
What is the Right Notion of Distance between Predict-then-Optimize Tasks? (Supplementary Material)

A PROOF OF PROPOSITION 5.2

To demonstrate that the OTD³, $d_{OT}(\cdot, \cdot; c_{PtO})$ is a valid metric, it is sufficient to verify that the ground cost function c_{PtO} used in the optimal transport problem is a metric on $\mathcal{X} \times \mathcal{Y} \times \Omega$. If c_{PtO} is indeed a metric, then $d_{OT}(\cdot, \cdot; c_{PtO})$ corresponds to the Wasserstein distance Villani [2008]. In Equation 7, $d_{OT}(\cdot, \cdot; c_{PtO})$ is defined as a convex combination of $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$, which are metrics on \mathcal{X} and \mathcal{Y} respectively, and the decision quality disparity l_q . To show that c_{PtO} is a metric, it suffices to show that l_q satisfies the four metric properties: non-negativity, identity of indiscernibles, symmetry, and the triangle inequality. If l_q does not individually satisfy these properties, we must demonstrate that the convex combination of $d_{\mathcal{X}}$, $d_{\mathcal{Y}}$, and l_q satisfies these properties collectively under the assumption that $\alpha_X, \alpha_Y, \alpha_W > 0$.

First, l_q is clearly non-negative because it is defined as an absolute value. It is symmetric in the convex combination of c_{PtO} because it is taken as the absolute difference between two decision qualities with fixed true costs.

$$\begin{aligned} l_q(z, z'; y', y') &= |q(z; y') - q(z'; z')| \\ &= |q(z'; y') - q(z; z')| \\ &= l_q(z', z; y', y') \end{aligned}$$

Moreover, l_q satisfies triangle inequality due to the triangle inequality property of the absolute value.

$$\begin{aligned} &l_g(z_1, z_2; y_1, y_2) + l_g(z_2, z_3; y_2, y_3) \\ &= |g(z_1; y_1) - g(z_2; y_2)| + |g(z_2; y_2) - g(z_3; y_3)| \\ &\leq |g(z_1; y_1) - g(z_2; y_2) + g(z_2; y_2) - g(z_3; y_3)| \\ &= |g(z_1; y_1) - g(z_3; y_3)| \\ &= l_g(z_1, z_3; y_1, y_3) \end{aligned}$$

Lastly, while l_q might not satisfy the identity of indiscernibles in isolation (specifically, $l_q(y, y'; z, z) = 0$ does not necessarily imply $y = y'$; meaning two different decisions can lead to the same objective value), c_{PtO} does satisfy this property for $\alpha_X, \alpha_Y, \alpha_W > 0$. If $(x, y, z) = (x', y', z')$, then $l_g(z, z'; y', y') = |g(z; y) - g(z'; y)| = 0$ because $z = z'$ implies $g(z; y) = g(z'; y)$ and hence $c_{PtO}((x, y, z), (x', y', z')) = 0$. Conversely, if $c_{PtO}((x, y, z), (x', y', z')) = 0$, then $d_{\mathcal{X}}(x, x') = 0$, $d_{\mathcal{Y}}(y, y') = 0$, and $l_q(y, y'; z, z) = 0$ because $\alpha_X, \alpha_Y, \alpha_W > 0$. Since $d_{\mathcal{Y}}(y, y') = 0$ implies $y = y'$ (because $d_{\mathcal{Y}}$ is a metric), it follows that $w^*(y) = w^*(y')$ and hence $z = z'$.

Therefore, c_{PtO} satisfies the identity of indiscernibles. Consequently, since l_q satisfies non-negativity, symmetry, and the triangle inequality, and since c_{PtO} satisfies the identity of indiscernibles, $d_{OT}(\cdot, \cdot; c_{PtO})$ is indeed a valid metric with c_{PtO} a valid metric on $\mathcal{X} \times \mathcal{Y} \times \Omega$.

B PREAMBLE FOR THEOREM 5.5

B.1 VALIDITY ASSUMPTION 5.3

Assumption 5.3 imposes a specific structure on the downstream optimization problem by assuming that the decision quality function has a bounded rate of change with respect to both the predicted and true cost vectors. This is a reasonable assumption for certain downstream optimization tasks, as highlighted in the following lemmas.

Lemma B.1. *If $M(\cdot)$ is a convex program with a strongly convex objective and constraints with independent derivatives (Linear Independence Constraint Qualification (LICQ)), Assumption 5.3 holds.*

The strong convexity of the objective ensures that the gradient is Lipschitz continuous, while the LICQ guarantees that the optimal solutions depend continuously on the parameters. By the smoothness of the objective and the continuity of the optimal solutions, the difference in the decision quality function q between two sets of parameters and their corresponding optimal solutions can be bounded by a linear combination of the distances between the parameters and the distances between the optimal solutions.

Lemma B.2. *If $M(\cdot)$ has a linear optimization objective with a strongly convex feasible region, Assumption 5.3 holds.*

When $M(\cdot)$ has a linear optimization objective and a strongly convex feasible region, the decision quality function q satisfies the k_1, k_2 -Lipschitz property. The linearity of the objective ensures that changes in the parameters lead to proportional changes in the objective value, while the strong convexity of the feasible region guarantees that the optimal solutions are unique and vary smoothly with respect to the parameters. This smooth dependence, combined with the linear structure of the objective, implies that the difference in q between two sets of parameters and their corresponding optimal solutions can be bounded by a linear combination of the distances between the parameters and the distances between the optimal solutions.

B.2 LIPSCHITZNESS OF THE DECISION QUALITY DISPARITY FUNCTION

To establish the bound presented in Theorem 5.5, we rely on the fact that l_g is k_1, k_2 -Lipschitz under Assumption 5.3. The following proposition demonstrates that l_g indeed satisfies the Lipschitz condition given this assumption.

Proposition B.3. *If g , the objective function of the downstream optimization problem, is k_1, k_2 -Lipschitz (Assumption 5.3), then l_g is also k_1, k_2 -Lipschitz.*

Proof.

$$\begin{aligned} & |l_g(z, z_1; y, y_1) - l_g(z, z_2; y, y_2)| \\ &= ||g(z; y) - g(z_1; y_1)| - |g(z; y) - g(z_2; y_2)|| \\ &\leq |g(z; y) - g(z_1; y_1) - g(z; y) + g(z_2; y_2)| \end{aligned} \tag{9}$$

$$\begin{aligned} &= |g(z_2; y_2) - g(z_1; y_1)| \\ &= |g(z_2; y_2) - g(z_1; y_2) + g(z_1; y_2) - g(z_1; y_1)| \\ &\leq |g(z_2; y_2) - g(z_1; y_2)| + |g(z_1; y_2) - g(z_1; y_1)| \end{aligned} \tag{10}$$

$$\leq k_1 \|z_1 - z_2\| + k_2 \|y_1 - y_2\| \tag{11}$$

Inequalities (9) and (10) are a result of the triangle inequality of the absolute value. Inequality (11) is due to the $k_1 - k_2$ -lipschitzness of g . \square

C PROOF OF THEOREM 5.5

Theorem C.1. *Suppose Assumption 5.3 holds. For a feature space \mathcal{X} , a label space \mathcal{Y} , and a decision set Ω , let $\mathcal{W} := \mathcal{X} \times \mathcal{Y} \times \Omega$. Let \mathcal{P}_S and \mathcal{P}_T be the source and target distributions over $\mathcal{X} \times \mathcal{Y}$ respectively. For any labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$, let \mathcal{P}_T^f and \mathcal{P}_S^* be distributions over \mathcal{W} given by $\mathcal{P}_T^f := (x, y, w^*(f(x)))_{(x,y) \sim \mathcal{P}_T}$ and*

$\mathcal{P}_S^* := (x, y, w^*(y))_{(x,y) \sim \mathcal{P}_S}$. For a ground cost function of the form

$$c_{\mathcal{P}_{tO}}^\alpha((x, y, z), (x', y', z')) = \alpha_X d_{\mathcal{X}}(x, x') + \alpha_Y d_{\mathcal{Y}}(y, y') + \alpha_W l_g(z, z'; y', y'),$$

let Π^* be the coupling that minimizes the OT problem with ground cost $c_{\mathcal{P}_{tO}}^\alpha$ between \mathcal{P}_T^f and \mathcal{P}_S^* . Let \tilde{f} be a labeling function that is ϕ -Lipschitz transferable w.r.t. Π^* . We assume \mathcal{X} is bounded by K and \tilde{f} is l -Lipschitz, such that $|\tilde{f}(x_1) - \tilde{f}(x_2)| \leq 2lK = L$. Then, for all $\lambda > 0$ and $\alpha_W \in (0, 1)$ such that $(\lambda k_1 + k_2 + 1)\alpha_W = 1$, and $\alpha_X = \lambda k_1 \alpha_W$ and $\alpha_Y = k_2 \alpha_W$, we have with probability at least $1 - \delta$ that:

$$\text{err}(f; q_{\text{reg}}, \mathcal{P}_T) \leq \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_S) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_T) + k_1 L \phi(\lambda) + (1/\alpha_W) d_{OT}(\mathcal{P}_T^f, \mathcal{P}_S^*; c_{\mathcal{P}_{tO}}^\alpha)$$

Proof.

$$\begin{aligned} \text{err}(f; q_{\text{reg}}, \mathcal{P}_T) &= \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(f(x)), w^*(y); y, y) \\ &\leq \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(f(x)), w^*(\tilde{f}(x)); y, y) + \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(\tilde{f}(x)), w^*(y); y, y) \quad (12) \\ &= \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(\tilde{f}(x)), w^*(f(x)); y, y) + \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(\tilde{f}(x)), w^*(y); y, y) \quad (13) \\ &= \mathbb{E}_{(x,y,z) \sim \mathcal{P}_T^f} l_g(w^*(\tilde{f}(x)), z; y, y) + \mathbb{E}_{(x,y) \sim \mathcal{P}_T} l_g(w^*(\tilde{f}(x)), w^*(y); y, y) \quad (14) \\ &= \mathbb{E}_{(x,y,z) \sim \mathcal{P}_T^f} l_g(w^*(\tilde{f}(x)), z; y, y) - \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_S) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_S) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_T) \\ &= \mathbb{E}_{(x,y,z) \sim \mathcal{P}_T^f} l_g(w^*(\tilde{f}(x)), z; y, y) - \mathbb{E}_{(x,y,z) \sim \mathcal{P}_S^*} l_g(w^*(\tilde{f}(x)), z; y, y) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_S) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_T) \\ &\leq |\mathbb{E}_{(x,y,z) \sim \mathcal{P}_T^f} l_g(w^*(\tilde{f}(x)), z; y, y) - \mathbb{E}_{(x,y,z) \sim \mathcal{P}_S^*} l_g(w^*(\tilde{f}(x)), z; y, y)| + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_S) + \text{err}(\tilde{f}; q_{\text{reg}}, \mathcal{P}_T) \end{aligned}$$

Inequality (12) uses the fact that $l_g(\cdot; y, y)$ satisfies the triangle inequality and line (13) is due to the symmetry of $l_g(\cdot; y, y)$ for any $y \in \mathcal{Y}$. Line (14) comes from the fact that $\mathcal{P}_T^f := (x, f(x), y)_{(x,y) \sim \mathcal{P}_T}$. We continue by bounding the first term.

$$\begin{aligned} &|\mathbb{E}_{(x,y,z) \sim \mathcal{P}_T^f} l_g(w^*(\tilde{f}(x)), z; y, y) - \mathbb{E}_{(x,y,z) \sim \mathcal{P}_S^*} l_g(w^*(\tilde{f}(x)), z; y, y)| \\ &= \left| \int_{\mathcal{W}} l_g(w^*(\tilde{f}(x)), z; y, y) (\mathcal{P}_T^f(X = x, Y = y, Z = z) - \mathcal{P}_S^*(X = x, Y = y, Z = z)) dx dy dz \right| \\ &= \left| \int_{\mathcal{W}} l_g(w^*(\tilde{f}(x)), z; y, y) d\Pi^*((x_s, y_s, z_s), (x_t, y_t, z_t^f)) \right| \\ &\leq \int_{\mathcal{W}^2} |l_g(\tilde{z}_t, z_t^f; y_t, y_t) - l_g(\tilde{z}_s, z_s; y_s, y_s)| d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \quad (15) \\ &\leq \int_{\mathcal{W}^2} |l_g(\tilde{z}_t, z_t^f; y_t, y_t) - l_g(\tilde{z}_s, z_t^f; y_s, y_t)| + |l_g(\tilde{z}_s, z_t^f; y_s, y_t) - l_g(\tilde{z}_s, z_s; y_s, y_s)| d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \quad (16) \\ &\leq \int_{\mathcal{W}^2} k_1 d_{\mathcal{Y}}(\tilde{f}(x_t), \tilde{f}(x_s)) + k_2 d_{\mathcal{Y}}(y_t, y_s) + |l_g(\tilde{z}_s, z_t^f; y_s, y_t) - l_g(\tilde{z}_s, z_s; y_s, y_s)| d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \quad (17) \\ &\leq k_1 L \phi(\lambda) + \int_{\mathcal{W}^2} \lambda k_1 d_{\mathcal{X}}(x_t, x_s) + k_2 d_{\mathcal{Y}}(y_t, y_s) + |l_g(\tilde{z}_s, z_t^f; y_s, y_t) - l_g(\tilde{z}_s, z_s; y_s, y_s)| d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \quad (18) \\ &\leq k_1 L \phi(\lambda) + \int_{\mathcal{W}^2} \lambda k_1 d_{\mathcal{X}}(x_t, x_s) + k_2 d_{\mathcal{Y}}(y_t, y_s) + l_g(z_t^f, z_s; y_s, y_s) d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \end{aligned}$$

From line (15) onwards we take $\mathbf{w}_s := (x_s, y_s, y_s)$, $\mathbf{w}_t^f := (x_t, y_t^f, y_t)$ and $\tilde{z}_s = w^*(\tilde{f}(x_s))$, $\tilde{z}_t = w^*(\tilde{f}(x_t))$ for ease of notation. Given a weight α_W , we now normalize the last term such that the ground cost function is a convex combination of $d_{\mathcal{X}}$, d_{ym} and l_g .

$$\begin{aligned} &\int_{\mathcal{W}^2} \lambda k_1 d_{\mathcal{X}}(x_t, x_s) + k_2 d_{\mathcal{Y}}(y_t, y_s) + l_g(z_t^f, z_s; y_s, y_s) d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \\ &= \frac{1}{\alpha_W} \int_{\mathcal{W}^2} \lambda k_1 \alpha_W d_{\mathcal{X}}(x_t, x_s) + k_2 \alpha_W d_{\mathcal{Y}}(x_t, x_s) + \alpha_W l_g(z_t^f, z_s; y_s, y_s) d\Pi^*(\mathbf{w}_s, \mathbf{w}_t^f) \\ &= \frac{1}{\alpha_W} d_{OT}(\mathcal{P}_T^f, \mathcal{P}_S^*; c_{\mathcal{P}_{tO}}^\alpha) \end{aligned}$$

D EXPERIMENTAL SETTINGS DETAILS

D.1 LINEAR MODEL TOP-K Shah et al. [2022]

PtO task description. The Linear Model Top-K setting is a learning task designed to evaluate decision-focused learning approaches in scenarios where the true relationship between features and outcomes is nonlinear, yet the model used for prediction is constrained to be linear. Specifically, the objective is to train a linear model to perform top- K selection when the underlying data is generated by a cubic polynomial function. This controlled setup enables an assessment of how well decision-focused methods handle model misspecification. The predict-then-optimize (PtO) task in this setting is defined as follows:

Predict: Given the feature $x_n \sim \mathcal{P}_X$, where $\mathcal{P}_X = \text{Unif}[-1, 1]$, of a resource n , the prediction task consists of using a linear model to predict the corresponding utility \hat{y}_n , where the true utility $y_n = p(x_n)$ is a cubic polynomial in x_n . The predictions for N resources are aggregated into a vector $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_N]$, where each element corresponds to the predicted utility of a resource.

Optimize: The optimization task involves selecting the K out of N resources with the highest utility. This corresponds to solving the optimization problem $M(\hat{\mathbf{y}}) = \max_{\mathbf{z} \in [0,1]^N} \{\mathbf{z} \cdot \sigma_x(\hat{\mathbf{y}})\}$ such that $\|\mathbf{z}\|_0 = K$, where σ_x is the permutation that orders $\hat{\mathbf{y}}$ in ascending order of $\mathbf{x} = [x_1, \dots, x_N]$.

Synthetic distribution shift We introduce synthetic distribution shifts to create a scenario for transfer learning. We modify the original feature-label distribution $\mathcal{P} = (\text{Id}, p)_* U[-1, 1]$. Specifically, for various values of $\gamma \in [0, 1.3]$, we define the feature-label distributions $\mathcal{P}_\gamma = (\text{Id}, p_\gamma)_* U[-1, 1]$ where $p_\gamma(x_n) = 10(x_n^3 - \gamma x_n)$, using $\mathcal{P}_{0.65}$ as the target distribution.

Training details We use the implementation from Shah et al. [2022]¹ to train models by setting `loss="DFL"`. This implementation uses an entropy regularized Top-K loss function proposed by Xie et al. [2020] that reframes the Top-K problem with entropy regularization as an optimal transport problem, enabling end-to-end learning.

D.2 WARCRAFT SHORTEST PATH Pogančić et al. [2020]

PtO task description. This setting involves finding the minimum-cost path on $d \times d$ RGB grid maps from the Warcraft II tileset dataset, where each pixel represents terrain with an unknown traversal cost. The task is to first predict these costs from an input image and then determine the shortest path from the top-left to the bottom-right corner based on the predicted cost map. This benchmark is particularly notable because it involves image inputs, a modality not widely explored in other shortest-path learning tasks. Following Pogančić et al. [2020], we use 96×96 RGB images as input, with the shortest path being computed on a coarser 12×12 grid representation of the predicted costs.

Predict: Given the feature $x_n \in \mathbb{R}^{d \times d \times 3}$, predict the travel cost grid $\hat{\mathbf{y}}_n \in \mathbb{R}^{p \times p}$.

Optimize: Solve a shortest-path problem over the predicted cost grid. Specifically, find the path \mathbf{z} that minimizes the total traversal cost: $M(\hat{\mathbf{y}}) = \min_{\mathbf{z} \in [0,1]^p} \{\mathbf{z} \cdot \hat{\mathbf{y}}\}$ subject to boundary conditions $\mathbf{z}_{0,0} = \mathbf{z}_{p,p} = 1$ and connectivity constraints ensuring that \mathbf{z} represents a valid path from the top-left to the bottom-right corner.

Synthetic distribution shift. The original distribution \mathcal{P} , which we treat as the target distribution, is defined over $\mathbb{R}^{d \times d} \times \mathbb{R}^{p \times p}$, where $d = 96$ and $p = 12$. Here, $\mathbb{R}^{d \times d}$ represents the feature space depicting maps, while $\mathbb{R}^{p \times p}$ represents the traveling costs on these maps. We induce a target shift for \mathcal{P}_γ by uniformly sampling the costs for different pixel classes from the same range as \mathcal{P} ($[0.8, 9.2]$ for the Warcraft II tileset dataset). Figure 6 illustrates the costs coming from two different distributions over one same feature while highlighting the different decisions (shortest path) that these costs yield.

Training details. We use `pyepo`² implementation with SPO+ loss function on a truncated ResNet-18 consisting of the first five layers, followed by a final convolutional layer that reduces the number of output channels to one. Finally, we use an adaptive max-pooling layer to obtain a fixed $p \times p$ spatial resolution, allowing for a structured representation of the extracted features.

¹github.com/sanketkshah/LODLs

²github.com/khalil-research/PyEPO

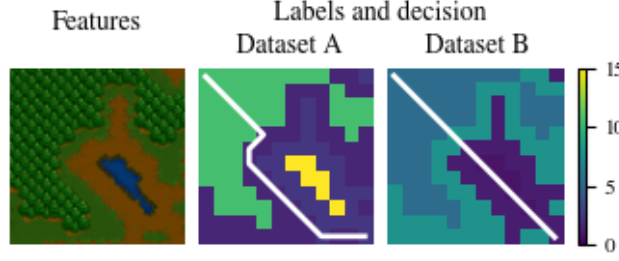


Figure 6: *Synthetic distribution shift in Warcraft Shortest Path.* The white line illustrates the decision, corresponding to the shortest path, on dataset A (center) and dataset B (right) for a sample with the same features (left map).

D.3 INVENTORY STOCK PROBLEM Donti et al. [2017]

PtO task description. In this problem a company must determine the optimal order quantity z of a product to minimize costs given a stochastic demand y , which is influenced by observed features x . The cost structure includes both linear and quadratic costs for the amount of product ordered, as well as different linear and quadratic costs for over-orders $[z - y]^+$ and under-orders $[y - z]^+$. The objective function is:

$$f_{\text{stock}}(y, z) = c_0 z + \frac{1}{2} q_0 z^2 + c_b [y - z]^+ + \frac{1}{2} q_b ([y - z]^+)^2 + c_h [z - y]^+ + \frac{1}{2} q_h ([z - y]^+)^2 \quad (19)$$

where $[v]^+ \equiv \max\{v, 0\}$. In our paper, we use $c_0 = 30, q_0 = 10, c_b = 10, q_b = 2, c_h = 30, q_h = 25$. For a given probability model $p(y|x; \theta)$, the proxy stochastic programming problem can be formulated as: $\underset{z}{\text{minimize}} \quad \mathbf{E}_{y \sim p(y|x; \theta)} [f_{\text{stock}}(y, z)]$.

To simplify the setting, we assume that the demands are discrete, taking on values d_1, \dots, d_k with probabilities (conditional on x) $(p_\theta)_i \equiv p(y = d_i|x; \theta)$. Thus, our stochastic programming problem can be succinctly expressed as a joint quadratic program:

$$\begin{aligned} \underset{z \in \mathbb{R}, z_b, z_h \in \mathbb{R}^k}{\text{minimize}} \quad & \left\{ c_0 z + \frac{1}{2} q_0 z^2 + \sum_{i=1}^k (p_\theta)_i \left(c_b (z_b)_i + \frac{1}{2} q_b (z_b)_i^2 + c_h (z_h)_i + \frac{1}{2} q_h (z_h)_i^2 \right) \right\} \\ \text{subject to} \quad & d - z\mathbf{1} \leq z_b, \quad z\mathbf{1} - d \leq z_h, \quad z, z_b, z_h \geq 0 \end{aligned} \quad (10)$$

Synthetic distribution shift We generate problem instances by randomly sampling $x \in \mathbb{R}^n$ and then generating $p(y|x; \theta)$ according to $p(y|x; \theta) \propto \exp((\theta^T x)^2)$. We introduce distribution shifts for both x and y . Specifically, x is sampled from a Gaussian distribution where the mean is sampled from $U[-0.5, 0.5]$, and θ is also sampled from a Gaussian distribution.

Training details We use the implementation from Donti et al. [2017]³ following their Inventory Stock Problem experiments.

E OTD³ IMPLEMENTATION DETAILS

Our implementation of the OTD³ relies on the POT⁴ package. The computation of dataset distance involves two main steps:

1. Computing Pairwise Pointwise Distances:

We first compute the pairwise distances between samples in the source and target datasets. This involves calculating distances separately for features, labels, and decisions, weighted according to the selected component weights $(\alpha_X, \alpha_Y, \alpha_W)$. Feature and label distances are computed using standard metric spaces (e.g., Euclidean or cosine distance), while decision distances are computed using decision quality disparity.

³github.com/locuslab/e2e-model-learning

⁴pythonot.github.io/

2. Solving the Optimal Transport Problem:

Given the computed pairwise distances, we compute the dataset distance using Earth Mover’s Distance (EMD) via POT’s `emd` solver. EMD finds the exact optimal transport plan, making it well-suited for capturing true correspondences between source and target datasets without introducing regularization bias. This approach was computationally feasible in our experiments due to the relatively small dataset sizes.

Additionally, for experiments involving hyperparameter tuning, we evaluate multiple weight combinations on a predefined grid and select the setting that maximizes correlation with regret transferability.

F ADDITIONAL RESULTS

F.1 SELECTING SOURCE DATASETS FOR TRANSFER LEARNING

In Section 7.1 we analyzed the correlation between dataset distance and transferability in PtO. The plots presented in Figure 7 show this correlation for the Linear Model TopK setting and the Inventory Stock problem for two weighting profiles: the best weighting profile where no decisions are included [left] and the best weighting profile with decisions included [right]. For both setting, decisions improve predictability of transferability. This improvement is less pronounced in the Inventory Stock problem.

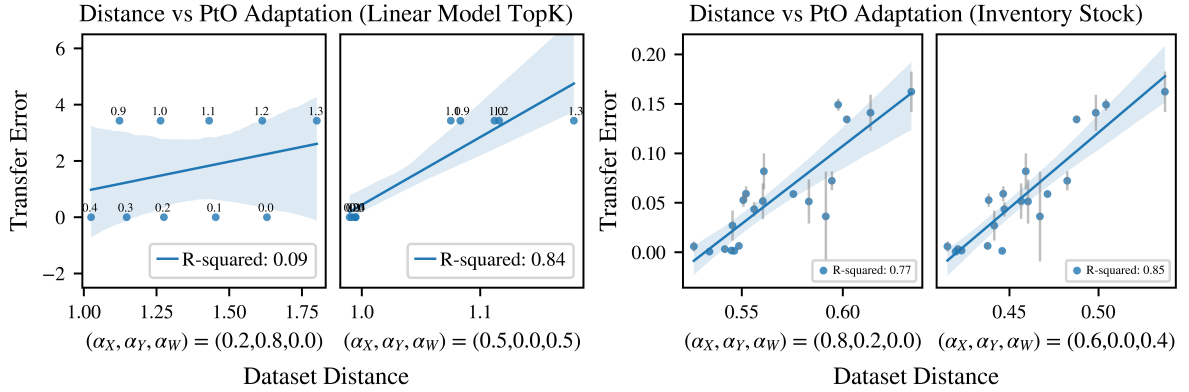


Figure 7: *Distance vs Adaptation*. OT distance for the best feature-label and feature-label-decision weighting against the regret transferability.

To illustrate the relationship between label space differences $d_y(y, y')$ and decision space differences $l_q(y, y', z, z')$ in different PtO tasks, we provide the following visualizations. Figure 8b shows this correlation for the Inventory Stock problem, while Figure 8a presents the same analysis for the Warcraft domain.

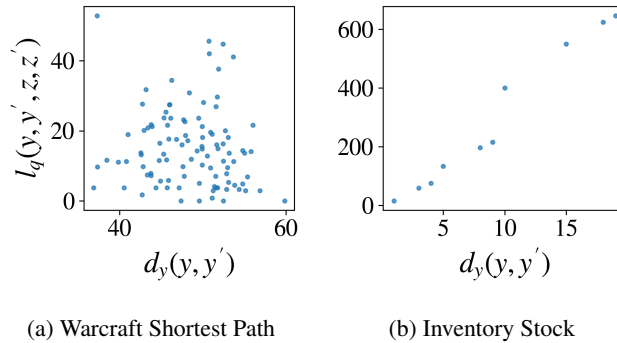


Figure 8: Difference in labels against difference in decisions