Meta Optimal Transport

Brandon Amos 1  Giulia Luise 2  Samuel Cohen 1 3 4  Ievgen Redko 5 6

Abstract
We study the use of amortized optimization to predict optimal transport (OT) maps from the input measures, which we call Meta OT. This helps repeatedly solve similar OT problems between different measures by leveraging the knowledge and information from past problems to rapidly predict and solve new problems. Otherwise, standard methods ignore the knowledge of the past solutions and sub-optimally re-solve each problem from scratch. We instantiate Meta OT models in discrete and continuous settings between grayscale images, spherical data, classification labels, and color palettes and use them to improve the computational time of standard OT solvers. Our source code is available at http://github.com/facebookresearch/meta-ot.

1. Introduction
Optimal transportation (Villani, 2009; Ambrosio, 2003; Santambrogio, 2015; Peyré et al., 2019; Merigot and Thibert, 2021) is thriving in domains including economics (Galichon, 2016), reinforcement learning (Dadashi et al., 2021; Fickinger et al., 2022), style transfer (Kolkin et al., 2019), generative modeling (Arjovsky et al., 2017; Seguy et al., 2018; Huang et al., 2021; Rout et al., 2022), geometry (Solomon et al., 2015; Cohen et al., 2021), domain adaptation (Couty et al., 2017; Redko et al., 2019), signal processing (Kolouri et al., 2017), fairness (Jiang et al., 2019), and cell reprogramming (Schiebinger et al., 2019). These settings couple two measures (α, β) supported on domains (X, Y) by solving a transport optimization problem such as the primal Kantorovich problem defined by

$$\pi^*(\alpha, \beta, c) \in \arg \min_{\pi \in \mathcal{U}(\alpha, \beta)} \int_{X \times Y} c(x, y) d\pi(x, y),$$  \hspace{1cm} (1)

where the optimal coupling π* is a joint distribution over the product space, \(\mathcal{U}(\alpha, \beta)\) is the set of admissible couplings between α and β, and \(c : X \times Y \to \mathbb{R}\) is the ground cost, that represents a notion of distance between elements in \(X\) and elements in \(Y\).

Challenges. Unfortunately, solving eq. (1) once is computationally expensive between general measures and computationally cheaper alternatives are an active research topic: Entropic optimal transport (Cuturi, 2013) smooths the transport problem with an entropy penalty, and sliced distances (Kolouri et al., 2016; 2019b; Deshpande et al., 2019) solve OT between 1-dimensional projections of the measures, where eq. (1) can be solved easily.

When an optimal transport method is deployed in practice, eq. (1) is not just solved once, but is repeatedly solved for new scenarios between different input measures (α, β). For example, the measures could be representations of images we care about optimally transporting between and in deployment we would receive a stream of new images to couple. Repeatedly solving optimal transport problems also comes up in the context of comparing seismic signals (Engquist and Froese, 2013) and in single-cell perturbations (Bunne et al., 2021; 2022b; a). Standard optimal transport solvers deployed in this setting re-solve the optimization problems from scratch and ignore the shared structure and information present between different coupling problems.

Overview. We study the use of amortized optimization and machine learning methods to rapidly solve multiple optimal transport problems and predict the solution from the input measures (α, β). This setting involves learning a meta model to predict the solution to the optimal transport problem, which we will refer to as Meta Optimal Transport. We learn Meta OT models to predict the solutions to optimal transport problems and significantly improve the computational time and number of iterations needed to solve eq. (1).

Settings that are not Meta OT. Meta OT is not useful in settings that do not repeatedly solve OT problems, e.g. 1) generative modeling settings, such as Arjovsky et al. (2017), that estimate the OT distance between the data and model distributions, and 2) out-of-sample settings (Seguy et al., 2018; Perrot et al., 2016) that couple measures and then extrapolate the map to larger measures.
2. Preliminaries and background

2.1. Entropic OT between discrete measures

We review foundations of OT, following the notation of Peyré et al. (2019) in most places. The discrete setting often favors the entropic regularized version since it can be computed efficiently and in a parallelized way using the Sinkhorn algorithm. While the primal Kantorovich formulation in eq. (1) provides an intuitive problem description, OT problems are rarely solved directly in this form due to the high-dimensionality of the couplings \( \pi \) and the difficulty of satisfying the coupling constraints \( U(\alpha, \beta) \).

Instead, most computational OT solvers use the dual of eq. (1), which we build our OT solvers on top of.

Let \( \alpha := \sum_{i=1}^{m} a_i \delta_i \) and \( \beta := \sum_{i=1}^{n} b_i \delta_i \), be discrete measures, where \( \delta_i \) is a Dirac at point \( z \) and \( a, \beta \in \Delta_{m-1} \) and \( b \in \Delta_{n-1} \) are in the probability simplex defined by

\[
\Delta_{k-1} := \{ x \in \mathbb{R}^k : x \geq 0 \text{ and } \sum_i x_i = 1 \}.
\]

**Discrete OT.** Eq. (1) becomes the linear program

\[
P^*(\alpha, \beta, c, \epsilon) \in \arg \min_{P \in U(a,b)} \{ C(P) - \epsilon H(P) \}
\]

where \( U(a,b) := \{ P \in \mathbb{R}_{+}^{m \times n} : P1_m = a, P^T1_n = b \} \), \( P \) is a coupling matrix, \( P^* \) is the optimal coupling, and the cost can be discretized as a matrix \( C \in \mathbb{R}^{m \times n} \) with entries \( C_{i,j} := c(x_i, y_j) \), and \( \{ C \} := \sum_{i,j} C_{i,j} P_{i,j} \).

**Entropic OT.** The linear program above can be regularized adding an entropy term to smooth the objective as in Cominetti and Martín (1994); Cuturi (2013), resulting in:

\[
P^*(\alpha, \beta, c, \epsilon) \in \arg \min_{P \in U(a,b)} \{ C(P) - \epsilon H(P) \}
\]

where \( H(P) := -\sum_{i,j} P_{i,j} (\log(P_{i,j}) - 1) \) is the discrete entropy of a coupling matrix \( P \).

**Entropic OT dual.** As presented in Peyré et al. (2019, Prop. 4.4), setting \( K \in \mathbb{R}^{m \times n} \) to the *Gibbs kernel* \( K_{i,j} := \exp\{-C_{i,j}/\epsilon\} \), the dual of eq. (4) is

\[
f^*, g^* \in \arg \max_{f \in \mathbb{R}^n, g \in \mathbb{R}^m} \{ \langle f, a \rangle + \langle g, b \rangle - \epsilon e^{f/g} Ke^{g/f} \}
\]

where the dual variables or potentials \( f \in \mathbb{R}^n \) and \( g \in \mathbb{R}^m \) are associated, respectively, with the marginal constraints \( P1_m = a \) and \( P^T1_n = b \). We omit the dependencies of the duals on the context, e.g., \( f^* \) is shorthand for \( f^*(\alpha, \beta, c, \epsilon) \).

**Recovering the primal solution from the duals.** Given optimal duals \( f^*, g^* \) that solve eq. (5) the optimal coupling \( P^* \) to the primal problem in eq. (4) can be obtained by

\[
P^*_{i,j}(\alpha, \beta, c, \epsilon) := \exp\{f^*_i/\epsilon\} K_{i,j} \exp\{g^*_j/\epsilon\}.
\]

**The Sinkhorn algorithm.** Algorithm 1 summarizes the log-space version, which takes closed-form block coordinate ascent updates on eq. (5) obtained from the first-order optimality conditions (Peyré et al., 2019, Remark 4.21). We will fine-tune Meta OT predictions with Sinkhorn.

**Algorithm 1** Sinkhorn(\( \alpha, \beta, c, \epsilon, f_0 = 0 \))

\[
\text{for iteration } i = 1 \text{ to } N \text{ do}
\begin{align*}
g_i &\leftarrow \epsilon \log b - \epsilon \log \{ K^T \exp\{f_{i-1}/\epsilon\} \} \\
f_i &\leftarrow \epsilon \log a - \epsilon \log \{ K \exp\{g_{i}/\epsilon\} \}
\end{align*}
\]

end for

Compute \( P_N \) from \( f_N, g_N \) using eq. (6)

return \( P_N \approx P^* \)

**Algorithm 2** W2GN(\( \alpha, \beta, \varphi_0 \))

\[
\text{for iteration } i = 1 \text{ to } N \text{ do}
\begin{align*}
\text{Sample from } (\alpha, \beta) \text{ and estimate } \mathcal{L}(\varphi_{i-1}) \text{ (eq. (13))} \\
\text{Update } \varphi_i \text{ with approximation to } \nabla_\varphi \mathcal{L}(\varphi_{i-1})
\end{align*}
\]

end for

return \( T_N(\cdot) := \nabla \varphi_{N}(\cdot) \approx T(\cdot) \)

2.2. OT between continuous (Euclidean) measures

Let \( \alpha \) and \( \beta \) be continuous measures in Euclidean space \( \mathcal{X} = \mathcal{Y} \subseteq \mathbb{R}^d \) (with \( \alpha \) absolutely continuous with respect to the Lebesgue measure) and the ground cost be the squared Euclidean distance \( c(x, y) := \|x-y\|^2 \). Then the minimum of eq. (1) defines the square of the Wasserstein-2 distance:

\[
W^2_2(\alpha, \beta) := \min_{\pi \in \mathcal{E}(\alpha, \beta)} \int_{\mathcal{X} \times \mathcal{Y}} \|x-y\|^2 d\pi(x,y)
\]

\[
= \min_T \int_{\mathcal{X}} \|x - T(x)\|^2 d\alpha(x),
\]

where \( T \) is a *transport map* pushing \( \alpha \) to \( \beta \), i.e., \( T\# \alpha = \beta \) with the pushforward operator defined by \( T\# \alpha(B) := \alpha(T^{-1}(B)) \) for any measurable set \( B \).

**Convex dual potentials.** The primal in eq. (10) is difficult to solve due to the constraints and many computational methods (Makkuva et al., 2020; Taghvaei and Jalali, 2019; Korotin et al., 2021; Amos, 2023) solve the dual

\[
\psi^*(\cdot; \alpha, \beta) \in \arg \min_{\psi \in \text{convex}} \int_{\mathcal{X}} \psi(x)d\alpha(x) + \int_{\mathcal{Y}} \bar{\psi}(y)d\beta(y),
\]

where \( \psi^* \) is the dual functional of \( \psi \).
where $\psi$ is a convex function referred to as a potential, and $\overline{\psi}(y) := \max_{x \in X} (x, y) - \psi(x)$ is the Legendre-Fenchel transform or convex conjugate of $\psi$ (Fenchel, 1949; Rockafellar, 2015). The potential may be approximated with an input-convex neural network (ICNN) (Amos et al., 2017).

### Recovering the primal solution from the dual

Given an optimal dual $\psi^*$ for eq. (11), Brenier (1991) shows that an optimal map $T^*$ for eq. (10) can be obtained with

$$T^*(x) = \nabla_x \psi^*(x).$$

### Wasserstein-2 Generative Networks (W2GNs)

Korotin et al. (2021a) model $\psi_\alpha$ and $\overline{\psi}_\phi$ in eq. (11) with two separate ICNNs parameterized by $\phi$. The separate model for $\overline{\psi}_\phi$ is useful because the conjugate operation in eq. (11) becomes computationally expensive. They optimize the loss

$$L(\hat{\phi}) := \mathbb{E}_{x \sim \alpha} [\psi_{\hat{\phi}}(x)] + \mathbb{E}_{y \sim \beta}[\langle \nabla \overline{\psi}_{\hat{\phi}}(y), y - \psi_{\hat{\phi}}(\nabla \overline{\psi}_{\hat{\phi}}(y))\rangle]$$

$$+ \gamma \mathbb{E}_{y \sim \beta} \|\nabla \psi_{\hat{\phi}} \circ \nabla \overline{\psi}_{\hat{\phi}}(y) - y\|_2^2,$$

where $\psi_{\hat{\phi}}$ is a detached copy of the parameters and $\gamma$ is a hyper-parameter. The first term are the cyclic monotone correlations (Chartrand et al., 2009; Taghvaei and Jalali, 2019), that optimize the dual objective in eq. (11), and the second term provides cycle consistency (Zhu et al., 2017) to estimate the conjugate $\overline{\psi}$. Algorithm 2 shows how $L$ is typically optimized using samples from the measures, which we use to fine-tune Meta OT predictions.

### 2.3. Amortized optimization and learning to optimize

Our paper is an application of amortized optimization methods that predict the solutions of optimization problems, as surveyed in, e.g., Chen et al. (2021); Amos (2022). We use the setup from Amos (2022), which considers unconstrained continuous optimization problems

$$z^*(\phi) \in \arg \min_z J(z; \phi),$$

where $J$ is the objective, $z \in Z$ is the domain, and $\phi \in \Phi$ is some context or parameterization. In other words, the context conditions the objective but is not optimized over. Given a distribution over contexts $P(\phi)$, we learn a model $\hat{z}_\theta$ parameterized by $\theta$ to approximate eq. (14), i.e. $\hat{z}_\theta(\phi) \approx z^*(\phi)$. $J$ will be differentiable, so we optimize the parameters using objective-based learning with

$$\min_{\theta} \mathbb{E}_{\phi \sim P(\phi)} J(\hat{z}_\theta(\phi); \phi),$$

which does not require ground-truth solutions $z^*$ and can be optimized with a gradient-based solver.

### 3. Meta Optimal Transport

We refer to Meta Optimal Transport as the setting when amortized optimization (sect. 2.3) is used for predicting solutions to optimal transport problems such as eq. (1). We refer to the distribution over the OT problems (measures and costs) as the meta-distribution and denote it as $D(\alpha, \beta, c)$, which we call meta to distinguish it from the measures $\alpha$, $\beta$. For example, sects. 4.1.1 and 4.1.2 considers meta-distributions over the weights of the atoms, i.e. $(a, b) \sim D$, where $D$ is a distribution over $\Delta_{m-1} \times \Delta_{n-1}$. While a model could directly predict the primal solution to eq. (1), i.e. $P_\theta(\alpha, \beta, c) \approx P^*(\alpha, \beta, c)$ for $(\alpha, \beta, c) \sim D$, this is difficult due to the coupling constraints. We instead opt to predict the dual variables. Figure 1 illustrates Meta OT in discrete and continuous settings.

### 3.1. Meta OT between discrete measures

We build on standard methods for entropic OT reviewed in sect. 2.1 between discrete measures $\alpha := \sum_{i=1}^m a_i \delta_{x_i}$ and $\beta := \sum_{j=1}^n b_j \delta_{x_j}$, with $a \in \Delta_{m-1}$ and $b \in \Delta_{n-1}$ coupled using a cost $c$. In the Meta OT setting, the measures and
cost are the contexts for amortization and sampled from a meta-distribution, i.e. \((\alpha, \beta, c) \sim D(\alpha, \beta, c)\). For example, sects. 4.1.1 and 4.1.2 considers meta-distributions over the weights of the atoms, i.e. \((a, b) \sim D\), where \(D\) is a distribution over \(\Delta_{m-1} \times \Delta_{n-1}\).

Amortization objective. We will seek to predict the optimal potential. At optimality, the pair of potentials are related to each other via eq. (8), i.e. 
\[ g(f; \alpha, \beta, c) := \epsilon \log b - \epsilon \log (K^\top \exp \{f/e\}) \]
where \(K \in \mathbb{R}^{m \times n}\) is the Gibbs kernel from eq. (5). Hence, it is sufficient to predict one of the potentials, e.g. \(f\), and recover the other. We thus re-formulate eq. (5) to just optimize over \(f\) with
\[ f^*(\alpha, \beta, c, e) \in \arg \min_{f \in \mathbb{R}^n} J(f; \alpha, \beta, c), \tag{16} \]
where
\[ -J(f; \alpha, \beta, c) := \langle f, a \rangle + \langle g, b \rangle - \epsilon \exp \{f/e\}, K \exp \{g/e\} \]
is the (negated) dual objective. Even though most solvers optimize over \(f\) and \(g\) jointly as in eq. (16), amortizing over these would likely need to have a higher capacity than a model just predicting \(f\) and learn how \(f\) and \(g\) are connected through eq. (8) while in eq. (16) we explicitly provide this knowledge.

Amortization model. We predict the solution to eq. (16) with \(\hat{f}_\theta(\alpha, \beta, c)\) parameterized by \(\theta\), resulting in a computationally efficient approximation \(\hat{f}_\theta \approx f^*\). Here we use the notation \(\hat{f}_\theta(\alpha, \beta, c)\) to mean that the model \(\hat{f}_\theta\) depends on representations of the input measures and cost. In our settings, we define \(\hat{f}_\theta\) as a fully-connected MLP mapping from the atoms of the measures to the duals.

Amortization loss. Applying objective-based amortization from eq. (15) to the dual in eq. (16) completes the learning setup. The model should optimize the expected dual value:
\[ \min_{\theta} \mathbb{E}_{(\alpha, \beta, c) \sim D} J(\hat{f}_\theta(\alpha, \beta, c); \alpha, \beta, c), \tag{17} \]
which is appealing as it does not require ground-truth solutions \(f^*\). The ground-truth solutions may be expensive to obtain, but if they are available, a regression term can also be added (Amos, 2022). Algorithm 3 shows a basic training loop for eq. (17) using a gradient-based optimizer such as Adam (Kingma and Ba, 2015).

Sinkhorn fine-tuning. The dual prediction made by \(\hat{f}_\theta\) with an associated \(\hat{g}\) can be used to initialize a standard Sinkhorn solver. This allows for the predicted solution to be refined to an optimality threshold.

On accelerated solvers. While we have considered fine-tuning the Meta OT prediction with a log-Sinkhorn solver, Meta OT can also be combined with accelerated variants of entropic OT solvers such as Thibault et al. (2017); Altschuler et al. (2017); Alaya et al. (2019); Lin et al. (2019) that otherwise solve every problem from scratch.

<table>
<thead>
<tr>
<th>Algorithm 3 Training Meta OT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize amortization model with (\theta_0)</td>
</tr>
<tr>
<td>for iteration (i) do</td>
</tr>
<tr>
<td>Sample ((\alpha, \beta, c) \sim D)</td>
</tr>
<tr>
<td>Predict duals (\hat{f}<em>\theta) or (\hat{g}</em>\theta) on the sample</td>
</tr>
<tr>
<td>Estimate the loss in eq. (17) or eq. (18)</td>
</tr>
<tr>
<td>Update (\theta_{i+1}) with a gradient step</td>
</tr>
<tr>
<td>end for</td>
</tr>
</tbody>
</table>

3.2. Meta OT between continuous measures

We take an analogous approach to predicting the Wasserstein-2 map between continuous measures for Wasserstein-2 as reviewed in sect. 2.2. Here the measures \(\alpha, \beta\) are supported in continuous space \(\mathcal{X} = \mathcal{Y} = \mathbb{R}^d\) and we focus on computing Wasserstein-2 couplings from instances sampled from a meta-distribution \((\alpha, \beta) \sim D(\alpha, \beta)\). The cost \(c\) is not included in \(D\) as it remains fixed to the squared Euclidean cost everywhere here.

One challenge here is that the optimal dual potential \(\psi^*(\cdot; \alpha, \beta)\) in eq. (11) is a convex function and not simply a finite-dimensional real vector. The dual potentials in this setting are approximated by, e.g., an ICNN. We thus propose a Meta ICNN that predicts the parameters \(\varphi\) of an ICNN \(\psi_\varphi\) that approximates the optimal dual potentials, which can be seen as a hypernetwork (Stanley et al., 2009; Ha et al., 2017). The dual prediction made by \(\hat{\varphi}_\theta\) can easily be input as the initial value to a standard W2GN solver. App. D.2 discusses other modeling choices we considered: we tried models based on MAML (Finn et al., 2017) and neural processes (Garnelo et al., 2018b;a).

Amortization objective. We build on the W2GN formulation (Korotin et al., 2021a) and seek parameters \(\varphi^*\) optimizing the dual ICNN potentials \(\psi_{\varphi^*}\) and \(\psi_{\varphi}\) with \(\mathcal{L}(\varphi; \alpha, \beta)\) from eq. (13). We chose W2GN due to the stability, but could also easily use other losses optimizing ICNN potentials.

<table>
<thead>
<tr>
<th>Algorithm 3 Training Meta ICNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize amortization model with (\theta_0)</td>
</tr>
<tr>
<td>for iteration (i) do</td>
</tr>
<tr>
<td>Sample ((\alpha, \beta, c) \sim D)</td>
</tr>
<tr>
<td>Predict duals (\hat{f}<em>\theta) or (\hat{g}</em>\theta) on the sample</td>
</tr>
<tr>
<td>Estimate the loss in eq. (17) or eq. (18)</td>
</tr>
<tr>
<td>Update (\theta_{i+1}) with a gradient step</td>
</tr>
<tr>
<td>end for</td>
</tr>
</tbody>
</table>

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

We demonstrate how Meta OT models improve the convergence of the state-of-the-art solvers in settings where solving multiple OT problems naturally arises. We implemented our code in JAX (Bradbury et al., 2018) as an extension to the Optimal Transport Tools (OTT) package (Cuturi et al., 2022). App. B covers further experimental and implementation details, and shows that all of our experiments take a few hours to run on our single Quadro GP100 GPU. The source code to reproduce all of our experiments is available at http://github.com/facebookresearch/meta-ot.

4.1. Discrete OT

4.1.1. Grayscale image transport

Images provide a natural setting for Meta OT where the distribution over images provides the meta-distribution \( D \) over OT problems. Given a pair of images \( \alpha_0 \) and \( \alpha_1 \), each grayscale image is cast as a discrete measure in 2-dimensional space where the intensities define the probabilities of the atoms. The goal is to compute the optimal transport interpolation between the two measures as defined in eq. (7).

Amortization loss. Applying objective-based amortization from eq. (15) to the W2GN loss in eq. (13) completes our learning setup. We optimize the loss

\[
\min_\theta \mathbb{E}_{(\alpha, \beta) \sim \mathcal{D}} L(\hat{\varphi}_\theta(\alpha, \beta); \alpha, \beta).
\]

As in the discrete setting, this loss does not require ground-truth solutions \( \varphi^* \) and we find the solution with Adam.

Figure 2. Interpolations between MNIST test digits using couplings obtained from (left) solving the problem with Sinkhorn, and (right) Meta OT model’s initial prediction, which is \( \approx 100 \) times computationally cheaper and produces a nearly identical coupling.

Figure 3. Meta OT successfully predicts warm-start initializations that significantly improve the convergence of Sinkhorn iterations on test data. The error is the marginal error defined in eq. (7).

Our Meta OT model \( \hat{f}_\theta \) (sect. 3) is an MLP that predicts the transport map between pairs of MNIST digits. We train on every pair from the standard training dataset. Figure 2 shows that even without fine-tuning, Meta OT’s predicted Wasserstein interpolations between the measures are close to the ground-truth interpolations obtained from running the Sinkhorn algorithm to convergence. We then fine-tune Meta OT’s prediction with Sinkhorn. Figure 3 shows that the near-optimal predictions can be quickly refined in fewer iterations than running Sinkhorn with the default initialization, and table 1 shows the runtime required to reach an error threshold of \( 10^{-2} \), showing that the Meta OT initialization help solve the problems faster by an order of magnitude. We compare our learned initialization to the standard zero initialization, as well as the Gaussian initialization proposed in Thornton and Cuturi (2022), which takes a continuous Gaussian approximation of the measures and initializes the potentials to be the known coupling between the Gaussians. This Gaussian initialization assumes the squared Euclidean cost, which is not the case in our spherical transport problem, but we find it is still helpful over the zero initialization.

Out-of-distribution generalization We now test the abil-
Figure 4. Test set coupling predictions of the spherical transport problem. Meta OT’s initial prediction is $\approx 37500$ times faster than solving Sinkhorn to optimality. Supply locations are shown as black dots and the blue lines show the spherical transport maps $T$ going to demand locations at the end. The sphere is visualized with the Mercator projection.

4.1.2. Supply-Demand Transportation on Spherical Data

We next set up a synthetic transport problem between supply and demand locations where the supply and demands may change locations or quantities frequently, creating another Meta OT setting to be able to rapidly solve the new instances. We specifically consider measures living on the 2-sphere defined by $S_2 := \{ x \in \mathbb{R}^3 : \| x \| = 1 \}$, i.e. $\mathcal{X} = \mathcal{Y} = S_2$, with the transport cost given by the spherical distance $c(x, y) = \arccos(x \cdot y)$. We then randomly sample supply locations uniformly from Earth’s landmass and demand locations from Earth’s population density to induce a class of transport problems on the sphere obtained from the CC-licensed dataset from Doxsey-Whitfield et al. (2015). Figure 4 shows that the predicted transport maps on test instances are close to the optimal maps obtained from Sinkhorn to convergence. Similar to the MNIST setting, fig. 3 and table 1 show improved convergence and runtime.

4.1.3. Wasserstein Adversarial Regularization

Wasserstein losses has recently attracted a considerable attention in the field of multi-label (Froger et al., 2015; Yang et al., 2018; Jawanpuria et al., 2021; Toyokuni et al., 2021) and multi-class classification (Liu et al., 2020a;b; 2019; Han et al., 2020; Fatras et al., 2021) as they both require finding an informative way of comparing discrete distributions given by the true labeling of the data points and those predicted by the classification model. In this experiment, we aim to show that meta OT model can be learned alongside the training of the multi-class classification model and used to make predictions for the Wasserstein loss term appearing in the objective function of the latter. For this, we consider as an example the setup of Fatras et al. (2021) where the authors define an adversarial loss term (called WAR) aiming at limiting the effect of label noise on the generalization capacity of deep vision neural networks. In particular, given a neural network $p_\theta$ predicting a vector of class memberships in $\mathbb{R}^c$, the regularization term is

$$R_{\text{WAR}}(x_i) = W^c_\epsilon(p_\theta(x_i + r^a_i), p_\theta(x_i))$$

$$r^a_i = \arg\max_{r_i : ||r_i|| \leq \epsilon} W^c_\epsilon(p_\theta(x_i + r_i), p_\theta(x_i)).$$

(19)
where $W^C_C$ is the Wasserstein distance with entropic regularization introduced in eq. (4) with a cost matrix $C \in \mathbb{R}^{c \times c}$. Learning $p_0$ is done by optimizing the cross entropy loss together with $R_{\text{WAR}}(x_i)$ using stochastic optimization. This means that OT problems in eq. (19) are solved repeatedly, for every batch in the input dataset and during multiple epochs thus making the meta OT warm-starts particularly computationally attractive in this context. For this task, we optimize a meta OT model defined as a MLP with 3 hidden layers over the same data alongside the main optimization procedure. We use meta OT model to predict the solutions to both OT problems in eq. (19) and use only 25% of iterations in the Sinkhorn loop to compute $W^C_C$. As in Fatras et al. (2021), we evaluate the efficiency of such learning strategy on three computer vision datasets, namely: Fashion MNIST, Cifar-10 and Cifar-100. For each of them, we consider the clean version of the data (0% noise), and two variations with 20% and 40% of noise in labels. The authors of Fatras et al. (2021) experiment with two cost matrices: one is defined based on the distances between the class centroids of 30000 samples from the original dataset when embedded with ResNet18; second one is defined as the Euclidean distance between the word2vec embeddings of the classes of the original dataset. To show the versatility of our approach with respect to different geometries, we use the first cost matrix for Fashion MNIST dataset, and the second one for Cifar-10 and Cifar-100 datasets.

We evaluate meta OT for this task based on three criteria. First, we want to make sure that reducing the number of iterations in the Sinkhorn loop is not detrimental for the overall performance of the learned classification model. These results are presented in table 2, where we can see that meta OT leads to the same performance as the original WAR model while doing only 5 iterations of Sinkhorn on top of the initial predictions. Second, we show in sect. 4.1.2 that our meta OT model predicts warm-start initializations that have a low marginal error so that even its initial predictions are become at least as qualitative as the solution obtained using 20 iterations of the Sinkhorn algorithm. Finally, we show in table 3, that training a meta OT model alongside the main model doesn’t introduce any additional overhead in terms of computational time. In this table, we compare the average runtime of each of the considered baselines and account for the time needed to make a backward pass for the meta OT model. This result is important as once a meta OT model is trained, it can be further used to make predictions without any fine-tuning for other training runs with different hyperparameters leading to an important reduction in terms of computational time.

Table 2. Comparison of the original WAR implementation with WAR implementation using only 5 Sinkhorn iterations and our Meta OT model with 5 Sinkhorn iterations on top of initial predictions. We report the mean and std across 3 random seeds.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Noise level</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>WAR (5 iter.)</td>
</tr>
<tr>
<td>Fashion MNIST</td>
<td>0%</td>
<td>94.75 ± 0.05</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>93.00 ± 0.15</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>88.67 ± 0.10</td>
</tr>
<tr>
<td>Cifar-10</td>
<td>0%</td>
<td>91.96 ± 0.13</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>88.80 ± 0.11</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>81.09 ± 0.06</td>
</tr>
<tr>
<td>Cifar-100</td>
<td>0%</td>
<td>70.93 ± 0.23</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>66.23 ± 0.29</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>52.69 ± 0.12</td>
</tr>
</tbody>
</table>

Figure 5. Marginal errors throughout WAR training for CIFAR-100 classification. The bumps correspond to when the loss and learning rate are updated during training as described in Fatras et al. (2021)
4.2. Continuous OT for color transfer

The problem of color transfer between two images consists in mapping the color palette of one image into the other one. The images are required to have the same number of channels, for example RGB images. The continuous formulation that we use from Korotin et al. (2021a), takes i.e. \( X = Y = [0, 1]^3 \) with \( \epsilon \) being the squared Euclidean distance. We collected \( \approx 200 \) public domain images from WikiArt and trained a Meta ICNN model from sect. 3.2 to predict the color transfer maps between every pair of them. Figure 6 shows the predictions on test pairs.
complexity. In the meta-OT setting, we consider learning to rapidly compute OT mappings between new pairs measures. All these works can hence benefit from an acceleration effect with amortization.

Embedding measures where OT distances are discriminative. Effort has been invested in learning encodings/projections of measures through a nested optimization problem, which aims to find discriminative embeddings of the measures to be compared (Genevay et al., 2018; Deshpande et al., 2019; Nguyen and Ho, 2022). While these works share an encoder and/or a projection across task with the aim of leveraging more discriminative alignments (and hence an OT distance with a metric different from the Euclidean metric), our work differs in the sense that we find good initializations to solve the OT problem itself with fixed cost more efficiently across tasks.

Optimal transport and amortization. Courty et al. (2018) learn a latent space in which the Wasserstein distance between the measure’s embeddings is equivalent to the Euclidean distance. Nguyen and Ho (2022) amortizes the estimation of the optimal projection in the max-sliced objective, which differs from our work where we instead amortize the estimation of the optimal coupling directly. Lacombe et al. (2021) learns to predict Wasserstein barycenters of pixel images by training a convolutional networks that, given images as input, outputs their barycenters. Our work is hence a generalization of this pixel-based work to general measures – both discrete and continuous. One limitation is that the barycenter predictions do not provide the optimal couplings. Gracyk and Chen (2022) learn a neural operator, e.g. from Kovachki et al. (2021); Li et al. (2020) to amortize the solution to the PDE from the dynamic OT formulation. Bunne et al. (2022a) predict the solutions to continuous neural OT problems.

6. Conclusions

We have presented foundations for modeling and learning to solve OT problems with Meta OT by using amortized optimization to predict optimal transport plans. This works best in applications that require solving multiple OT problems with shared structure. We instantiated it to speed up entropic regularized optimal transport and unregularized optimal transport with squared cost by multiple orders of magnitude. We envision extensions of the work in: 1) Continuous settings. Learning solutions continuous OT problems is a budding topic in the community: Gracyk and Chen (2022) amortize solutions to dynamic OT problems between continuous measures, and Bunne et al. (2022a) uses a partially input-convex neural network (PICNN) from Amos et al. (2017) to predict continuous OT solutions from contextual information. Related to these, app. D presents a more general extension of Meta OT and provides a demonstration on transferring color palettes, which is shown in fig. 6. Future directions for amortizing continuous OT problems include exploring modeling (PICNN vs. a hyper-network), loss, and fine-tuning choices. 2) Meta OT models. While we mostly consider models based on hypernetworks, other meta-learning paradigms can be connected in. In the discrete setting, we only considered settings where the cost remains fixed, but the Meta OT model can also be conditioned on the cost by considering the entire cost matrix as an input (which may be too large for most models to handle), or considering a lower-dimensional parameterization of the cost that changes between the Meta OT problem instances. Another modeling dimension is the ability to capture variable-length input measures. Design decisions for this can be inspired from by VeLO (Metz et al., 2022), which learns a generic optimizer for large-scale machine learning models that can predict updates to models with 500M parameters. 3) OT algorithms. While we instantiated models on top of log-Sinkhorn, Meta OT could be built on top of other methods, and 4) OT applications that are computationally expensive and repeatedly solved, e.g. in multi-marginal and barycentric settings, or for Gromov-Wasserstein distances between metric-measure spaces.

Limitations. While we have illustrated successful applications of Meta OT, it is also important to understand the limitations that also arise in more general amortization settings: 1) Meta OT does not make previously intractable problems tractable. All of the baseline OT solvers we consider solve our problems within milliseconds or seconds. 2) Out-of-distribution generalization. Meta OT may not generate good predictions on instances that are not close to the training OT problems from the meta-distribution D over the measures and cost. If the model makes a bad prediction, one fallback option is to re-solve the instance from scratch.

Acknowledgments

We would like to thank Eugene Vinitsky, Mark Tygert, Mathieu Blondel, Maximilian Nickel, and Muhammad Izatullah for insightful comments and discussions. The core set of tools in Python (Van Rossum and Drake Jr, 1995; Oliphant, 2007) enabled this work, including Hydra (Yadan, 2019), JAX (Bradbury et al., 2018), Matplotlib (Hunter, 2007), numpy (Oliphant, 2006; Van Der Walt et al., 2011), Optimal Transport Tools (Cuturi et al., 2022), and pandas (McKinney, 2012).
References


Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua
Meta Optimal Transport


A. Selecting $\epsilon$ for MNIST

![Transport Maps for Different $\epsilon$ Values]

Figure 8. We selected $\epsilon = 10^{-2}$ for our MNIST coupling experiments as it results in transport maps that are not too blurry or sharp.

B. Additional experimental and implementation details

Our Jax source code is available at [http://github.com/facebookresearch/meta-ot](http://github.com/facebookresearch/meta-ot) and contains:

```
- meta_ot     Meta OT Python library code
  - conjugate.py  Exact conjugate solver for the continuous setting
  - data.py
  - models.py
  - utils.py
- config      Hydra configuration for the experiments (containing hyper-parameters)
  - train_discrete.py  Train Meta OT models for discrete OT
  - train_color_single.py  Train a single ICNN with W2GN between 2 images (for debugging)
  - train_color_meta.py  Train a Meta ICNN with W2GN
  - plot_mnist.py      Visualize the MNIST couplings
  - plot_world_pair.py  Visualize the spherical couplings
  - eval_color.py      Evaluate the Meta ICNN in the continuous setting
  - eval_discrete.py   Evaluate the Meta ICNN for the discrete tasks
```

Connecting to the data is one difficulty in running the experiments. The easiest experiment to re-run is the MNIST one, which will automatically download the dataset:

```
1. ./train_discrete.py  # Train the model, outputting to <exp_dir>
2. ./eval_discrete.py <exp_dir>  # Evaluate the learned models
3. ./plot_mnist.py <exp_dir>  # Produce further visualizations
```
B.1. Hyper-parameters

We briefly summarize the hyper-parameters we used for training, which we did not extensively tune. In the discrete setting, we use the same hyper-parameters for the MNIST and spherical settings.

<table>
<thead>
<tr>
<th>Table 5. Discrete OT hyper-parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Batch size</td>
</tr>
<tr>
<td>Number of training iterations</td>
</tr>
<tr>
<td>MLP Hidden Sizes</td>
</tr>
<tr>
<td>Adam learning rate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6. Continuous OT hyper-parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Meta batch size (for $\alpha, \beta$)</td>
</tr>
<tr>
<td>Inner batch size (to estimate $\mathcal{L}$)</td>
</tr>
<tr>
<td>Cycle loss weight ($\gamma$)</td>
</tr>
<tr>
<td>Adam learning rate</td>
</tr>
<tr>
<td>$\ell_2$ weight penalty</td>
</tr>
<tr>
<td>Max grad norm (for clipping)</td>
</tr>
<tr>
<td>Number of training iterations</td>
</tr>
<tr>
<td>Meta ICNN Encoder ResNet18</td>
</tr>
<tr>
<td>Encoder output size (both measures)</td>
</tr>
<tr>
<td>Meta ICNN Decoder Hidden Sizes</td>
</tr>
</tbody>
</table>

B.2. Sinkhorn convergence times, varying thresholds

In the main paper, table 1 reports the runtime of Sinkhorn to reach a convergence threshold of the marginal error being below a tolerance of $10^{-23}$. Tables 7 and 8 report the results from sweeping over other thresholds and show that Meta OT’s initialization is consistently able to help.

| Table 7. Sinkhorn runtime to reach a thresholded marginal error on MNIST. |
|-------------------|-----------------|-----------------|-----------------|-----------------|
| Initialization   | Threshold=$10^{-2}$ | Threshold=$10^{-3}$ | Threshold=$10^{-4}$ | Threshold=$10^{-5}$ |
| Zeros            | 4.5 $\cdot$ 10^{-3} | 1.5 $\cdot$ 10^{-3} | 7.7 $\cdot$ 10^{-3} | 1.2 $\cdot$ 10^{-3} | 1.1 $\cdot$ 10^{-2} | 1.8 $\cdot$ 10^{-3} | 1.5 $\cdot$ 10^{-2} | 2.3 $\cdot$ 10^{-3} |
| Gaussian         | 4.1 $\cdot$ 10^{-3} | 1.2 $\cdot$ 10^{-3} | 7.7 $\cdot$ 10^{-3} | 1.4 $\cdot$ 10^{-3} | 1.1 $\cdot$ 10^{-2} | 1.7 $\cdot$ 10^{-3} | 1.4 $\cdot$ 10^{-2} | 2.4 $\cdot$ 10^{-3} |
| Meta OT          | 2.3 $\cdot$ 10^{-3} | 9.2 $\cdot$ 10^{-3} | 3.9 $\cdot$ 10^{-3} | 1.6 $\cdot$ 10^{-3} | 6.7 $\cdot$ 10^{-3} | 1.4 $\cdot$ 10^{-3} | 1.0 $\cdot$ 10^{-2} | 2.4 $\cdot$ 10^{-3} |

| Table 8. Sinkhorn runtime to reach a thresholded marginal error on the spherical transport problem. |
|-------------------|-----------------|-----------------|-----------------|-----------------|
| Initialization   | Threshold=$10^{-2}$ | Threshold=$10^{-3}$ | Threshold=$10^{-4}$ | Threshold=$10^{-5}$ |
| Zeros            | 8.8 $\cdot$ 10^{-1} $\pm$ 1.3 $\cdot$ 10^{-1} | 1.4 $\pm$ 1.9 $\cdot$ 10^{-1} | 2.1 $\pm$ 3.6 $\cdot$ 10^{-1} | 2.8 $\pm$ 5.6 $\cdot$ 10^{-1} |
| Gaussian         | 5.6 $\cdot$ 10^{-1} $\pm$ 9.9 $\cdot$ 10^{-2} | 1.1 $\pm$ 2.0 $\cdot$ 10^{-1} | 1.7 $\pm$ 3.5 $\cdot$ 10^{-1} | 2.4 $\pm$ 5.4 $\cdot$ 10^{-1} |
| Meta OT          | 7.8 $\cdot$ 10^{-2} $\pm$ 3.4 $\cdot$ 10^{-2} | 0.44 $\pm$ 1.5 $\cdot$ 10^{-1} | 0.97 $\pm$ 3.2 $\cdot$ 10^{-1} | 1.7 $\pm$ 6.8 $\cdot$ 10^{-1} |
B.3. Experimental runtimes and convergence

App. B.3 shows the convergence during training of Meta OT models in the discrete and continuous settings over 10 trials on our single Quadro GP100 GPU. The MNIST models are consistently trained to optimality within 2 minutes (!) while the continuous model takes a few hours to train.

Figure 9. Convergence of Meta OT models during training, reported over iterations and wall-clock time. We run each experiment for 10 trials with different seeds and report each trial as a line.
C. Cross-domain experimental results

![Cross-domain experimental results](image)

**Figure 10.** Cross-domain experiments evaluating how well a model trained on one dataset generalizes to another dataset. Notably, we are able to train only on a uniform distribution and transfer reasonable initializations to the image datasets. This indicates that training larger-scale Meta OT models for more general classes of discrete OT problems may be able to provide a fast and reasonable initialization.

D. More information: Meta OT between continuous measures

D.1. Meta ICNN Diagram

![Meta ICNN Diagram](image)

**Figure 11.** A Meta ICNN for image-based input measures. A shared ResNet processes the input measures $\alpha$ and $\beta$ into latents $z$ that are decoded with an MLP into the parameters $\varphi$ of an ICNN dual potential $\psi_{\varphi}$. The derivative of the ICNN provides the transport map $^\varphi T$.

D.2. Other models for continuous OT

We explored a hyper-network model because it is conceptually the most similar to predicting the optimal dual variables in the continuous setting and results in rapid predictions. However, it may not scale well to predicting high-dimensional parameters of ICNNs. This section presents two alternatives based on MAML (Finn et al., 2017) and neural processes (Garnelo et al., 2018b;a), and conditional OT maps (Bunne et al., 2022a).
D.2.1. Optimization-based meta-learning (MAML-inspired)

The model-agnostic meta-learning setup proposed in MAML (Finn et al., 2017) could also be applied in the Meta OT setting to learn an adaptable initial parameterization. In the continuous setting, one initial version would take a parameterized dual potential model \( \psi_{\phi}(x) \) and seek to learn an initial parameterization \( \phi_0 \) so that optimizing a loss such as the W2GN loss \( L \) from eq. (13) results in a minimal \( L(\varphi_K) \) after adapting the model for \( K \) steps. Formally, this would optimize:

\[
\arg\min_{\varphi_0} L(\varphi_K) \quad \text{where} \quad \varphi_{t+1} = \varphi_t - \nabla \varphi L(\varphi_t)
\]  

Tancik et al. (2021) explores similar learned initializations for coordinate-based neural implicit representations for 2D images, CT scan reconstruction, and 3d shape and scene recovery from 2D observations.

Challenges for Meta OT. The transport maps given by \( T = \nabla \psi \) can significantly vary depending on the input measures \( \alpha, \beta \). We found it difficult to learn an initialization that can be rapidly adapted, and optimizing eq. (20) is more computationally expensive than eq. (18) as it requires unrolling through many evaluations of the transport loss \( L \). And, we found that only learning to predict the optimal parameters with eq. (18), conditional on the input measures, and then fine-tuning with W2GN to be stable.

Advantages for Meta OT. Exploring MAML-inspired methods could further incorporate the knowledge that the model’s prediction is going to be fine-tuned into the learning process. One promising direction we did not try could be to integrate some of the ideas from LEO (Rusu et al., 2019) and CAVIA (Zintgraf et al., 2019), which propose to learn a latent space for the parameters where the initialization is also conditional on the input.

D.2.2. Neural process and conditional Monge maps

The (conditional) neural process models considered in Garnelo et al. (2018b;a) can also be adapted for the Meta OT setting, and is similar to the model proposed in Bunne et al. (2022a). In the continuous setting, this would result in a dual potential that is also conditioned on a representation of the input measures, e.g. \( \psi_{\phi}(x; z) \) where \( z := f_{\phi}^\text{emb}(\alpha, \beta) \) is a learned embedding of the input measures that is learned with the parameters of \( \psi \). This could be formulated as

\[
\arg\min_{\varphi} \mathbb{E}_{(\alpha, \beta) \sim D} L(\varphi, f_{\phi}^\text{emb}(\alpha, \beta)),
\]

where \( L \) modifies the model used in the loss eq. (13) to also be conditioned on the context extracted from the measures.

Challenges for Meta OT. This raises the issue on best-formulating the model to be conditional on the context. One way could be to append \( z \) to the input point \( x \) in the domain. Bunne et al. (2022a) proposes to use the Partially Input-Convex Neural Network (PICNN) from (Amos et al., 2017) to make the model convex with respect to \( x \) and not \( z \).

Advantages for Meta OT. A large advantage is that the representation \( z \) of the measures \( \alpha, \beta \) would be significantly lower-dimensional than the parameters \( \varphi \) that our Meta OT models are predicting.
D.3. Continuous Wasserstein-2 color transfer

The following public domain images are from WikiArt:

- Distant View of the Pyramids by Winston Churchill (1921)
- Charing Cross Bridge, Overcast Weather by Claude Monet (1900)
- Houses of Parliament by Claude Monet (1904)
- October Sundown, Newport by Childe Hassam (1901)
- Landscape with House at Ceret by Juan Gris (1913)
- Irises in Monet’s Garden by Claude Monet (1900)
- Crystal Gradation by Paul Klee (1921)
- Senecio by Paul Klee (1922)
- Váza s květinami by Josef Capek (1914)
- Sower with Setting Sun by Vincent van Gogh (1888)
- Three Trees in Grey Weather by Claude Monet (1891)
- Vase with Daisies and Anemones by Vincent van Gogh (1887)
Figure 12. Meta ICNN (initial prediction). The sources are given in the beginning of app. D.3.
$\alpha$  $\beta$  $T_{\#\alpha}$  $T_{\#}^{-1}\beta$

*Figure 13.* Meta ICNN + W2GN fine-tuning. The sources are given in the beginning of app. D.3.
Figure 14. W2GN (final). The sources are given in the beginning of app. D.3.