y))$ optimizing the matrix often struggles to learn a good cost, as evidenced by large amounts of incorrectly transport mass in Table 1. Some of the points for the cost $h(\Phi(x) - \Phi(y))$ are approximately correct, indicating that the learned cost has placed the mass correctly along some rows of the coupling matrix but has converged to a local minimum. The general MLP $c(x, y)$ learns a cost that places mass along the correct trajectories, which is unsurprising given the flexibility of such a cost parameterization. However, as the cost is unstructured it is difficult to interpret. Moreover, the unstructured cost means that it is difficult to obtain an estimator for the OT map, and consequently the map obtained using the Monge Gap regularizer does not align well with the Live-seq trajectories from which the cost was learned.

D. Additional Experiments

D.1. Synthetic Limited Labelled Pairs

We validate the findings from the Live-seq experiment by performing the same procedure on similar synthetic datasets, which enables us to evaluate performance on newly sampled points. We also investigate the effect of increasing the number of paired points on the out-of-sample performance of the learned mapping.

Data generation. We generate the source and target datasets by pushing samples from a $\mathcal{U} = \text{Unif}([0, 1]^d)$ distribution through two respective randomly generated diffeomorphisms Φ_1, Φ_2, so we have $\mu = \Phi_1 \# \mathcal{U}, \nu = \Phi_2 \# \mathcal{U}$. This corresponds to a continuous transformation from the source to the target via the composition of the diffeomorphisms, $\nu = (\Phi_2 \circ \Phi_1^{-1}) \# \mu$. Note that while it is likely not the case that such mappings constitute an optimal transport map, such transformations provide data distributions that resemble the Live-seq data. In the experiments, the datasets consist of an unpaired majority (i.e. most are generated by independently sampling from $\mathcal{U}$ and pushing through the corresponding mapping). A subset of the datasets are paired, meaning that $y = \Phi_2 \circ \Phi_1^{-1}(x)$. Such pairs simulate the known trajectories in the Live-seq data.

Experimental details. We perform the procedure in 2, 4 and 6 dimensions, with 32, 128 and 256 source samples and 40, 160 and 320 target samples respectively. A subset of these datasets are known pairings. We use the same experimental setup as for the Live-seq data. In Figure 3, we report the incorrectly transported mass in the entropy-regularized coupling matrix for the training data as previously, as well as the RMSE error for the observed paired points. We also generate unseen
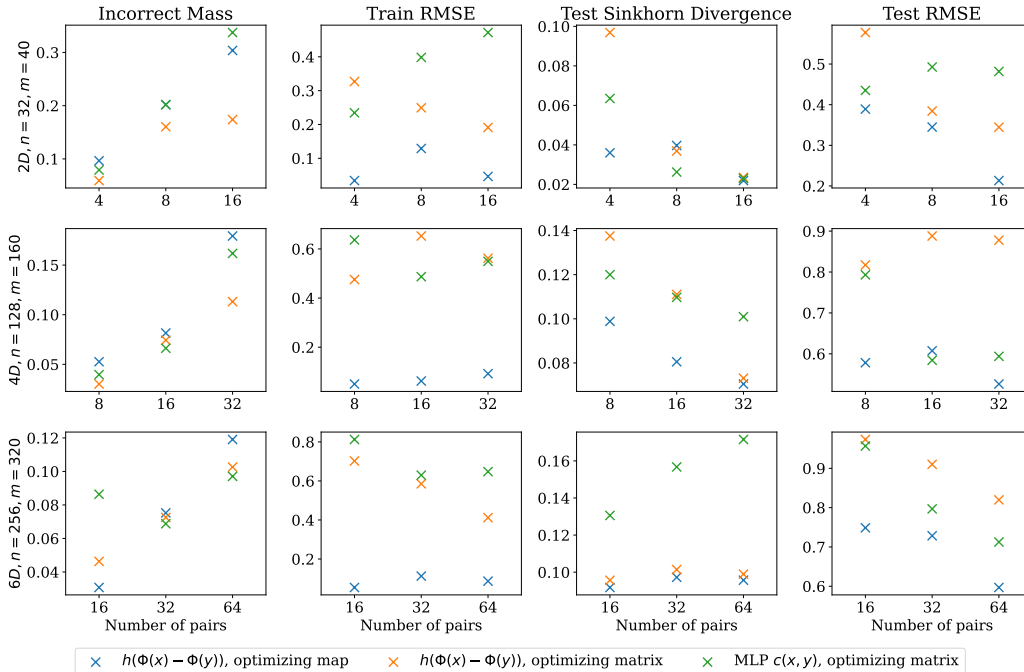
Figure 3. Results for the synthetic limited labelled pairs experiment. Optimizing the map estimator results in a mapping that aligns significantly better with the known paired points, and also demonstrates improved prediction for out-of-sample points. The learned cost appears comparable to those learned by optimizing the coupling.

test samples $\tilde{\mu}, \tilde{\nu}$ consisting of fully paired points, and report the Sinkhorn divergence $S_\varepsilon(\hat{T}\#\tilde{\mu}, \tilde{\nu})$ and the RMSE of the resulting mapping estimator $\hat{T}$ for these newly-sampled points. The results are averaged over 5 different randomly-generated data distributions.

Results. The results are consistent with the observations in the Live-seq experiment. Optimizing the mapping appears to result in good learned costs. The amount of mass placed along incorrect directions in the coupling matrix is generally similar to those learned by optimizing the matrix, which are in fact optimized to minimize this objective.

The OT map estimator obtained from optimizing the mapping shows significantly better alignment with the known pairings, which is as expected given that is the objective being optimized. The resulting map estimator also provides better out-of-sample performance on the newly-sampled test points. In contrast, those learned from optimizing the matrix are much less consistent with the known ground-truths and generally perform worse on the out-of-sample points. Optimizing the mapping also appears to result in a mapping giving a lower Sinkhorn divergence when transporting the newly sampled points, demonstrating an improved fit to the target at a distributional level.

We also remark that the entropic mapping obtained from the cost $h(\Phi(x) - \Phi(y))$ learned by optimizing the matrix occasionally failed to give an appropriate mapping (giving very large Sinkhorn divergences between the mapped points and the target distribution). This is presumably because of a poor choice of ε when constructing the estimator. We disregard such results when calculating the averages in Table 3. In contrast, optimizing the map directly ensured that the final entropic mapping was always reasonable.

D.2. Inducing properties on the transport map displacements

As we are optimizing according to the OT mapping estimator itself, we can also optimize to encourage properties we wish to hold on the resulting displacements themselves. This allows us to leverage knowledge about the structure of the mapping as an inductive bias when learning adapted cost functions and OT map estimators. We demonstrate the ability to induce low-rank and 2-directional displacements by training according to the loss functions proposed in Appendix A.2.2, with an addition Sinkhorn divergence term $S_\varepsilon(\hat{T}\#\mu, \nu)$ to ensure a good fit to the target distribution. We train the map estimators using 3-dimensional empirical measures, each consisting of 128 datapoints, and use a symmetric cost of the form
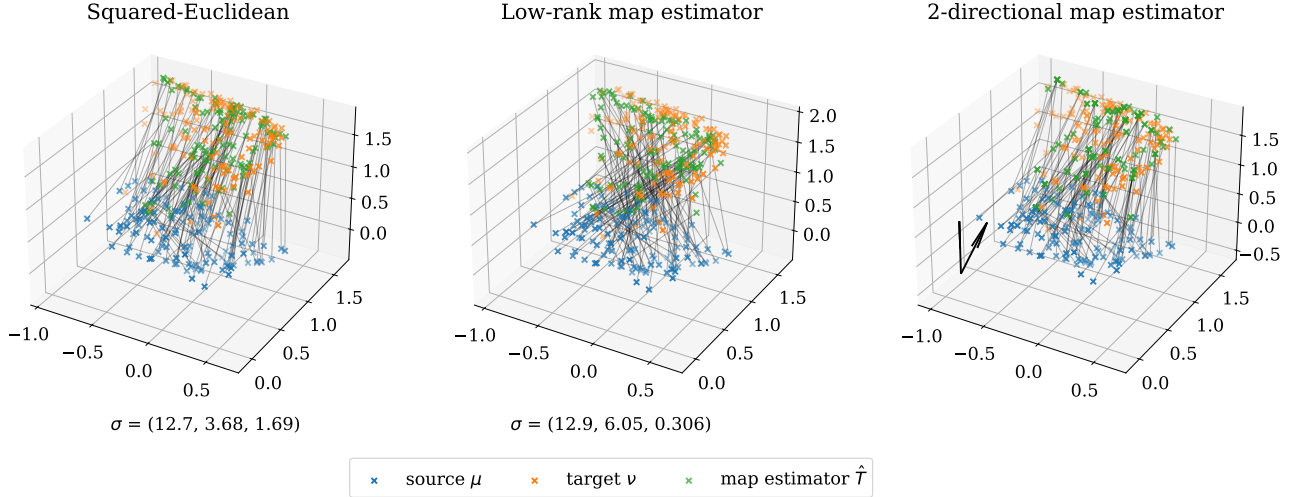
Figure 4. *(middle)* The learned low-rank mapping exhibits displacements primarily along a 2-dimensional plane, as demonstrated by the small final singular value. *(right)* The displacements of the learned 2-directional mapping occur primarily along the displayed directions, which were learned during training.

$c(x, y) = h(x - y)$ with $\alpha = 0.01$ and an ICNN with hidden dimensions $[64, 64]$. Figure 4 plots the learned mappings applied to 128 newly sampled points, again with the squared-Euclidean entropic map estimator as a comparison. We see that the learned mappings display the desired structural properties.

E. Proofs
Proposition 3.1. *For a smooth, strictly convex function h with domain $\mathbb{R}^d$, we have*

$$(\nabla h)^{-1}(x) = \arg\min_z \left\{ h(z) - \langle z, x \rangle \right\}. \tag{7}$$

Proof. Let h^* denote the convex conjugate, $h^*(x) = \sup_z \left\{ \langle z, x \rangle - h(z) \right\}$ where $\langle \cdot, \cdot \rangle$ denotes the usual inner product on $\mathbb{R}^d$. As h is smooth and convex on the domain $\mathbb{R}^d$, it is essentially smooth [Rockafellar, 1970]. As it is also strictly convex and differentiable, it is of Legendre type on $\mathbb{R}^d$, and we therefore have $(\nabla h)^{-1} = \nabla h^*$ [Rockafellar, 1970, Theorem 26.5]. The expression for ∇h^* then follows from Danskin's theorem. $\square$

Theorem 3.2. *Under the conditions of Theorem 2.1 with a cost of the form $c(x, y) = h(\Phi_\mu(x) - \Phi_\nu(y))$ for strictly convex h and diffeomorphisms Φ_μ, Φ_ν, the optimal plan is unique, of the form $(Id, T^\star)\#\mu$, where $T^\star$ can be written:*

$$T^\star(x) = \Phi_\nu^{-1}\left[\Phi_\mu(x) - (\nabla h)^{-1} \circ \nabla f^\star \circ \Phi_\mu(x) \right]. \tag{8}$$

Proof. Define the push-forward measures $\tilde{\mu} = \Phi_\mu \# \mu, \tilde{\nu} = \Phi_\nu \# \nu$ and consider the following two Kantorovich problems, denoting the original problem (K) and a transformed version $(\tilde{K})$.

$$\arg\inf_{\pi \in \Gamma(\mu, \nu)} \iint_{\mathbb{R}^d \times \mathbb{R}^d} h(\Phi_\mu(x) - \Phi_\nu(y)) \mathrm{d}\pi(x, y) \tag{K}$$

$$\arg\inf_{\tilde{\pi} \in \Gamma(\tilde{\mu}, \tilde{\nu})} \iint_{\mathbb{R}^d \times \mathbb{R}^d} h(x - y) \mathrm{d}\tilde{\pi}(x, y) \tag{$\tilde{K}$}$$

We can construct a mapping $F : \Gamma(\mu, \nu) \to \Gamma(\tilde{\mu}, \tilde{\nu})$ between the respective sets of admissible transport plans defined as $\pi \mapsto \tilde{\pi} = (\Phi_\mu \otimes \Phi_\nu)\#\pi$. The fact that the coupling $\tilde{\pi}$ has the correct marginals $\tilde{\mu}, \tilde{\nu}$ is a consequence of the definition of

push-forward; for a test function $\varphi \in \mathcal{C}^\infty(\mathbb{R}^d)$, we have

$$
\begin{aligned}
\iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(x)\mathrm{d}\tilde{\pi}(x,y) &= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(x)\mathrm{d}(\Phi_\mu \otimes \Phi_\nu)\#\pi(x,y) \\
&= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(\Phi_\mu(x))\mathrm{d}\pi(x,y) \\
&= \int_{\mathbb{R}^d} \varphi(\Phi_\mu(x))\mathrm{d}\mu(x) \\
&= \int_{\mathbb{R}^d} \varphi(x)\mathrm{d}\tilde{\mu}(x).
\end{aligned}
$$

This shows that $\tilde{\pi}$ has correct first marginal $\tilde{\mu}$ and it can be shown similarly that the second marginal is $\tilde{\nu}$, confirming that $\tilde{\pi} \in \Gamma(\tilde{\mu}, \tilde{\nu})$. Consider too the mapping G given by $\tilde{\pi} \mapsto \pi = (\Phi_\mu^{-1} \otimes \Phi_\nu^{-1})\#\tilde{\pi}$, which defines a map from $\Gamma(\tilde{\mu}, \tilde{\nu})$ to $\Gamma(\mu, \nu)$. For a test function $\varphi \in \mathcal{C}^\infty(\mathbb{R}^d \times \mathbb{R}^d)$,

$$
\begin{aligned}
\iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(x,y)\mathrm{d}(\Phi_\mu^{-1} \otimes \Phi_\nu^{-1})\#(\Phi_\mu \otimes \Phi_\nu)\#\pi(x,y) &= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(\Phi_\mu^{-1}(x), \Phi_\nu^{-1}(y))\mathrm{d}(\Phi_\mu \otimes \Phi_\nu)\#\pi(x,y) \\
&= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi((\Phi_\mu \circ \Phi_\mu^{-1})(x), (\Phi_\nu \circ \Phi_\nu^{-1})(y))\mathrm{d}\pi(x,y) \\
&= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \varphi(x,y)\mathrm{d}\pi(x,y).
\end{aligned}
$$

This shows that $G \circ F = Id$, and similarly we can show that $F \circ G = Id$. We thus conclude $G = F^{-1}$, and that F is a bijection between $\Gamma(\mu, \nu)$ and $\Gamma(\tilde{\mu}, \tilde{\nu})$.

Define I_K and $I_{\tilde{K}}$ to be the respective infimums for the Kantorovich problems (K) and $(\tilde{K})$. For any $\pi \in \Gamma(\mu, \nu)$, the above shows that $\tilde{\pi} = F(\pi)$ is an admissible transport plan for $(\tilde{K})$ and thus

$$
I_{\tilde{K}} \le \iint_{\mathbb{R}^d \times \mathbb{R}^d} h(x - y)\mathrm{d}\tilde{\pi}(x,y) = \iint_{\mathbb{R}^d \times \mathbb{R}^d} h(\Phi_\mu(x) - \Phi_\nu(y))\mathrm{d}\pi(x,y). \tag{17}
$$

Note that as Φ_μ, Φ_ν are diffeomorphisms, the transformed Kantorovich problem $(\tilde{K})$ satisfies the conditions of Theorem 2.1 so has a unique solution $\tilde{\pi}^\star$. Letting $\hat{\pi} = F^{-1}(\tilde{\pi}^\star)$, we have

$$
\iint_{\mathbb{R}^d \times \mathbb{R}^d} h(\Phi_\mu(x) - \Phi_\nu(y))\mathrm{d}\hat{\pi}(x,y) = \iint_{\mathbb{R}^d \times \mathbb{R}^d} h(x - y)\mathrm{d}\tilde{\pi}^\star(x,y) = I_{\tilde{K}}. \tag{18}
$$

We see that $\hat{\pi}$ attains the lower bound in (17), and is therefore an optimal plan for the original Kantorovich problem (K).

For uniqueness, note that if there are two such optimal plans $\pi_1^\star, \pi_1^\star \in \Gamma(\mu, \nu)$, then (again using the definition of pushforwards) we have that both $F(\pi_1^\star), F(\pi_2^\star)$ attain the infimum $I_{\tilde{K}}$. By uniqueness of $\tilde{\pi}^\star$, we must have $F(\pi_1^\star) = F(\pi_2^\star) = \tilde{\pi}^\star$, and inverting F we see $\pi_1^\star = \pi_2^\star$ as required.

The structure of the optimal plan $\pi^\star$ follows from that of $\tilde{\pi}^\star$. Recall from Theorem 2.1 that $\tilde{\pi}^\star$ is of the form $(Id, \tilde{T}^\star)$ where $\tilde{T}^\star = Id - (\nabla h)^{-1} \circ (\nabla f^\star)$ for the Kantorovich potential $f^\star$. Applying the above result, we therefore see that the optimal plan $\pi^\star$ for our original Kantorovich problem (K) is also of the form $(Id, T^\star)$. The map $T^\star$ is now given by

$$
T^\star(x) = \Phi_\nu^{-1}\left[\Phi_\mu(x) - (\nabla h)^{-1} \circ \nabla f^\star \circ \Phi_\mu(x)\right],
$$

where $f^\star$ is the Kantorovich potential for the transformed Kantorovich problem $(\tilde{K})$.

$\square$