Supervised Training of Conditional Monge Maps

Anonymous Author(s)
Affiliation
Address
email

Abstract

Optimal transport (OT) theory describes general principles to define and select, among many possible choices, the most efficient way to map a probability measure onto another. That theory has been mostly used to estimate, given a pair of source and target probability measures (μ, ν), a parameterized map T_θ that can efficiently map μ onto ν. In many applications, such as predicting cell responses to treatments, the data measures μ, ν (features of untreated/treated cells) that define optimal transport problems do not arise in isolation but are associated with a *context* c (the treatment). To account for and incorporate that context in OT estimation, we introduce CONDOT, an approach to estimate OT maps conditioned on a context variable, using several pairs of measures (μ_i, ν_i) tagged with a context label c_i. Our goal is to learn a *global* map $\mathcal{T}_\theta$ which is not only expected to fit *all pairs* in the dataset $\{(c_i, (\mu_i, \nu_i))\}$, i.e., $\mathcal{T}_\theta(c_i)\sharp\mu_i \approx \nu_i$, but should *generalize* to produce meaningful maps $\mathcal{T}_\theta(c_{\text{new}})$ conditioned on unseen contexts c_{new}. Our approach harnesses and provides a novel usage for *partially input convex neural networks*, for which we introduce a robust and efficient initialization strategy inspired by Gaussian approximations. We demonstrate the ability of CONDOT to infer the effect of an arbitrary combination of genetic or therapeutic perturbations on single cells, using only observations of the effects of said perturbations separately.

1 Introduction

A key challenge in the treatment of cancer is to predict the effect of (a combination of) drugs on cells of a particular patient. Recent techniques enable us to obtain measurements from individual cells, in treated and untreated conditions, which however are not in correspondence. Given such examples of untreated and treated cells under different drugs, can we predict the effect of new drug combinations? We develop a general approach motivated by this and related problems, through the lens of *optimal transport theory*, and, in that process, develop tools that might be of interest for other application domains of OT. Given a collection of N pairs of measures (μ_i, ν_i) over $\mathbb{R}^d$ (cell measurements), tagged with a context c_i (encoding the treatment), we seek to learn a context-dependent, parameterized transport map $\mathcal{T}_\theta$ such that, on training data, $\mathcal{T}_\theta(c_i)\sharp\mu_i \approx \nu_i$. Additionally, our goal is to expect this parameterized map to be able to generalize to unseen contexts c_{new} in the sense that given a patient's cells described in μ one might predict the effect of context c_{new} on these cells as $\mathcal{T}_\theta(c_{\text{new}})\sharp\mu$.

Learning Mappings Between Measures From generative adversarial networks to normalizing flows and diffusion models, the problem of learning maps that move points from a source to a target distribution is central to machine learning and its applications. Optimal transport theory (Santambrogio, 2015) has emerged as a principled approach to estimate such maps. For a pair of measures μ, ν supported in $\mathbb{R}^d$, OT suggests that, among all maps T such that ν can be reconstructed by applying T to every point in the support of μ, summarized with the push-forward notation as $T\sharp\mu \approx \nu$, one should favor maps T that minimize the length of the average displacement $\|T(x) - x\|^p$, where $p \geq 1$.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

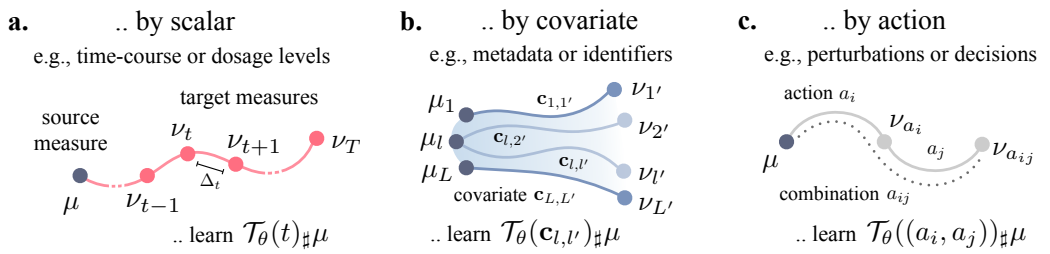

Figure 1: The evolution of a source μ into a target measure ν can be dependent on context variables c of various nature. This comprises **a.** scalars such as time or dosage t which determine the magnitude of an optimal transport, **b.** flow of measures into another one based on additional information (possibly different between μ and ν) stored in vectors $\mathbf{c}_{l,l'}$, or **c.** discrete and complex actions a_i, possibly in combination a_{ij}. We seek a unified framework for learning a map $\mathcal{T}_\theta$ to predict a target measure ν from any type of condition.

A rich literature, summarized by Peyré and Cuturi (2019), addresses computational challenges of estimating such couplings and maps, and has found impactful applications in several areas of science (cf., Hashimoto et al., 2016; Schiebinger et al., 2019; Yang et al., 2020; Janati et al., 2020).

We focus in this work on neural approaches that parameterize the optimal maps T via neural networks. An early approach is the work on Wasserstein GANs (Arjovsky et al., 2017), albeit the transport map is not explicitly estimated. Several recent results have exploited a more explicit connection between OT and NNs, derived from the celebrated Brenier theorem (1987), that is applicable to the case $p = 2$. That theorem states that Monge maps are necessarily gradients of convex functions. As a result, one can rely on the family of input convex neural networks (ICNN) (Amos et al., 2017) to parameterize either the Monge map (Jacob et al., 2018; Yang and Uhler, 2019) or a dual potential (Makkuva et al., 2020; Korotin et al., 2020) as, respectively, the gradient of an ICNN or an ICNN itself. These approaches have since been used to model perturbation effects on single-cell populations (Bunne et al., 2021, 2022). In this paper, we build on this line of work, but substantially generalize it, to learn a *parametric* family of context-aware transport maps, using a collection of labeled pairs of measures.

Contributions We propose a framework that can leverage *labeled* pairs of measures $\{(c_i, (\mu_i, \nu_i))\}_i$ to infer a *global* parameterized map $\mathcal{T}_\theta$. Hereby, the context c_i belongs to an arbitrary set $\mathcal{C}$. We construct $\mathcal{T}_\theta$ so that it should be able, given a possibly unseen context label $c \in \mathcal{C}$, to output a map from $\mathbb{R}^d$ onto itself, $\mathcal{T}_\theta(c) : \mathbb{R}^d \to \mathbb{R}^d$, ensuring that it is itself the gradient of a convex function. To that end, we propose to learn these parameterized Monge maps $\mathcal{T}_\theta$ as the gradients of partially input convex neural networks (PICNN), which we borrow from the foundational work of Amos et al. (2017). Our framework can be alternatively interpreted as a hypernetwork (Ha et al., 2016): The PICNN architecture can be interpreted as an ICNN whose weights and biases are *modulated* using a context vector c; in that sense c parameterizes a family of convex potentials in the input space. Because both ICNN —and to a greater extent PICNN— are notoriously difficult to train (Richter-Powell et al., 2021), we use closed-form solutions between Gaussian approximations to derive relevant parameter initializations for (P)ICNNs: These choices ensure that, *upon initialization*, the gradient of the (P)ICNNs mimics an affine Monge map between Gaussian approximations of measures μ_i, ν_i, which is then gradually refined to a fully non-linear map using a proper loss minimization on its parameters. We demonstrate the usefulness of our framework on three crucial scenarios: Parameterization of transport through a real variable (time or drug dosage), through an auxiliary informative variable (cell covariates) and through action variables (genetic perturbations in combination) (see Fig. 1). Our results demonstrate the ability of our architectures to better capture on out-of-sample observations the effects due to these variables in various settings, even when considering never-seen, composite context labels. These results suggest potential applications of conditional OT to model personalized medicine outcomes or to guide novel experiments, where OT could serve as a predictor for never tested context labels.

2 Background on Neural Solvers for the 2-Wasserstein Problem

Optimal Transport The Monge problem between two measures $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, here restricted to measures supported on $\mathbb{R}^d$ and compared with the squared Euclidean metric, reads

$$T^\star_{\mu,\nu} := \arg\inf_{T\sharp\mu=\nu} \int_{\mathbb{R}^d} \|x - T(x)\|^2 d\mu(x)\,. \tag{1}$$

The existence of $T^\star_{\mu,\nu}$ is guaranteed under fairly general conditions (Santambrogio, 2015, Theorem 1.22), which require that μ and ν have finite L_2 norm, and that μ puts no mass on $(d-1)$ surfaces of class $\mathcal{C}_2$. This can be proved with the celebrated Brenier theorem (1987), which states that there must exist a unique (up to the addition of a constant) potential $\psi^\star_{\mu,\nu} : \mathbb{R}^d \to \mathbb{R}$ such that $T^\star_{\mu,\nu} = \nabla\psi^\star_{\mu,\nu}$. This theorem has far-reaching impact: It is sufficient, when seeking optimal transport maps, to restrict the computational effort to seek a "good" convex potential such that its gradient pushes μ towards ν. This result has been exploited to propose OT solvers that rely on input convex neural networks (ICNNs) (Amos et al., 2017). The practical estimation of such maps can be carried out using various tricks (Makkuva et al., 2020; Korotin et al., 2020; Bunne et al., 2022; Alvarez-Melis et al., 2021; Mokrov et al., 2021). For instance, a related reformulation of Problem (1) is to recover the following identy linking Monge and dual optimal solutions (writing f^* for the Legendre transform of f):

$$f^\star_{\mu,\nu} := \arg\sup_{f \text{ convex}} \int_{\mathbb{R}^d} f^* \mathrm{d}\mu + \int_{\mathbb{R}^d} f \mathrm{d}\nu \;= \psi^\star_{\mu,\nu}\,. \tag{2}$$

Convex Neural Architectures Input convex neural networks (ICNN) are neural networks ψ_θ that admit certain constraints on their architecture and parameters θ, such that their output $\psi_\theta(x)$ is a convex function of their input x (Amos et al., 2017). As a result, they have been increasingly used as drop-in replacements to the set of admissible functions in (2). Practically speaking, an ICNN is a K-layer, fully connected network such that, at each layer index k from 0 to $K-1$, one has

$$z_{k+1} = \sigma_k(W^x_k x + W^z_k z_k + b_k) \text{ and } \psi_\theta(x) = z_K\,, \tag{3}$$

where, by convention, z_0 and W^z_0 are 0, and σ_k are *convex* non-decreasing activation functions, $\theta = \{b_k, W^z_k, W^x_k\}_{k=0}^{K-1}$ are the weights and biases of the neural network. While ample flexibility is provided to choose dimensions for intermediate hidden states z_k, the last layer must necessarily produce a scalar, hence W^x_{K-1} and W^z_{k-1} are line vectors and $b_{K-1} \in \mathbb{R}$. ICNNs are characterized by the fact that all weight matrices W^z_k associated to latent representations z must have non-negative entries, which ensures their convexity w.r.t. inputs x. We encode this constraint by identifying these matrices as the softplus of other matrices of the same size, or, alternatively, using a regularizer that penalizes the negative entries of these matrices. Since the work by Amos et al. (2017), convex neural architectures have been used within the context of OT to model convex dual functions (Makkuva et al., 2020), or normalizing flows derived from convex potentials (Huang et al., 2021). Their expressivity and universal approximation properties have been studied by Chen et al. (2019), who show that any convex function over a compact convex domain can be approximated in sup norm by an ICNN.

3 Supervised Training of Conditional Monge Maps

We assume we are given N pairs of measures $(\mu_i, \nu_i) \in \mathcal{P}(\mathbb{R}^d)^2$, each labeled by a context value $c_i \in \mathcal{C}$. We make minimal assumptions on $\mathcal{C}$. For instance, $\mathcal{C}$ could be $\mathbb{R}$: This can be useful to encode a time-span Δt that occurred between the observation of the same population on two timestamps, i.e., μ_i, then ν_i, or a quantification of strength of dosage of a single treatment of interest. More generally, context c can encode metadata or identifiers provided as additional inputs to describe the mass transfer from μ_i to ν_i (capturing, e.g., the effect of different experimental conditions). Finally, they can encode more complex actions, possibly described through a learned representation, and, as we present later in §3.2, allowing for combinations of such actions.

3.1 Integrating Context in Convex Architectures

Towards Parameterized Monge Maps Building on Brenier's theorem and following Korotin et al. (2020); Makkuva et al. (2020), we retain the idea that OT maps between two measures μ_i and ν_i should be estimated as the gradients of a convex Brenier potential $\nabla\psi$, and that such convex functions can be modeled using neural architectures. Unlike previous approaches, however, our goal is to *(i.)* avoid the estimation of a new ICNN for each available pair (μ_i, ν_i), and, instead, pool

efficiently these samples to train in a multi-task manner an overall neural architecture whose weights can be shared efficiently, while *(ii.)* leveraging context values c_i as much as possible to facilitate joint training and improve generalization. To reach these two goals, we propose to learn a *unique* parameterized map $\mathcal{T}_\theta$ that outputs, for any context value c (either known and seen in the dataset, or unknown), a $\mathbb{R}^d \to \mathbb{R}^d$ map $\mathcal{T}_\theta(c)$ that has the property that if $c \approx c_i$ then $\mathcal{T}_\theta(c)\#\mu_i \approx \nu_i$. This goal can be realized by defining a parameterized potential function $\psi_\theta : \mathbb{R}^d \times \mathcal{C}$ that takes that context as an additional input, to define next our parameterized families of OT maps as

$$\mathcal{T}_\theta(c) := x \mapsto \nabla_1 \psi_\theta(x, c).$$

A crucial observation is that the resulting parameterized potential $\psi_\theta(x, c)$ only needs to be input convex in x, *but not in the context value* c to ensure that $\mathcal{T}_\theta(c)$ is an OT map. Thus, to obtain this property, we build conditional Monge maps using *partially input convex neural networks* (PICNN). Note that the use of PICNNs within the context of OT was recently proposed and explored in (Fan et al., 2021, Appendix B), where the PICNN provides an elegant reformulation of the Wasserstein barycenter problem using a neural approach. In Fan et al. (2021), however, the role of our context vector c is restricted to simplicial weights.

Partially Convex Neural Architectures PICNNs are neural networks that can be evaluated over a pair of inputs (x, c), but which are only required to be convex w.r.t. x. Such networks were proposed by Amos et al. (2017, Eq. 3) to address a problem that is arguably symmetric to ours: Their inputs are labeled as (y, x), where y is a label vector, whose dimension is typically much smaller than that of vector x. Their PICNN is convex w.r.t. y, in order to easily recover, given a datapoint x (e.g., an image) the best label y that corresponds to x using gradient descent as a subroutine, i.e. $y^\star(x) = \arg\min_y \text{PICNN}_\theta(x, y)$. PICNN were therefore originally proposed to learn a parameterized, implicit classification layer amortized over samples, whose entire motivation rests on the property that it is convex w.r.t. label variable y. By contrast, our proposal is to make it convex w.r.t. x, the large dimensional input vector of interest. In addition to that swap, we do not use the convexity of the PICNN to define an implicit layer (or to carry out gradient descent). Indeed, it does not make sense in our setting to minimize $\psi_\theta(x, c)$ as a function of x, since x is an observation. Instead, our work rests on the property that $\nabla_1 \psi_\theta(x, c)$ describes a parameterized family of OT maps.

Given inputs x (samples of source distribution in $\mathbb{R}^d$) and context $c \in \mathcal{C}$, we define the PICNN over (x, c). We assume for now that $\mathcal{C} \subset \mathbb{R}^q$ and is encoded as a vector. When $\mathcal{C}$ is not a vector space, we use embeddings or combinators as described in §3.2. A K-layer PICNN is defined as:

$$\begin{aligned} u_{k+1} &= \tau_k \left(V_k u_k + v_k\right) \\ z_{k+1} &= \sigma_k \left(W_k^z \left(z_k \circ [W_k^{zu} u_k + b_k^z]_+\right) + W_k^x \left(x \circ (W_k^{xu} u_k + b_k^x)\right) + W_k^u u_k + b_k^u\right) \\ \psi_\theta(x, c) &= z_k, \end{aligned} \tag{4}$$

where the PICNN is initialized as $u_0 = c, z_0 = \mathbf{0}$, $\circ$ denotes the Hadamard elementwise product, and τ_k is any activation function. The parameters of the PICNN are then given by

$$\theta = \{V_k, W_k^z, W_k^{zu}, W_k^x, W_k^{xu}, W_k^u, v_k, b_k^z, b_k^x, b_k^u\}.$$

Similar to ICNNs, the convexity w.r.t. input variable x is guaranteed as long as activation functions σ_i are convex and non-decreasing, and the weight matrices W_k^z have non-negative entries. We parameterize this by storing them as elementwise applications of softplus operations on precursor matrices of the same size, or, alternatively, by regularizing their negative part. Finally, all matrices at the $K-1$ layer are line vectors, and their biases scalars.

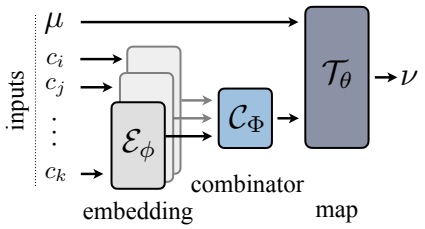

Figure 2: **CONDOT Architecture and Modules**. The embedding module $\mathcal{E}_\phi$ embeds arbitrary conditions c, which are then combined via module $\mathcal{C}_\Phi$. Using the processed contexts c, the map $\mathcal{T}_\theta(c)$ acts on μ to predict the target measure ν.

3.2 Conditional Monge Map Architecture

Using PICNNs as a base module, the CONDOT architecture integrates operations on the contexts $\mathcal{C}$. As seen in Figure 1, context values c may take various forms:

1. A scalar t denoting a strength or a temporal effect. For instance, McCann's interpolation and its time parameterization, $\alpha_t = ((1-t)\text{Id} + tT)_\sharp \alpha_0$ (McCann, 1997) can be interpreted as a trivial conditional OT model that creates, from an OT map ψ_θ, a set of maps parameterized by t, $\mathcal{T}_\theta(t) := x \mapsto \nabla_x \left((1-t)\|x\|^2/2 + t\psi_\theta(x)\right)$.

2. A covariate vector influencing the nature of the effect that led μ_i to ν_i.
3. One or multiple actions, possibly discrete, representing decisions or perturbations applied onto μ_i.

To provide a flexible architecture capable of modeling different types of conditions as well as conditions appearing in combinations, the more general CONDOT architecture consists of the hypernetwork $\mathcal{T}_\theta$ that is itself given a context vector through embedding and combinator modules. This generic architecture provides a one-size fits all approach to integrate all types of context conditions c.

Embedding Module To give greater flexibility when setting the context variable c, CONDOT contains an embedding module $\mathcal{E}$ that translates arbitrary contexts into real-valued vectors. Besides simple scalars t (Fig. 1a) for which no embedding is required, discrete contexts can be handled with an embedding module $\mathcal{E}_\phi$. When the set $\mathcal{C}$ is small, this can be done effectively using one-hot embeddings $\mathcal{E}_{\text{ohe}}$. For more complicated actions a such as treatments, there is no simple way to vectorize a context c. Similarly to action embeddings in reinforcement learning (Chandak et al., 2019; Tennenholtz and Mannor, 2019), we can learn embeddings for discrete actions into a learned continuous representation. This often requires domain-knowledge on the context values. For molecular drugs, for example, we can learn molecular representations $\mathcal{E}_{\text{mol}}$ such as chemical, motif-based (Rogers and Hahn, 2010) or neural fingerprints (Rong et al., 2020). However, often this domain knowledge is not available. In this work, we thus construct so-called *mode-of-action* embeddings, by computing an embedding $\mathcal{E}_{\text{moa}}$ that encourages actions a with similar effect on target population ν to have a similar representation. In § 5, we analyze several embedding types for different use-cases.

Combinator Module While we often have access to contexts c in isolation, it is crucial to infer the effect of contexts applied in combination. A prominent example are cancer combination therapies, in which multiple treatment modalities are administered in combination to enhance treatment efficacy (Kummar et al., 2010). In these settings, the mode of operation between individual contexts c is often not known, and can thus not be directly modeled via simple arithmetic operations such as `min, max, sum, mean`. While we test as a baseline the case, applicable to one-hot-embeddings, where simple additions are used to model these combinations, we propose to augment the CONDOT architecture with a parameterized combinator module $\mathcal{C}_\Phi$. If the order in which the actions are applied is irrelevant or unknown, the corresponding network $\mathcal{C}_\Phi$ needs to be permutation-invariant, which can be achieved by using a deep set architecture (Zaheer et al., 2017). Receiving a flexible number of inputs from the embedding module $\mathcal{E}_\phi$, CONDOT allows for a joint training of the PICNN parameters θ, embedding parameters ϕ, and combinator parameters Φ in a single, end-to-end differentiable architecture.

Training Procedure Given a dataset $\mathcal{D} = \{c_i, (\mu_i, \nu_i)\}_{i=0}^N$ of N pairs of populations before μ_i and after transport ν_i connected to a context c_i, we detail in Algorithm 1 provided in § B, a training loop that incorporates all of the architecture proposals described above. The training loss aims at making sure the map $\mathcal{T}_\theta(c_i)$ is an OT map from μ_i to ν_i, where c_i may either be the original label itself or its embedded/combined formulation in more advanced tasks. To that end, we use the $\widetilde{W}_2^2$ proxy dual objective defined in (Makkuva et al., 2020, Eq.6). This involves training the CONDOT architecture using two PICNNs, i.e., PICNN_{θ_f} and PICNN_{θ_g}, that share the same embedding/combinator module, with a regularization $\mathcal{V}$ promoting that for any c, the $\text{PICNN}_{\theta_g}(\cdot, c)$ resembles the Legendre transform of the other, $\text{PICNN}^*_{\theta_f}(\cdot, c)$.

4 Initialization Strategies for Neural Convex Architectures

We address the problem of initializing the parameters of (P)ICNNs to ensure their gradient evaluated at every point is (initially) meaningful in the context of OT, namely that it is able to map the first and second moments of a measure μ into those of a target measure ν. The initializers we propose build heavily on the quadratic layers proposed in the seminal reference (Korotin et al., 2020, Appendix B.2), notably the "DenseQuad" layer, as well as on closed-form solutions available for Gaussian approximations of measures (Gelbrich, 1990)

Closed-Form Potentials for Gaussians Given two Gaussian distributions $\mathcal{N}_1, \mathcal{N}_2$ with means respectively $\mathbf{m}_1, \mathbf{m}_2$ and covariance matrices Σ_1, Σ_2 (where Σ_1 is assumed to be full rank), the

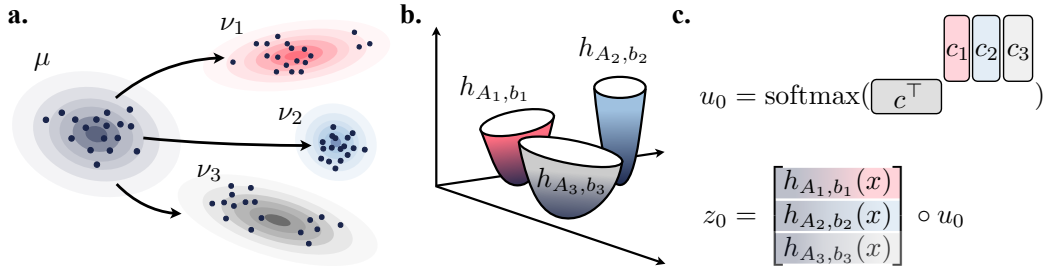

Figure 3: **a.** From a measure μ to several target measures ν_1, ν_2, ν_3 provided with labels c_1, c_2, c_3 we can extract three Gaussian (quadratic) potentials in closed form, **b.** whose gradients transport on a first approximation μ to areas in space that cover the three targets. **c.** Given a new label vector c, we compare it to known labels to modulate the magnitude of each of the three potentials.

Brenier potential solving the OT problem from the first to the second Gaussian reads:

$$f^\star_{\mathcal{N}_1,\mathcal{N}_2} = \tfrac{1}{2} x^T A^T A x + b^T x + t(A,b) = \tfrac{1}{2}\|Ax\|_2^2 + b^T x + t(A,b), \text{ where,} \tag{5}$$

$$A := \left(\Sigma_1^{-1/2}\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\Sigma_1^{-1/2}\right)^{1/2}, \quad b := \mathbf{m}_2 - A^T A \mathbf{m}_1,$$

define both quadratic and linear terms and $t(A,b)$ can be any constant. Importantly, note that we write the quadratic term in factorized form AA^T to enforce psd-ness, as done by Korotin et al. (2020), not as usually done with a single psd matrix (Peyré and Cuturi, 2019, Remark 2.31).

Our quadratic potentials are only injected in the first state of hidden vector z_0, to populate it with a collection of relevant full-rank quadratic convex functions, with the goal of recovering an affine OT map from the start, as illustrated in the experiments from § C.1.

Quadratic Potentials Lower Bounded by 0 Naturally, for any choice of $t(A,b)$ one recovers the property that $\nabla f^\star_{\mu,\nu}\sharp\mathcal{N}_1 = \mathcal{N}_2$. When used in deep architectures, the level of that constant does, however, play a role, since convex functions in ICNN are typically thresholded or modulated using rectifying functions. To remove this ambiguity, we settle on a choice for $t(A,b)$ that is such that the lowest value reached by $f^\star_{\mathcal{N}_1,\mathcal{N}_2}$ is 0. This can be obtained by setting

$$t(A,b) := b^T (A^T A)^{-1} b\,, \tag{6}$$

which results in the following choice, writing $\omega = \mathbf{m}_1 - (A^T A)^{-1}\mathbf{m}_2$,

$$f^\star_{\mathcal{N}_1,\mathcal{N}_2}(x) = \tfrac{1}{2}\|A\left(x + (A^T A)^{-1} b\right)\|_2^2 = \tfrac{1}{2}\|A\left(x - \omega\right)\|_2^2\,. \tag{7}$$

To mimic these potential functions, we introduce a quadratic *layer* parameterized by a weight matrix M and a "bias" vector m, defined as $q_{M,m}(x) = \frac{1}{2}\|M(x-m)\|_2^2$. By design, $q_{M,m}(x)$ is a convex quadratic, non-negative layer. Finally, one has the following relationships,

$$\nabla q_{I,\mathbf{0}_d} = \text{Id}, \quad \nabla q_{A,\omega}\sharp\mathcal{N}_1 = \mathcal{N}_2\,. \tag{8}$$

ICNN Initialization We explore two possible ICNN (3) initializers for OT.

Identity Initialization The first approach ensures that upon initialization the ICNN's gradient mimics the *identity* map, i.e., $\nabla\psi_\theta(x) = x$ for any x. We do so by injecting in the initial hidden state z_0 the norm of the input vector $\frac{1}{2}\|x\|^2$, cast as a trainable layer $q_{M,m}$ initialized with $M = I$ and $m = \mathbf{0}_d$, see (8). The remaining parameters are chosen to propagate that norm throughout layers using averages. This amounts to the following choices:

1. Set all σ_i to be activations such that $\sigma_i'(u) \approx 1$ for u large enough, e.g., (leaky) ReLU or softplus.
2. Introduce an initialization layer, $z_0 = q_{M,m}(x)\mathbf{1}$, itself initialized with $M = I$ and $m = \mathbf{0}_d$.
3. Initialize all matrices W_i^z to $\approx \mathbf{1}_{d_2,d_1}/d_1$, where d_1, d_2 are the dimensions of these matrices.
4. Initialize all matrices W_i^x to ≈ 0.
5. Initialize biases b_i to $s\mathbf{1}$, where s is a large enough value s so that $\sigma_i'(s) \approx 1$.

Gaussian Initialization The second approach can be used to initialize an ICNN so that its gradient mimics the affine transport between the Gaussian approximations of μ and ν. To this end, we follow all of the steps outlined above, except for step 2 where the quadratic layer $q_{M,m}$ is initialized instead with $M = A$ and $m = \mathbf{m}_1 - (A^T A)^{-1}\mathbf{m}_2$ using notations in (5), (6), (7), where $\mathbf{m}_1, \mathbf{m}_2, \Sigma_1, \Sigma_2$

Table 1: Evaluation of drug effect predictions from control cells to cells treated with drug Givinostat when conditioning on various covariates influencing cellular responses such as drug dosage and cell type. Results are reported based on MMD and the ℓ_2 distance between perturbation signatures of marker genes in the 1000 dimensional gene expression space.

Method	**Conditioned on Drug Dosage**				**Conditioned on Cell Line**	
	In-Sample		Out-of-Sample		In-Sample	
	MMD	ℓ_2(PS)	MMD	ℓ_2(PS)	MMD	ℓ_2(PS)
CPA (Lotfollahi et al., 2021)	0.1502 ± 0.0769	2.47 ± 2.89	0.1568 ± 0.0729	2.65 ± 2.75	0.2551 ± 0.006	2.71 ± 1.51
ICNN OT (Makkuva et al., 2020)	0.0365 ± 0.0473	2.37 ± 2.15	0.0466 ± 0.0479	2.24 ± 2.39	0.0206 ± 0.0109	1.16 ± 0.75
CondOT (Identity initialization)	0.0111 ± 0.0055	0.63 ± 0.09	0.0374 ± 0.0052	2.02 ± 0.10	0.0148 ± 0.0078	0.39 ± 0.06
CondOT (Gaussian initialization)	0.0128 ± 0.0081	0.60 ± 0.11	0.0325 ± 0.0062	1.84 ± 0.14	0.0146 ± 0.0074	0.41 ± 0.07

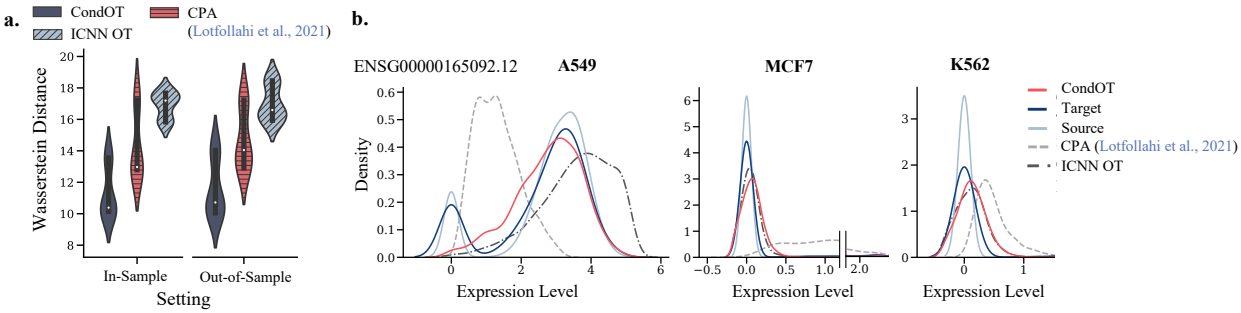

Figure 4: **a.** Predictive performance of CondOT and baselines w.r.t. the entropy-regularized Wasserstein distance on drug dosages *in-sample*, i.e., seen during training, and *out-of-sample*, i.e., unseen during training. **b.** Marginal distributions of observed source and target distributions, as well as predictions on perturbed distributions by CondOT and baselines of an exemplary gene across different cell lines. Predicted marginals of each method should match the marginal of the target population.

are replaced by the empirical mean and covariances of μ and ν. Throughout the experiments, we use the Gaussian and identity initialization. Further comparisons between the vanilla initialization and those introduced in this work can be found in § C.1 (Fig. 8).

PICNN Initialization Recall for convenience that a K-layer PICNN architecture reads:

$$\begin{aligned} u_{k+1} &= \tau_k \left(V_k u_k + v_k\right) \\ z_{k+1} &= \sigma_k \left(W_k^z \left(z_k \circ [W_k^{zu} u_k + b_k^z]_+\right) + W_k^x \left(x \circ (W_k^{xu} u_k + b_k^x)\right) + W_k^u u_k + b_k^u\right) \\ \psi_\theta(x, c) &= z_K. \end{aligned}$$

In their original form (Amos et al., 2017, Eq. 3), PICNNs are initialized by setting $u_0 = c$ and $z_0 = \mathbf{0}$ to a zero vector of suitable size. Intuitively, the hidden states u_k act as context-dependent modulators, whereas vectors z_k propagate, layer after layer, a collection of convex functions in x that are iteratively refined, while retaining the property that they are each convex in x. A reasonable initialization for a PICNN that is provided a context vector c is that if $c \approx c_j$ (where j is in the training set), one has that $\nabla_1 \psi_{\theta_0}(\cdot, c)$ maps approximately μ_j to ν_j, which can be obtained by having $\psi_{\theta_0}(\cdot, c)$ mimic the closed-form Brenier potential between the Gaussian approximations of μ_j, ν_j. Alternatively, one may also default to an identity initialization as discusses above. To obtain either behavior, we make the following modifications, and refer to the illustration in Fig. 3:

1. The modulator $u_0(c) = \text{softmax}(c^T M)$, where C is initialized as $M = [c_j]_j$, and $V_i = I, v_i = \mathbf{0}$.
2. $z_0 = [q_{M_j, m_j}(x)]_j$, where weight matrices and bias (M_j, m_j) are either initialized to $(I, \mathbf{0})$ or as (A_j, ω_j) recovered by solving the Gaussian affine map from μ_j to ν_j using (7).
3. Modulator u_0 is passed directly to hidden state upon first iteration $W_0^{zu} = I, b_0^z = 0$.
4. All subsequent matrices W_k^z are initialized to $\approx \mathbf{1}_{d_2, d_1}/d_1$, where d_1, d_2 are their dimensions,
5. W_k^x and W_k^{xu} are ≈ 0, the biases $b_k^z \approx \mathbf{1}$, $b_k^u \approx \mathbf{0}$.

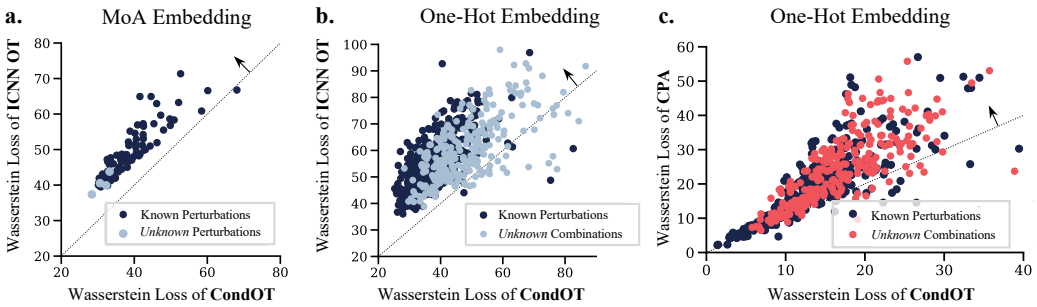

Figure 5: Comparison between **a.** CONDOT and ICNN OT (Makkuva et al., 2020) based on embedding $\mathcal{E}_{\text{moa}}$ **b.** as well as $\mathcal{E}_{\text{ohe}}$, and **c.** CONDOT and CPA (Lotfollahi et al., 2021) based on embedding $\mathcal{E}_{\text{ohe}}$ on *known* and *unknown* perturbations or combinations. Results above the diagonal suggest higher predictive performance of CONDOT.

5 Evaluation

Biological cells undergo changes in their molecular profiles upon chemical, genetic, or mechanical perturbations. These changes can be measured using recent technological advancements in high-resolution multivariate single-cell biology. Measuring single cells in their unperturbed or perturbed state requires, however, to destroy them, resulting in populations μ and ν that are unpaired. The relevance of OT to that comes from its ability to resolve such ambiguities through OT maps, that hold the promises of a better understanding of human health and disease. We consider various high-dimensional problems arising from this scenario to evaluate the performance of CONDOT (§ 3) versus other baselines.

5.1 Population Dynamics Conditioned on *Scalars*

Upon application of a molecular drug, the state of each cell x_i of the unperturbed population is altered, and observed in population ν. Molecular drugs are often applied at different dosage levels t, and the magnitude of changes in the gene expression profiles of single cells highly correlates with that dosage. We seek to learn a global, parameterized transport map $\mathcal{T}_\theta$ sensitive to that dosage.We evaluate our method on the task of inferring single-cell perturbation responses to the cancer drug Givinostat, a histone deacetylase inhibitor with potential anti-inflammatory, anti-angiogenic, and antineoplastic activities (Srivatsan et al., 2020), applied at different dosage levels, i.e., $t \in \{10\,\text{nM}, 100\,\text{nM}, 1,000\,\text{nM}, 10,000\,\text{nM}\}$. The dataset contains $3,541$ cells described with the gene expression levels of $1,000$ highly-variable genes. In a first experiment, we measure how well CONDOT captures the drug effects at different dosage levels via distributional distances such as MMD (Gretton et al., 2012) and the ℓ_2-norm between the corresponding perturbation signatures (PS), as well as the entropy-regularized Wasserstein distance (Cuturi, 2013). We compute the metrics on 50 marker genes, i.e., genes mostly affected upon perturbation. For more details on evaluation metrics, see § E.2. To put CONDOT's performance into perspective, we compare it to current state-of-the-art baselines (Lotfollahi et al., 2021) as well as parameterized Monge maps without context variables (ICNN OT), see § E.1. As visible in Table 1 and Fig. 4a, CONDOT achieves consistently more accurate predictions of the target cell populations at different dosage levels than OT approaches that cannot utilize context information, demonstrated through a lower average loss and a smaller variance. This becomes even more evident when moving to the setting where the population has been trained only on a subset of dosages and we test CONDOT on *out-of-sample* dosages. Table 1 and Fig. 4a demonstrate that CONDOT is able to generalize to previously *unknown* dosages, thus learning to interpolate the perturbation effects from dosages seen during training. For further analysis, we refer the reader to § E (see Fig. 9 and 10).

5.2 Population Dynamics Conditioned on *Covariates*

Molecular processes are often highly dependent on additional covariates that steer experimental conditions, and which are not present in the features measures in population μ or ν. This can be, for instance, factors such as different cell types clustered within the populations. When the model can only

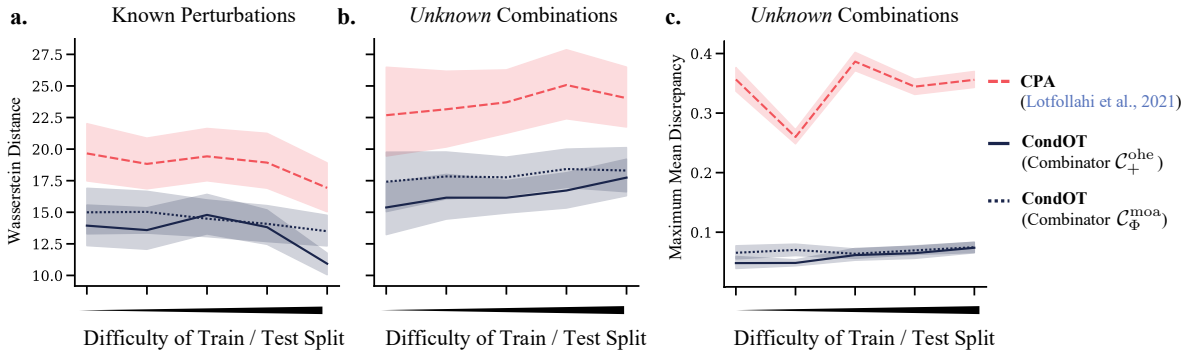

Figure 6: Predictive performance for **a.** known perturbations, **b.** unknown perturbations in combination w.r.t. regularized Wasserstein distance and **c.** MMD over different train / test splits of increasing difficulty for baseline CPA as well as CONDOT with different combinators $\mathcal{C}_{+}^{\text{ohe}}$ and $\mathcal{C}_{\Phi}^{\text{moa}}$. For more details on the dataset splits, see §D.2.

be conditioned w.r.t. a small and *fixed* set of metadata information, such as cell types, it is sufficient to encode these contexts using a one-hot embedding module $\mathcal{E}_{\text{ohe}}$. To illustrate this problem, we consider cell populations comprising three different cell lines (A549, MCF7, and K562). As visible in Table 1, CONDOT outperforms current baselines which equally condition on covariate information such as CPA (Lotfollahi et al., 2021), assessed through various evaluation metrics. Figure 4b displays a gene showing highly various responses towards the drug Givinostat dependent on the cell line. CONDOT captures the distribution shift from control to target populations consistently across different cell lines.

5.3 Population Dynamics Conditioned on *Actions*

To recommend personalized medical procedures for patients, or to improve our understanding of genetic circuits, it is key to be able to predict the outcomes of novel perturbations, arising from combinations of drugs or of genetic perturbations. Rather than learning individual maps T_θ^a predicting the effect of individual treatments, we aim at learning a global map $\mathcal{T}_\theta$ which, given as input the unperturbed population μ as well as the action a of interest, predicts the cell state perturbed by a. Thanks to its modularity, CONDOT can not only learn a map T_θ for all actions *known* during training, but also to generalize to *unknown* actions, as well as potential *combinations* of actions. We will discuss all three scenarios below.

5.3.1 *Known* Actions

In the following, we analyze CONDOT's ability to accurately predict phenotypes of genetic perturbations based on single-cell RNA-sequencing pooled CRISPR screens (Norman et al., 2019; Dixit et al., 2016), comprising $98,419$ single-cell gene expression profiles with 92 different genetic perturbations, each cell measured via a $1,500$ highly-variable gene expression vector. As, in a first step, we do not aim at generalizing beyond perturbations encountered during training, we utilize again a one-hot embedding $\mathcal{E}_{\text{ohe}}$ to condition $\mathcal{T}_\theta$ on each perturbation a. We compare our method to other baselines capable of modeling effects of a large set of perturbations such as CPA (Lotfollahi et al., 2021). Often, the effect of genetic perturbations are subtle in the high-dimensional gene expression profile of single cells. Using ICNN-parameterized OT maps without context information, we can thus assess the gain in accuracy of predicting the perturbed target population by incorporating context-awareness over simply predicting an average perturbation effect. Figure 5a and b demonstrate that compared to OT ablation studies, Fig. 5c and Fig. 6a for the current state-of-the-art method CPA (Lotfollahi et al., 2021). Compared to both, CONDOT captures the perturbation responses more accurately w.r.t. the Wasserstein distance.

5.3.2 *Unknown* Actions

With the emergence of new perturbations or drugs, we aim at inferring cellular responses to settings not explored during training. One-hot embeddings, however, do not allow us to model *unknown* perturbations. This requires us to use an embedding $\mathcal{E}$, which can provide us with a representation of an

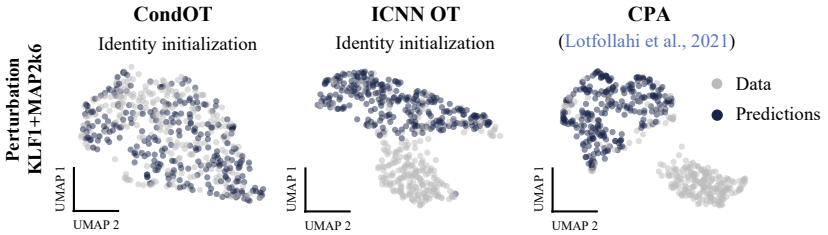

Figure 7: UMAP embeddings of cells perturbed by the combination KLF1+MAP2K6 (gray) and predictions of CONDOT (ours), ICNN OT (Makkuva et al., 2020), and CPA (Lotfollahi et al., 2021) (blue). While CONDOT covers the space of observed perturbed cells, the baselines fail to capture subpopulations.

unknown action a'. As genetic perturbations further have no meaningful embeddings as, for example, molecular fingerprints for drugs, we resort to mode-of-action embeddings introduced in § 3.2. Assuming marginal sample access to all individual perturbations, we compute a multidimensional scaling (MDS)-based embedding from pairwise Wasserstein distances between individual target populations, such that perturbations with similar effects are closely represented. For details, see § E. As current state-of-the-art methods are restricted to modeling perturbations via one-hot encodings, we compare our method to ICNN OT only. As displayed in Fig. 5a, CONDOT accurately captures the response of *unknown* actions (BAK1, FOXF1, MAP2K6, MAP4K3), which were not seen during training, at a similar Wasserstein loss as perturbation effects seen during training. For more details, see § E.

5.3.3 Actions in Combination

While experimental studies can often measure perturbation effects in biological systems in isolation, the combinatorial space of perturbations in composition is too large to capture experimentally. Often, however, combination therapies are cornerstones of cancer therapy (Mokhtari et al., 2017). In the following, we test different combinator architectures to predict genetic perturbations in combination from single targets. Similarly to Lotfollahi et al. (2021), we can embed combinations by adding individual one-hot encodings of single perturbations (i.e., $\mathcal{C}_{+}^{\text{ohe}}$). In addition, we parameterize a combinator via a permutation-invariant deep set, as introduced in § 3.2, based on mode-of-action embeddings of individual perturbations (i.e., $\mathcal{C}_{\Phi}^{\text{moa}}$). We split the dataset into train / test splits of increasing difficulty (details on the dataset splits in §D.2). Initially containing all individual perturbations as well as some combinations, the number of perturbations seen in combination during training decreases over each split. For more details, see § E. We compare different combinators to ICNN OT (Fig. 5b) and CPA (Lotfollahi et al., 2021) (Fig. 5c, Fig. 6b, c). While the performance drops compared to inference on *known* perturbations (Fig. 6a) and decreases with increasing difficulty of the train / test split, CONDOT outperforms all baselines. When embedding these high-dimensional populations in a low-dimensional UMAP space (McInnes et al., 2018), one can see that CONDOT captures the entire perturbed population, while ICNN OT and CPA fail in capturing certain subpopulations in the perturbed state (see Fig. 7 and 11).

6 Conclusion

We have developed the CONDOT framework that is able to infer OT maps from not only one pair of measures, but many pairs that come labeled with a context value. To ensure that CONDOT encodes optimal transports, we parameterize it as a PICNN, an input-convex NN that modulates the values of its weights matrices according to a sequence of feature representations of that context vector. We showcased the generalization abilities of CONDOT in the extremely challenging task of predicting outcomes for unseen combinations of treatments. These abilities and PICNN more generally hold several promises, both as an augmentation of the OTT toolbox (Cuturi et al., 2022), and for future applications of OT to single cell genomics.

References

D. Alvarez-Melis, Y. Schiff, and Y. Mroueh. Optimizing Functionals on the Space of Probabilities with Input Convex Neural Networks. *arXiv Preprint arXiv:2106.00774*, 2021.

B. Amos, L. Xu, and J. Z. Kolter. Input Convex Neural Networks. In *International Conference on Machine Learning (ICML)*, volume 34, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*. PMLR, 2017.

Y. Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305, 1987.

C. Bunne, S. G. Stark, G. Gut, J. S. del Castillo, K.-V. Lehmann, L. Pelkmans, A. Krause, and G. Ratsch. Learning Single-Cell Perturbation Responses using Neural Optimal Transport. *bioRxiv*, 2021.

C. Bunne, L. Meng-Papaxanthos, A. Krause, and M. Cuturi. Proximal Optimal Transport Modeling of Population Dynamics. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 25, 2022.

Y. Chandak, G. Theocharous, J. Kostas, S. Jordan, and P. Thomas. Learning Action Representations for Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2019.

Y. Chen, Y. Shi, and B. Zhang. Optimal Control Via Neural Networks: A Convex Approach. In *International Conference on Learning Representations (ICLR)*, 2019.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE, 2005.

M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

M. Cuturi and G. Peyré. Semidual Regularized Optimal Transport. *SIAM Review*, 60(4):941–965, 2018.

M. Cuturi, L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul. Optimal Transport Tools (OTT): A JAX Toolbox for all things Wasserstein. *arXiv Preprint arXiv:2201.12324*, 2022.

J. De Leeuw and P. Mair. Multidimensional Scaling Using Majorization: SMACOF in R. *Journal of Statistical Software*, 31, 2009.

A. Dixit, O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R. Raychowdhury, et al. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7), 2016.

J. Fan, A. Taghvaei, and Y. Chen. Scalable computations of wasserstein barycenter via input convex neural networks. In *International Conference on Machine Learning*, pages 1571–1581. PMLR, 2021.

M. Gelbrich. On a Formula for the L^2 Wasserstein Metric between Measures on Euclidean and Hilbert Spaces. *Mathematische Nachrichten*, 147(1), 1990.

A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. Sample Complexity of Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, 2019.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

D. Ha, A. Dai, and Q. V. Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

T. Hashimoto, D. Gifford, and T. Jaakkola. Learning Population-Level Diffusions with Generative Recurrent Networks. In *International Conference on Machine Learning (ICML)*, volume 33, 2016.

L. Hetzel, S. Böhm, N. Kilbertus, S. Günnemann, M. Lotfollahi, and F. Theis. Predicting single-cell perturbation responses for unseen drugs. *arXiv preprint arXiv:2204.13545*, 2022.

C.-W. Huang, R. T. Q. Chen, C. Tsirigotis, and A. Courville. Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

L. Jacob, J. She, A. Almahairi, S. Rajeswar, and A. Courville. W2GAN: Recovering an Optimal Transport Map with a GAN. In *arXiv Preprint*, 2018.

H. Janati, T. Bazeille, B. Thirion, M. Cuturi, and A. Gramfort. Multi-subject meg/eeg source imaging with sparse multi-task regression. *NeuroImage*, 220:116847, 2020.

L. Kantorovich. On the transfer of masses (in Russian). In *Doklady Akademii Nauk*, volume 37, 1942.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

A. Korotin, V. Egiazarian, A. Asadulaev, A. Safin, and E. Burnaev. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2020.

A. Korotin, L. Li, A. Genevay, J. M. Solomon, A. Filippov, and E. Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34, 2021.

S. Kummar, H. X. Chen, J. Wright, S. Holbeck, M. D. Millin, J. Tomaszewski, J. Zweibel, J. Collins, and J. H. Doroshow. Utilizing targeted cancer therapeutic agents in combination: novel approaches and urgent requirements. *Nature Reviews Drug discovery*, 9(11), 2010.

M. Lotfollahi, F. A. Wolf, and F. J. Theis. scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8), 2019.

M. Lotfollahi, M. Naghipourfar, F. J. Theis, and F. A. Wolf. Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics*, 36, 2020.

M. Lotfollahi, A. K. Susmelj, C. De Donno, Y. Ji, I. L. Ibarra, F. A. Wolf, N. Yakubova, F. J. Theis, and D. Lopez-Paz. Compositional perturbation autoencoder for single-cell response modeling. *bioRxiv*, 2021.

A. Makkuva, A. Taghvaei, S. Oh, and J. Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning (ICML)*, volume 37, 2020.

R. J. McCann. A Convexity Principle for Interacting Gases. *Advances in Mathematics*, 128(1), 1997.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv Preprint*, 2018.

A. Mead. Review of the Development of Multidimensional Scaling Methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1), 1992.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations (ICLR), Workshop Track*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013b.

T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2013c.

R. B. Mokhtari, T. S. Homayouni, N. Baluch, E. Morgatskaya, S. Kumar, B. Das, and H. Yeger. Combination therapy in combating cancer. *Oncotarget*, 8(23):38022, 2017.

P. Mokrov, A. Korotin, L. Li, A. Genevay, J. Solomon, and E. Burnaev. Large-Scale Wasserstein Gradient Flows. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

T. M. Norman, M. A. Horlbeck, J. M. Replogle, A. Y. Ge, A. Xu, M. Jost, L. A. Gilbert, and J. S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455), 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2011.

G. Peyré and M. Cuturi. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11(5-6), 2019. ISSN 1935-8245.

N. Prasad, K. Yang, and C. Uhler. Optimal Transport using GANs for Lineage Tracing. *arXiv preprint arXiv:2007.12098*, 2020.

A. Rambaldi, C. M. Dellacasa, G. Finazzi, A. Carobbio, M. L. Ferrari, P. Guglielmelli, E. Gattoni, S. Salmoiraghi, M. C. Finazzi, S. Di Tollo, et al. A pilot study of the Histone-Deacetylase inhibitor Givinostat in patients with JAK2V617F positive chronic myeloproliferative neoplasms. *British journal of haematology*, 150(4):446–455, 2010.

D. Rezende and S. Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning (ICML)*, 2015.

J. Richter-Powell, J. Lorraine, and B. Amos. Input convex gradient networks. *arXiv preprint arXiv:2111.12187*, 2021.

D. Rogers and M. Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5), 2010.

Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang. Self-Supervised Graph Transformer on Large-Scale Molecular Data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

F. Santambrogio. Optimal Transport for Applied Mathematicians. *Birkhäuser, NY*, 55(58-63):94, 2015.

G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, et al. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4), 2019.

S. R. Srivatsan, J. L. McFaline-Figueroa, V. Ramani, L. Saunders, J. Cao, J. Packer, H. A. Pliner, D. L. Jackson, R. M. Daza, L. Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473), 2020.

G. Tennenholtz and S. Mannor. The Natural Language of Actions. In *International Conference on Machine Learning (ICML)*, 2019.

C. Villani. *Topics in Optimal Transportation*, volume 58. American Mathematical Soc., 2003.

F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1), 2018.

K. D. Yang and C. Uhler. Scalable Unbalanced Optimal Transport using Generative Adversarial Networks. *International Conference on Learning Representations (ICLR)*, 2019.

K. D. Yang, K. Damodaran, S. Venkatachalapathy, A. C. Soylemezoglu, G. Shivashankar, and C. Uhler. Predicting cell lineages using autoencoders and optimal transport. *PLoS Computational Biology*, 16(4), 2020.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code to reproduce the datasets will be available upon publication of this work. The datasets are public and the sources are referenced in the paper § D. The data processing pipeline and further experimental details such as hyperparameter or algorithmic decisions are outlined in § E.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The datasets and corresponding splits are described in § D. The hyperparameter setup in all experiments is identical and outlined in § E.4. The baseline configurations are described in § E.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Background

Primal and Dual Optimal Transport The primal optimal transport problem (POT) was introduced in (1), and quickly linked in our background section § 2 to the dual optimal transport problem (DOT) (2). We provide for completeness an intermediary step to facilitate understanding, which works in the case where $p = 2$, and explain why the optimal transport map T can also be recovered via the dual optimal transport problem. Introduced by Kantorovich in 1942, the dual formulation is a constrained concave maximization problem defined as

$$W(\mu, \nu) = \sup_{(f,g)\in\Psi_c} \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x) + \int_{\mathcal{Y}} g(y)\mathrm{d}\nu(y),$$

where the set of admissible potentials is $\Psi_c := \{(f, g) \in L^1(\mu) \times L^1(\nu) : f(x) + g(y) \leq \frac{1}{2}\|x - y\|,$ $\forall (x, y) d\mu \otimes d\nu$ a.e.$\}$ (Villani, 2003, Theorem 1.3). The machinery of c-transforms (Santambrogio, 2015, §1.3 1) can be used to simplify that problem. When the cost c is half the square Euclidean distance as considered here, this results in a simpler, so-called semi-dual problem (Cuturi and Peyré, 2018) that only involves a single potential function.

$$f^\star_{\mu,\nu} := \arg\sup_{f \text{ convex}} \int_{\mathbb{R}^d} f^* \mathrm{d}\mu + \int_{\mathbb{R}^d} f \mathrm{d}\nu \; = \psi^\star_{\mu,\nu}\,. \tag{9}$$

The optimal convex potential function ψ is then related to the optimal dual potential $f^\star$ expressed above, through the identity $\psi = f^\star_{\mu,\nu}$.

Neural Optimal Transport Learning optimal transport problems based on neural networks is at the core of many machine learning applications, including normalizing flows (Rezende and Mohamed, 2015; Huang et al., 2021) and generative models (Arjovsky et al., 2017; Genevay et al., 2019). Directly parameterizing the doubly-stochastic matrix T of the primal optimal transport (1) as done in previous work (Jacob et al., 2018; Yang and Uhler, 2019; Prasad et al., 2020) has been shown to yield an unstable and thus difficult to solve optimization problem (Makkuva et al., 2020, Table 1). We thus follow previous work (Makkuva et al., 2020; Bunne et al., 2022; Korotin et al., 2020; Alvarez-Melis et al., 2021) and instead learn map T via the convex Brenier potential ψ connected to the primal and dual optimal transport problem as outlined above. We parameterize the convex function ψ via convex neural architectures (see § 2), which can thus be used in two contexts, either to model the Brenier potential, or to model a dual function. Both lead to the same results since the Brenier potential $\psi^\star_{\mu,\nu}$ is equal to the optimal dual potential associated with the second measure ν, $f^\star_{\mu,\nu}$, as described above and in §2 around (2).

B The CondOT Algorithm

CondOT provides a generalized approach that from *labeled* pairs of measures $\{(c_i, (\mu_i, \nu_i))\}_i$ infers a *global* parameterized conditional Monge map $\mathcal{T}_\theta$. This is achieved by jointly learning an the embedding module $\mathcal{E}_\phi$, a combinator module $\mathcal{C}_\Phi$, as well as transport map $\mathcal{T}_\theta$. The algorithmic procedure is outlined in Algorithm 1. We describe CondOT's modules as well as their parameterization in detail in §E.3. In the following, we will cover in more depth algorithmic approaches on how to learn transport map $\mathcal{T}_\theta$. Several approaches have been proposed on inferring transport map $\mathcal{T}_\theta$ from paired source and target populations, including the primal (1) or dual optimal transport problem (2).

A possible approach to learn our model could consist in minimizing a primal OT problem. In that case, we can learn $\mathcal{T}_\theta$ via the gradient of the Brenier potential parameterized via a PICNN, i.e., $\mathcal{T}_\theta = \nabla\psi^*_\theta = \nabla_1 \text{PICNN}_\theta$. The PICNN is then trained using the entropy-regularized Wasserstein distance (15) between the predictions $\hat{\nu} = \nabla\psi^*_\theta{}_\sharp\,\mu = \nabla_1 \text{PICNN}_\theta(\cdot, c)_\sharp\,\mu$ given source samples μ and condition c and the observed target population ν as a loss function, i.e.,

$$\ell_{\text{POT}}(\mu, \nu, c; \theta) = W_\varepsilon(\nabla_1 \text{PICNN}_\theta(\cdot, c)_\sharp\,\mu, \nu). \tag{10}$$

Throughout this work, we choose a different route and propose instead to learn $\mathcal{T}_\theta$ via the dual optimal transport problem. We consider the strategy proposed by Makkuva et al. (2020) and utilized by Bunne et al. (2021) in the context of single-cell perturbation analyses. $\mathcal{T}_\theta$ is then parameterized via the pair of dual potentials f and g, which themselves are defined by a pair of PICNNs $g : \text{PICNN}_{\theta_g}(\cdot, c)$

Algorithm 1 CONDOT Algorithm.

Input: Dataset $\mathcal{D} = \{\mu_i, \nu_i, c_i\}_{i=0}^{N}$ of N pairs of populations before μ_i and after transport ν_i connected to a context c_i, θ^0 transport map $\mathcal{T}$ parameter initialization, ϕ^0 embedding $\mathcal{E}$ parameter initialization, Φ^0 embedding $\mathcal{C}$ parameter initialization, learning rates lr_θ, lr_ϕ and lr_Φ, and flag which `loss` function to use. In the case of the dual, we have $\theta = (\theta_f, \theta_g)$ parameterizing the dual potentials f and g and `train_freq_f` specifies the training frequency of dual potential f.

Output: Transport map $\mathcal{T}_\theta$, embedding $\mathcal{E}_\phi$, and combinator $\mathcal{C}_\Phi$.

$\theta, \phi, \Phi \leftarrow \theta^0, \phi^0, \Phi^0$
for $\{\mu_i, \nu_i, c_i\} \in \mathcal{D}$ **do**
  `# Split (combination) context` c_i `into individual contexts.`
  $c_i^1, c_i^2, \ldots, c_i^k = c_i$
  $\hat{c}_i = \mathcal{C}_\Phi(\mathcal{E}_\phi(c_i^1), \mathcal{E}_\phi(c_i^2), \ldots, \mathcal{E}_\phi(c_i^k))$
  if `setting` == 'dual' **then**
    if i % `train_freq_f` == 0 **then**
      $\ell \leftarrow \ell_{\text{DOT}}^f(\mu_i, \nu_i, \hat{c}_i; \theta_f)$ (13)
    else
      $\ell \leftarrow \ell_{\text{DOT}}^g(\mu_i, \nu_i, \hat{c}_i; \theta_g)$ (14)
  else
    $\ell \leftarrow \ell_{\text{POT}}(\mu_i, \nu_i, \hat{c}_i; \theta)$ (10)
  `# Jointly optimize parameters` θ, ϕ, Φ `given loss` ℓ.
  $\theta \leftarrow \theta - \text{lr}_\theta \times \nabla_\theta \ell$
  $\phi \leftarrow \phi - \text{lr}_\phi \times \nabla_\phi \ell$
  $\Phi \leftarrow \Phi - \text{lr}_\Phi \times \nabla_\Phi \ell$
return

and $f : \text{PICNN}_{\theta_f}(\cdot, c)$ such that $\hat{\nu} = \nabla g_\sharp \mu = \nabla_1 \text{PICNN}_{\theta_g}(\cdot, c)\sharp\mu$ is approximately ν, as well as $\hat{\mu} = \nabla f_\sharp \nu = \nabla_1 \text{PICNN}_{\theta_f}(\cdot, c)_\sharp \nu$ is approximately μ on a labeled observation $((\mu, \nu), c)$ with parameters $\theta = (\theta_g, \theta_f)$. In order to optimize the pair of PICNNs, which parameterize the two dual functions, Makkuva et al. (2020) derive an approximate formulation of (2). First, Villani (2003, Theorem 2.9) rephrases (2) over the pair of dual potentials (f, g) to

$$W(\mu, \nu) = \underbrace{\frac{1}{2}\mathbb{E}\left[\|x\| + \|y\|\right]}_{\mathcal{C}_{\mu,\nu}} - \inf_{f \text{ convex}} \mathbb{E}_\mu[f(X)] + \mathbb{E}_\nu\left[f^*(Y)\right], \tag{11}$$

where $f^*(y) = \sup_x \langle x, y\rangle - f(x)$ is f's convex conjugate. In a second step, Makkuva et al. (2020) derive a min-max formulation by approximating the convex conjugate in (11) via

$$W(\mu, \nu) = \sup_{\substack{f \text{ convex} \\ f^* \in L^1(\nu)}} \inf_{g \text{ convex}} \mathcal{C}_{\mu,\nu} - \underbrace{\mathbb{E}_\mu[f(x)] - \mathbb{E}_\nu[\langle y, \nabla g(y)\rangle - f(\nabla g(y))]}_{\mathcal{V}_{\mu,\nu}(f,g)}, \tag{12}$$

and by relaxing the constraints on g. Thus, the dual potentials f and g can be learned via an alternate min-max optimization problem with loss functions

$$\ell_{\text{DOT}}^f(\mu, \nu, c; \theta_f) = \mathbb{E}_{x\sim\mu}[\text{PICNN}_{\theta_g}(x, c)] - \mathbb{E}_{y\sim\nu}[\text{PICNN}_{\theta_f}(\nabla\text{PICNN}_{\theta_g}(y, c), c)], \text{ and} \tag{13}$$

$$\ell_{\text{DOT}}^g(\mu, \nu, c; \theta_g) = -\mathbb{E}_{y\sim\nu}[\langle y, \nabla\text{PICNN}_{\theta_g}(y, c)\rangle - \text{PICNN}_{\theta_f}(\nabla\text{PICNN}_{\theta_g}(y, c), c)]. \tag{14}$$

For more details, see Makkuva et al. (2020); Korotin et al. (2021).

Thus, dependent on the strategy chosen, $\mathcal{T}_\theta$ is parameterized via a single or a pair of PICNN. Each network takes as input the source distribution μ —in which it is input convex— as well as an embedded context variable $\hat{c}$, returned by combinator $\mathcal{C}_\Phi$ and embedding module $\mathcal{E}_\phi$. Parameters of all three modules are jointly trained based on the derived optimal transport loss $\ell = \{\ell_{\text{POT}}, \ell_{\text{DOT}}\}$, which measures how close predicted target cells $\hat{\nu}$ are from the observed target population ν, given source population μ and context c as inputs.

C Additional Experimental Results

C.1 Comparison of Initialization Methods

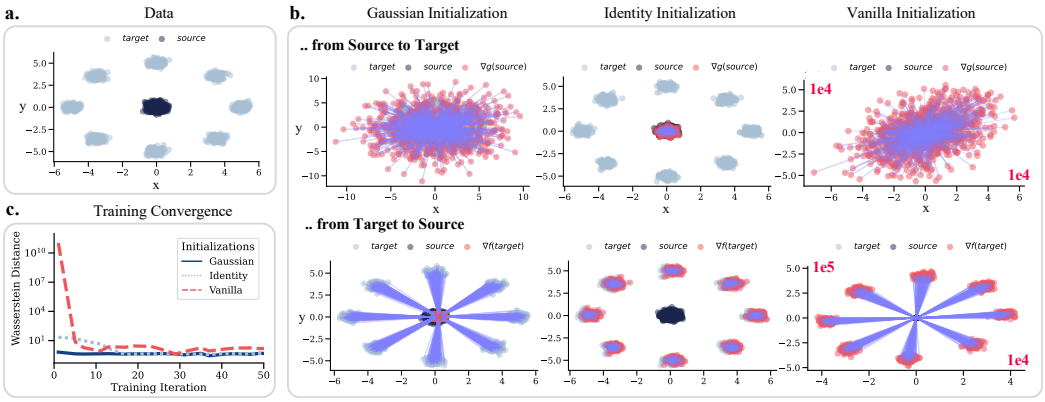

Figure 8: Comparison of ICNN initialization methods on a **a.** synthetic dataset containing source (dark blue) and target samples (light blue). **b.** Predicted samples (red) are obtained by transporting source samples (with dual potential g, first row) or target samples (with dual potential f, second row) to match the respective observations. The ICNNs are initialized such that they resemble a Gaussian closed-form approximation, the identity, or a random vanilla map (more details in § 4). Without any pretraining, the Gaussian initialization transports the samples to the Gaussian approximation of the respective target distribution. The identity initialization mimics the identity map and thus does not move the samples. The naïve vanilla initialization, on the other hand, starts with a solution far off from the target (i.e., values are in the range of $1e4$ or $1e5$). **c.** The chosen initialization strongly affects the convergence of the solution over the course of the training, here measured by the Wasserstein distance.

We conduct a simple experiment based on a synthetic dataset displayed in Fig. 8a, in which we seek to learn a mapping between source and target samples by parameterizing the dual potentials f and g with two ICNNs based on different initialization schemes (see Algorithm 1, § B). To showcase different initialization methods, we compare the initial predictions (at training iteration 0) of transported samples for the vanilla, the identity, and the Gaussian initialization (Fig. 8b). As the Gaussian initialization instantiates maps which transport source samples to the Gaussian approximation of the target samples, the initialization already captures well the source or target distribution using ICNN f or g, respectively (Fig. 8b, first column). The identity initialization, instead, configures maps which do not move the samples from the initial distribution (see Fig. 8b, second column). Both initialization schemes proposed in this work thus result in map parameterizations, which initially (before training) realize non-trivial and admissible Monge maps. The vanilla initialization, on the other hand, instantiates random maps, mapping the point far away from the source and target distribution, thus impeding fast and robust training (see Fig. 8, third column). We want to stress that this is achieved without costly and elaborate pretraining of the networks as proposed in Korotin et al. (2020, Appendix B).

The selected initialization method also strongly affects the convergence of the solution over the course of the training, which we monitor using the Wasserstein distance between observed and predicted target samples (see Fig. 8c). In this simple example, the Gaussian initialization is already close to the solution, thus the Wasserstein distance resembles the final solution already at the beginning of the training. The identity initialization similarly starts with a mapping closer to the solution compared to a random vanilla initialization, thus achieving fast and stable convergence of the min-max optimization problem (12). This experiment thus demonstrates that a proper initialization is not only crucial for fast convergence but also the overall robustness of training and result.

C.2 Out-of-Sample Predictions in Unknown Context

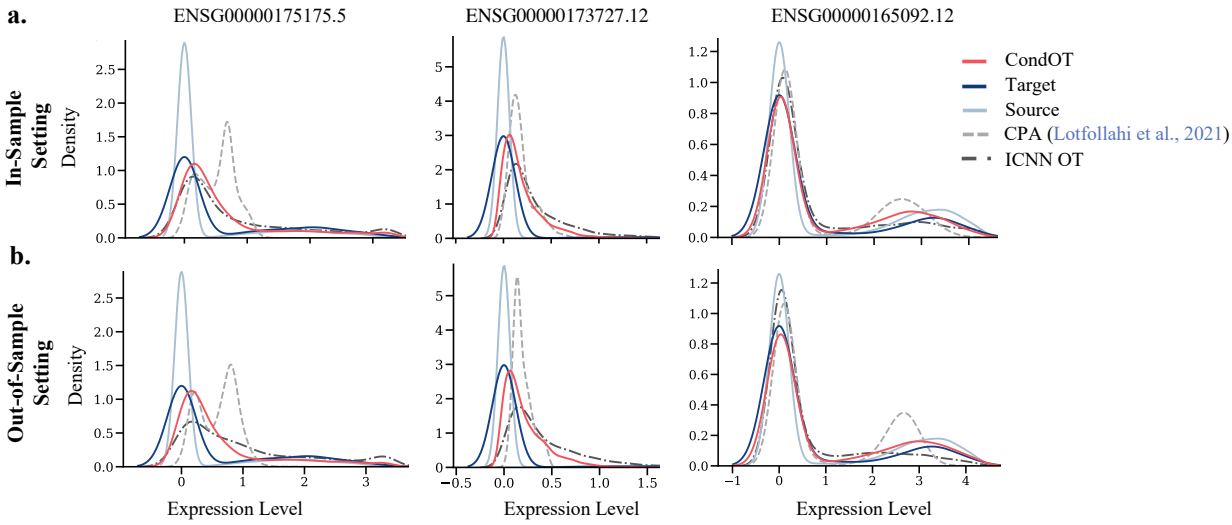

Figure 9: Marginal distributions of observed source (light blue) and target distributions (dark blue), as well as predictions on perturbed distributions by CONDOT (red) and baselines (gray) of different genes **a.** in the in-sample setting, where dosage 100nM was seen during training, and **b.** out-of-sample setting, where dosage 100nM was *not* seen during training. Predicted marginals of each method should match the marginal of the target population (dark blue. While the performance of CONDOT is consistent from the in-sample to the out-of-sample setting, both baselines show differences. These differences are subtle, however, only 3 out of $1,000$ genes are displayed.

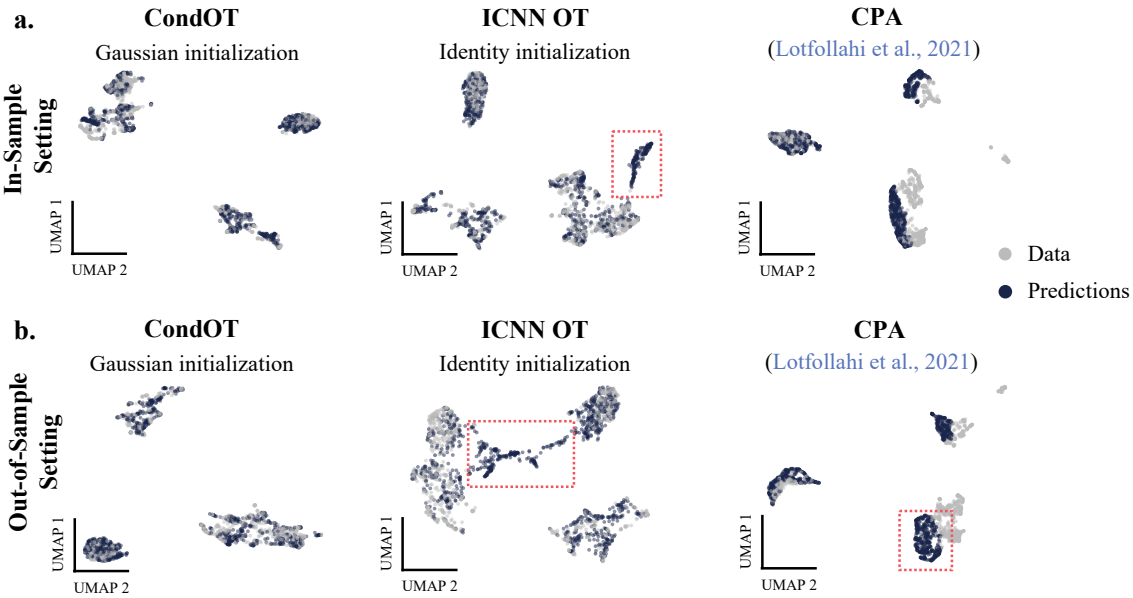

Figure 10: UMAP embeddings of cells perturbed by Givinostat dosage 100nM (gray) and predictions of CONDOT (ours), ICNN OT (Makkuva et al., 2020), and CPA (Lotfollahi et al., 2021) (blue). Contrary to the out-of-sample setting, the dosage 100nM was seen during training in the in-sample setting. While CONDOT covers the space of observed perturbed cells, the baselines fail to capture subpopulations (see red squares).

C.3 Predicting Unknown Perturbations and Perturbations in Combination

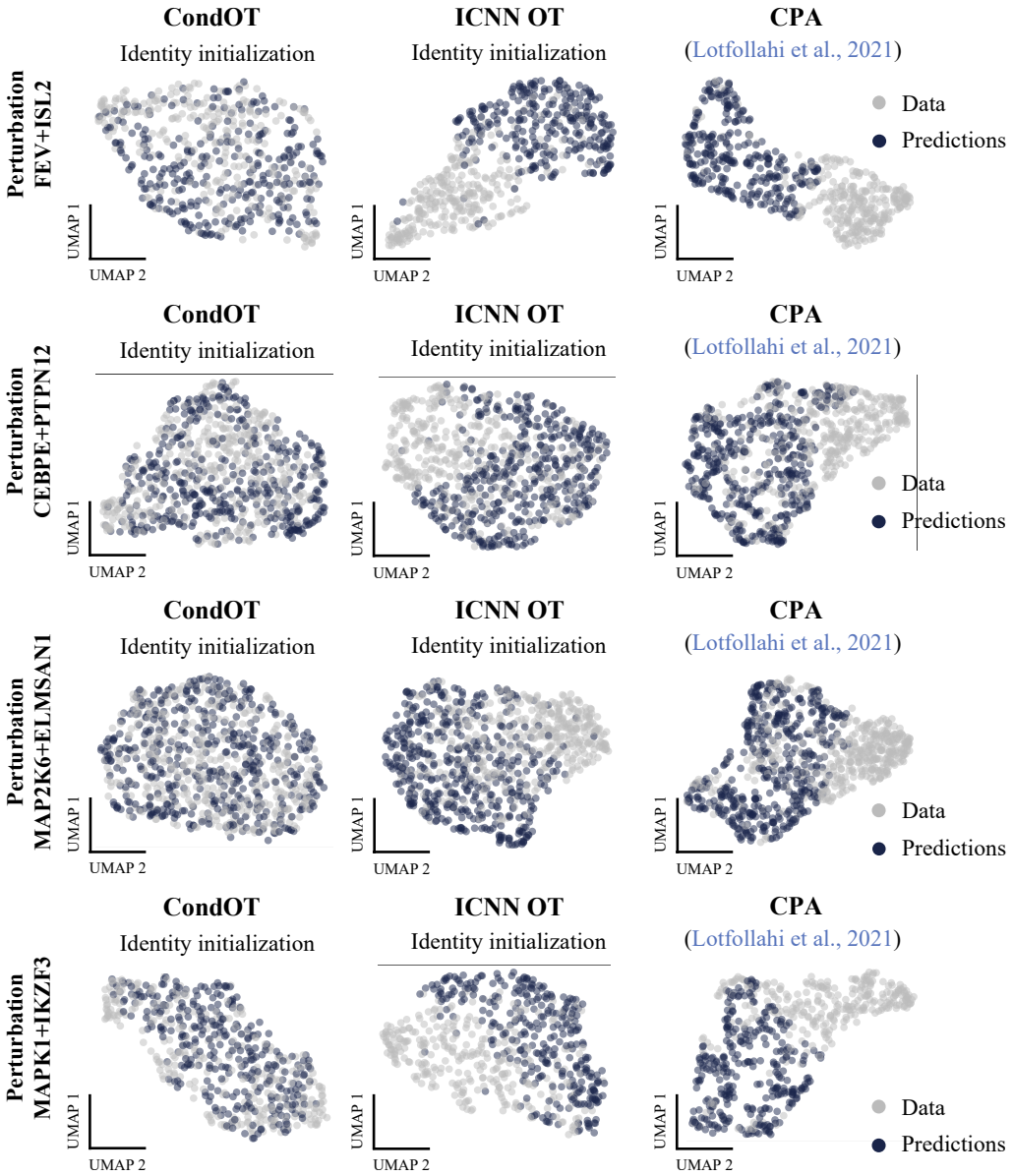

Figure 11: UMAP embeddings of cells perturbed by different combinations (grey) and predictions of CONDOT (ours), ICNN OT (Makkuva et al., 2020), and CPA (Lotfollahi et al., 2021) (blue). While CONDOT covers the space of observed perturbed cells, the baselines fail to capture subpopulations.

D Datasets

We evaluate CONDOT on different tasks, consisting of a pair of source μ and target measures ν, as well as context variables c of different nature. In particular, we consider single-cell datasets in which populations of single cells have been monitored with modern high-throughput methods such as single-cell RNA sequencing technologies. Characterizing and modeling perturbation responses at the level of single cells with access to *unpaired* populations of control and perturbed cells remains one of the grand challenges of biology. In this work, we consider the task of modeling molecular responses to cancer drugs with context variables being the drug's dosage (i.e., a scalar, § 5.1) as

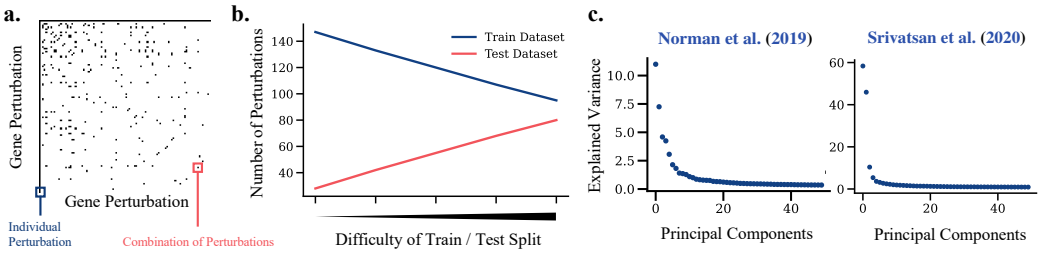

Figure 12: **a.** The indicator matrix of all individual perturbations as well as those perturbation pairs available in combination (black) in the dataset by Norman et al. (2019). **b.** Size of the different train / test splits of the dataset by Norman et al. (2019). The train set contains all single perturbations as well as a decreasing number of combinations with increasing difficulty of the data split. For more details, see §D.2. **c.** Explained variance of different datasets per principal component.

well as covariates such as different cancer cell lines present in the population (§ 5.1). Further, we study cellular responses to genetic perturbations, where we condition on the perturbation, i.e., action, chosen. Here we differentiate between settings where we encounter *known* actions (§ 5.3.1), *unknown* actions (§ 5.3.2), and action applied in combination (§ 5.3.3) during evaluation. In the following, we introduce the datasets in more depth, describe preprocessing steps, feature selection, and data splits.

D.1 ... by Srivatsan et al. (2020)

Cancer drugs reduce uncontrolled cell growth and proliferation by inhibiting DNA replication and RNA transcription as well as targeting proteins crucial for cancer progression. In doing so, they modulate downstream signaling cascades, affect cell growth and morphology, and alter gene expression profiles of single cells. Srivatsan et al. (2020) conduct a scRNA-seq–based phenotyping screen of transcriptional responses to thousands of independent perturbations at single-cell resolution. The measured cell population contains three well-characterized cancer cell lines, including A549, a human lung adenocarcinoma, K562, a chronic myelogenous leukemia, and MCF7, a mammary adenocarcinoma cell line. Due to different transcriptional profiles of each cancer cell line, drug compounds might cause divergent cellular responses in each subpopulation. For our analysis, we consider the drug Givinostat, a histone deacetylase inhibitor with potential anti-inflammatory, anti-angiogenic, and antineoplastic activities (Rambaldi et al., 2010). The dataset contains $17,565$ control cells as well as $3,541$ cells perturbed by Givinostat with different dosages, i.e., $10\,\text{nM}$, $100\,\text{nM}$, $1,000\,\text{nM}$, $10,000\,\text{nM}$.

Data Preprocessing The data is available for download in the Gene Expression Omnibus (GEO) database under accession number GSM4150378. For data quality control and preprocessing, we follow the analysis of Lotfollahi et al. (2021). The count matrix obtained from GEO consists of $581,777$ cells. The data was subset to half its size, with $290,888$ cells remaining after quality control for all 188 different compounds. We proceeded with log-transformation and the the selection of $1,000$ highly-variant genes using `scanpy` (Wolf et al., 2018).

Feature Selection Single-cell RNA sequencing data is very high-dimensional, even after selecting $1,000$ highly-variant genes. For the downstream analysis of how well the overall perturbation effect has been captured, we thus select the top 50 marker genes, i.e., those genes which show strong differences between perturbed and unperturbed states. This analysis is conducted based on the `scanpy`'s function `rank_genes_groups`, setting unperturbed cells as reference (Wolf et al., 2018). It is important to note here, that CONDOT operates on the full dataset and the marker genes are only considered to report meaningful evaluation measures.

D.2 ... by Norman et al. (2019)

Genetic interactions and their joint expression give rise to an inconceivable organismal complexity and uncountable many diverse phenotypes and behaviors. Constructing a systematic genetic interaction map is crucial for a better understanding of cellular mechanisms in health and disease. Thus, Norman

et al. (2019) conducted single-cell, pooled transcriptional profiling of CRISPR-mediated perturbations to link genetic perturbation to its transcriptional consequences using the Perturb-Seq technology (Dixit et al., 2016). The dataset consists of individual perturbations as well as joint knockouts of different genes, allowing us to study the phenotypic consequences of perturbing a pair of genes alone or in combination. The indicator matrix of all individual perturbations as well as those pairs available in combination can be found in Fig. 12a.

Data Preprocessing The data is available for download in the Gene Expression Omnibus (GEO) database under accession number GSE133344. For data quality control and preprocessing, we follow the analysis of Lotfollahi et al. (2021). We discarded those genetic perturbations with less than 250 cells, resulting in a dataset with 92 individual perturbations and 84 perturbations in combination. This further included, the exclusion of particular subsets of control cells with in total $98,419$ remaining, data normalization, log-transformation, and selection of $1,500$ highly-variant genes using `scanpy` (Wolf et al., 2018).

Feature Selection Similar as above, for evaluation we select the top 50 marker genes, i.e., those genes most strongly affected by the particular genetic perturbation.

Data Splits Following Lotfollahi et al. (2021), we create different train / test dataset splits of increasing difficulty. The train splits hereby always contain all 92 individual perturbations as well as varying numbers of combinations. The easiest train split contains 55 perturbations, while the test set only carries 28 combinations which are unknown in the evaluation. Consecutive splits get increasingly harder, comprising 42, 29, 16, and 4 combinations in the train set (besides all single perturbations) and 41, 54, 67, and 79 combinations in the test set, respectively (see Fig. 12b).

E Experimental Details

In the following, we describe the experimental setup by providing an overview on the baselines, evaluation metrics, parameterizations of CONDOT's modules, and hyperparameters chosen.

E.1 Baselines

We consider several baselines to put CONDOT's performance into perspective. This includes current state-of-the-art methods, as well as ablations of our methods.

Compositional Perturbation Autoencoder (CPA) Building up on previous work (Lotfollahi et al., 2019, 2020), the current state-of-the-art approach conditional perturbation autoencoder (CPA) learns transcriptional perturbation responses across different cell types, applied dosages, and perturbation combinations (Lotfollahi et al., 2021). The architecture hereby consists of several modules. CPA predicts perturbed states of populations by learning a factorized latent representation of both perturbations and covariates, with separate embeddings for particle feature vectors, perturbations, and external covariates. These embeddings are independent of each other by design to later allow modular recombination of different modules and thus allowing the model to make predictions on unseen perturbations in combination. We follow the experimental setup outlined in (Lotfollahi et al., 2021). Similarly to perturbations, covariates such as cell type or dosage are encoded via one-hot vectors. Thus, CPA can not be utilized to make predictions on *unknown* perturbations as studied in § 5.3.2. This limitation is addressed in concurrent work, building up on CPA (Hetzel et al., 2022).

ICNN OT A crucial ablation study of CONDOT is to learn the transition of source population μ to target population ν *without* considering context c. Thus, we use standard ICNNs (3) to parameterize the transport map module $\mathcal{T}_\theta$ via two dual potentials as proposed in Makkuva et al. (2020) and Bunne et al. (2021). As for the PICNNs, we utilize different initialization schemes as derived in § 4.

E.2 Evaluation Metrics

Since we lack access to the ground truth pair of perturbed and unperturbed observations on the single cell level, we consider evaluation metrics on the level of the distribution of real and predicted perturbation states to analyze the effectiveness of CONDOT. We report results based on several metrics:

Wasserstein Distance We measure accuracy of the predicted target population $\hat{\nu}$ to the observed target population ν using the entropy-regularized Wasserstein distance (Cuturi, 2013) provided in the `OTT` library (Cuturi et al., 2022) defined as

$$W_\varepsilon(\hat{\nu}, \nu) := \min_{\mathbf{P} \in U(\hat{\nu}, \nu)} \langle \mathbf{P}, [\|x_i - y_j\|^2]_{ij} \rangle - \varepsilon H(\mathbf{P}), \tag{15}$$

where $H(\mathbf{P}) := -\sum_{ij} \mathbf{P}_{ij}(\log \mathbf{P}_{ij} - 1)$ and the polytope $U(\hat{\nu}, \nu)$ is the set of $n \times m$ matrices $\{\mathbf{P} \in \mathbb{R}_+^{n \times m}, \mathbf{P}\mathbf{1}_m = \hat{\nu}, \mathbf{P}^\top \mathbf{1}_n = \nu\}$. Throughout the evaluation, we set $\varepsilon = 0.1$.

Maximum Mean Discrepancy Kernel maximum mean discrepancy (Gretton et al., 2012) is another metric to measure distances between distributions, i.e., for our purpose between the predicted target population $\hat{\nu}$ to the observed target population ν. Given two random variables x and y with distributions $\hat{\nu}$ and ν, and a kernel function ω, Gretton et al. (2012) define the squared MMD as:

$$\text{MMD}(\hat{\nu}, \nu; \omega) = \mathbb{E}_{x,x'}[\omega(x, x')] + \mathbb{E}_{y,y'}[\omega(y, y')] - 2\mathbb{E}_{x,y}[\omega(x, y)].$$

We report an unbiased estimate of $\text{MMD}(\hat{\nu}, \nu)$, in which the expectations are evaluated by averages over the population particles in each set. We utilize the RBF kernel, and as is usually done, report the MMD as an averaged over several length scales, i.e., $0.5, 0.1, 0.01$, and 0.005.

Perturbation Signatures ℓ_2(PS) refers to the ℓ_2-distance between the perturbation signatures (PS) computed on the observed and predicted distributions, ν and $\hat{\nu}$. A perturbation signatures itself is defined as the difference in means between the distribution of perturbed states and control states of each feature, e.g., here individual genes. As before, let μ be the set of observed unperturbed population particles, ν the set of observed perturbed particles, as well as $\hat{\nu}$ the predicted perturbed state of population μ. The perturbation signature is then defined as

$$\text{PS}(\nu, \mu) = \frac{1}{m} \sum_{y_i \in \nu} y_i - \frac{1}{n} \sum_{x_i \in \mu} x_i,$$

where n is the size of the unperturbed and m of the perturbed population. We report the ℓ_2 distance between the observed signature $\text{PS}(\nu, \mu)$ and the predicted signature $\text{PS}(\hat{\nu}, \mu)$, which is equivalent to simply computing the difference in the means between the observed and predicted distributions.

E.3 CONDOT Modules

CONDOT consists of several modules for which different choices can be considered. Here, we provide a brief overview on the options and their parameterization.

E.3.1 Embedding Module $\mathcal{E}$

The embedding module allows us to consider context variables c of various nature. In the case of scalars, no sophisticated embedding is necessary. In contrast, covariate contexts as well as potentially complex action descriptions require embeddings in order to be processed by the combinator $\mathcal{C}$, and transport map module $\mathcal{T}$.

One-Hot Embedding $\mathcal{E}_{\text{ohe}}$ Covariates, such as subpopulation or patient identifiers, can be simply embedded via one-hot encodings. These embeddings, however, are not able to capture unknown covariates after training.

Mode-of-Action Embedding $\mathcal{E}_{\text{moa}}$ In certain cases, actions might possess distinct properties which allow for a direct embeddings using this domain knowledge, i.e., molecular representations for molecules (Rong et al., 2020; Rogers and Hahn, 2010). In the case of genetic perturbations, however, no straightforward embedding is available. We thus introduce so-called mode-of-action embeddings, which map actions into a latent space-based on their mechanism of action and effect on the target population. In the fashion of word embeddings (Mikolov et al., 2013a,b,c), we require actions with

similar effect to be closely embedded in the learned representation. This means, however, that we require some sample access of target population particles, i.e., perturbed cells by individual compounds (not in combination). While several metric embeddings are possible (Chopra et al., 2005), we here test a simple multi-dimensional scaling-based embedding (Mead, 1992). For this we compute the pairwise Wasserstein distance matrix between all target populations of different individual perturbations. We then compute a 10-dimensional MDS embedding-based on the stress minimization using majorization algorithm (smacof) (De Leeuw and Mair, 2009) of `sklearn` (Pedregosa et al., 2011), which serves as descriptor for each individual perturbation.

E.3.2 Combinator Module $\mathcal{C}$

The combinator module allows us to pass an arbitrary number of context c to the transport map module $\mathcal{T}$.

Multi-Hot Combinator $\mathcal{C}_{+}^{\text{ohe}}$ A naïve way of constructing the combinator is to combine different actions via a multi-hot encoding. If all single perturbations are observed during training, each individual action can be represented via a one-hot encoding. The potential combination of different actions, is then encoded by adding the respective one-hot encodings, resulting in a multi-hot encoding for each combination. A limitation of this embedding, however, is that it cannot generalize to unknown action after training.

Deep Set Combinator $\mathcal{C}_{\Phi}^{\text{moa}}$ When not considering one-hot-based embeddings and when aiming to generalize to unseen perturbations, we need a combinator module which learns how to associate different individual embeddings with each other to receive a joint embedding. As we for now do not make an assumption on the order of the perturbation, we consider a permutation-invariance network architecture such as deep sets (Zaheer et al., 2017) with parameters Φ. Taking a set of arbitrary size k containing individual context embeddings $\{\mathcal{E}_{\text{moa}}(c^1), \mathcal{E}_{\text{moa}}(c^2), \ldots, \mathcal{E}_{\text{moa}}(c^k)\}$, it returns a learned combination embedding $\hat{c}_i = \mathcal{C}_{\Phi}(\mathcal{E}_{\text{moa}}(c^1), \mathcal{E}_{\text{moa}}(c^2), \ldots, \mathcal{E}_{\text{moa}}(c^k))$.

E.3.3 Transport Map Module $\mathcal{T}$

The transport map module takes as input samples of the source distribution μ as well as context c and returns the perturbed population ν. Map $\mathcal{T}_\theta$ is thereby parameterized via PICNNs as we require input convexity in μ but not c. In the case where we consider learning $\mathcal{T}_\theta$ via the dual (2), it is defined by a pair of PICNNs with parameters $\theta = (\theta_f, \theta_g)$, parameterizing the set of dual variables f and g. When deploying the primal OT problem (1), we parameterize a single Brenier potential via a PICNN with parameters θ.

As suggested by Makkuva et al. (2020), we relax the convexity constraint on PICNN g and instead penalize its negative weights W_k^z

$$R(\theta) = \lambda \sum_{W_k^z \in \theta} \|\max(-W_k^z, 0)\|_F^2 .$$

The convexity constraint on PICNN f is enforced after each update by setting the negative weights of all $W_k^z \in \theta_f$ to zero. Thus, the full objective then states

$$\max_{\theta_f : W_k^z \geq 0, \forall k} \min_{\theta_g} f_{\theta_f}(\nabla g_{\theta_g}(y)) - \langle y, \nabla g_{\theta_g}(y) \rangle - f_{\theta_f}(x) + \lambda R(\theta_g).$$

E.3.4 Projection Module

For very high-dimensional inputs such as single-cell RNA seq data, we project the data into a lower-dimensional space. The effect of a perturbation effect is then learned on the control particles encoded into a lower dimensional space. Subsequently, we decode the predicted target particles into the original data space. We consider both, principal component (PCA) as well as autoencoder-based projections. When conducting experiments in PCA space, we consider the first 50 principal components, as they contain $> 99\,\%$ of the explained variance (see Fig. 6c). The autoencoder architecture is inspired by (Lotfollahi et al., 2019), as it has been designed and tested for single-cell RNA seq data. The results reported in § 5 are based on autoencoder projections and the evaluation metrics are computed on the decoded target particles.

E.4 Hyperparameters

To learn the optimal transport maps, we use PICNN architectures of 4 hidden layers of width 64. The autoencoder parameterizing the projection module consists of an encoder and decoder with each 2 layers of 512 dimensional hidden layers. The size of the latent space is 50. The deep set consists of an encoder with 2 linear layers with 8 hidden units, followed by a `sum`-pooling operator and a 2 layer decoder with 8 hidden units, returning a set embedding of the same size as each individual input embedding, and passed through a final sigmoid activation function. For all networks, we use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001 ($\beta_1 = 0.5$, $\beta_2 = 0.9$) and λ=1. If $\mathcal{T}$ is learned via the OT dual, f and g are learned via an alternate min-max optimization. f is updated by fixing g and maximizing (13) with a single iteration. Then, for 10 iterations, i.e., `train_freq_f`$= 10$, f is fixed, and g is optimized by minimizing (14). For the baselines, we followed the default configurations specified by the authors on the same datasets. We use a default batch size of 256, which is adapted for perturbations with fewer cells (due to a train / test split of $80\%/20\%$).