Order Constraints in Optimal Transport

Fabian Lim 1  Laura Wynter 1  Shiau Hong Lim 1

Abstract

Optimal transport is a framework for comparing measures whereby a cost is incurred for transporting one measure to another. Recent works have aimed to improve optimal transport plans through the introduction of various forms of structure. We introduce novel order constraints into the optimal transport formulation to allow for the incorporation of structure. We define an efficient method for obtaining explainable solutions to the new formulation that scales far better than standard approaches. The theoretical properties of the method are provided. We demonstrate experimentally that order constraints improve explainability using the e-SNLI (Stanford Natural Language Inference) dataset that includes human-annotated rationales as well as on several image color transfer examples.

1. Introduction

Optimal transport (OT) is a framework for comparing measures whereby a cost is incurred for transporting one measure to another. Optimal transport enjoys both significant theoretical interest and widespread applicability (Villani, 2008; Peyré & Cuturi, 2019). Recent work (Swanson et al., 2020; Alvarez-Melis et al., 2018) aims to improve the interpretability of optimal transport plans through the introduction of structure. Structure can take many forms: text documents where context is given (Alvarez-Melis et al., 2020), images with certain desired features (Shi et al., 2020; Liu et al., 2021), fairness properties (Laclau et al., 2021), or even patterns in RNA (Forrow et al., 2019). Swanson et al. (2020) postulate that structure can be estimated by sparsi-fying optimal transport plans.

Motivated by advances in sparse model learning, order statistics and isotonic regression (Paty et al., 2020), we introduce novel order constraints (OC) into the optimal transport formulation so as to allow for more complex structure to be readily added to optimal transport plans. We show theoretically that, for convex cost functions with efficiently computable gradients, the resulting order-constrained optimal transport problem can be solved using a form of ADMM (Boyd et al., 2011) and can be δ-approximated efficiently. We further derive computationally efficient lower bounds for our formulation. Specifically, when the structure is not provided in advance, we show how it can be estimated. The bounds allow the use of an explainable method for identifying the most important constraints, rendered tractable through branch-and-bound. Each of these order constraints leads to an optimal transport plan computed sequentially using the algorithm we introduce. The end result is a diverse set of the most important optimal transport plans from which a user can select the plan that is preferred.

The order constraints allow context to be taken into account, explicitly, when known in advance, or implicitly, when estimated through the proposed procedure. Consider OT for document retrieval (e.g., see Kusner et al. (2015); Swanson et al. (2020)) where a user enters a text query. The multiple transport plans provided by OT with order constraints offer a diverse set of the best responses, each accounting for different possible contexts. The resulting method is weakly-supervised with only two tunable thresholds (or unsupervised if the parameters are set to default values), and is amenable to applications of OT where labels are not available to train a supervised model.

An example is provided in Fig. 2 which aims to learn the similarity between two phrases. The top box depicts the transport plan learnt by OT; the darker the line, the stronger the transport measure. The match provided by OT does not reveal the contradiction in the two phrases. The second to fourth transport plans are more human-interpretable. The second plan matches “red” clothing to the “black” pants. The third matches “piece” to “pair”, a relation not included in the standard OT plan. The last plan matches “clothing” to “pants”. The example makes use of OT with a single order constraint to better illustrate the idea of order constraints. OT with multiple order constraints can incorporate

1IBM Research, Singapore. Correspondence to: Fabian Lim <flim@sg.ibm.com>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).
Order Constraints in Optimal Transport

Fig. 1: Color transfer using OT with Order Constraints

Fig. 2: Text matching using OT with Order Constraints

all of the pairs in a single plan.

Fig. 1 illustrates using OT to transfer the color palette of a target image (Van Gogh’s “Café Terrace at Night”) to the source image of an alley. The second image from the left, obtained by standard OT, transfers yellow to bright areas and blue to dark. Our proposed order constraints allow adding context, which in this case gives desired effects such as transferring the Van Gogh night sky (second image from right) or the awning brilliant yellow (middle image) to the sky in the source image.

The contributions of this work are as follows: (1) we provide a novel formulation, Optimal Transport with Order Constraints, that allows explicitly incorporating structure to OT plans; (2) we provide an efficient method for solving the proposed OT with OC formulation related to an ADMM along with the convergence theory; (3) for cases where the order constraints are not known in advance, we propose an explainable method to estimate them using branch & bound and define bounds for use in the method, and (4) we demonstrate the benefits of using the proposed method on NLP and image color transfer tasks.

Next, we review related work. In the following section, we present the standard optimal transport formulation followed by the proposed OT with order constraints (OT with OC) formulation and a demonstration of the existence of a solution, which corresponds to contribution (1) above. Then, an algorithm leveraging the form of the polytope is defined to solve the OT with OC problem in a manner far more efficient than a standard convex optimization solver, as noted in contribution (2) above. In order to further improve explainability, when the order constraints are not known a priori, we present a branch & bound procedure to search the space for a set of diverse OT with OC plans, along with the bounds that reduce the polynomial search space, as noted in contribution (3) above. Finally, experiments are provided on the e-SNLI dataset and several image color transfer examples to demonstrate how order constraints allow for incorporating structure and improving explainability, as noted in contribution (4) above.

Related Work We consider point-point linear costs as in e.g., Cuturi (2013); Courty et al. (2017); Kusner et al. (2015); Wu et al. (2018); Yurochkin et al. (2019); Solomon (2018); Altschuler et al. (2019; 2017); Schmitzer (2019). Recently, a number of works have sought to add structure to optimal transport so as to obtain more intuitive and explainable transport plans. Alvarez-Melis et al. (2019) modeled invariances in OT, a form of regularization, to learn the most meaningful transport plan. Sparsity has been explored in OT to implicitly improve explainability. Swanson et al. (2020) used “dummy” variates to include fewer transport coefficients. Forrow et al. (2019) regularize using the OT rank to better deal with high dimensional data. Blondel et al. (2018) had a similar goal of learning sparse and potentially more interpretable transport plans through regularization and a dual formulation. We, on the other hand, propose an explicit approach to interpretability, rather than the implicit method of sparsifying solutions through regularization. (Su & Hua, 2017) add constraints to require two sequences to lie close to each other, in terms of indices; to respect sequence order, they penalize the constraints and then use the classic Sinkhorn algorithm to ob-
tains an approximate solution. Another way to add structure into OT is through dependency modeling; multi-marginal optimal transport takes this approach, wherein multiple measures are transported to each other. The formulation was shown to be generally intractable (Altschuler & Boix-Adsera, 2020a), however, and strong assumptions have to be made to solve it in polynomial time (Altschuler & Boix-Adsera, 2020b; Pass, 2015; Di Marino et al., 2015).

Standard OT can be solved in cubic time by primal-dual methods (Peyré & Cuturi, 2019) and the proposed OT with order constraints can be solved using generalized solvers with similar time complexity. The difficulty comes from the fact that there are quadratically many order constraints. Recent work (Cuturi, 2013; Schetbon et al., 2021; Altschuler et al., 2017; Goldfeld & Greenewald, 2020; Zhang et al., 2021; Lin et al., 2019; Guo et al., 2020; Jambulapati et al., 2019; Guminov et al., 2021) has favored solving OT via iterative methods which offer lower complexity at the expense of obtaining an approximate solution. In particular, (Schetbon et al., 2021) considered low-rank approximations of OT plans, and proposed a mirror descent with inner Dykstra iterations. Recent iterative methods for standard OT (Guminov et al., 2021; Jambulapati et al., 2019) ensure feasibility of the solution at each iteration through the use of a rounding algorithm proposed by (Altschuler et al., 2017). Iterative methods are essential for solving submodular OT (Alvarez-Melis et al., 2018) that would otherwise incur an exponential number of explicit constraints.

2. Order Constraints for Optimal Transport

Notation Let \( \mathbb{R} \) (resp. \( \mathbb{R}_+ \)) denote the set of real (resp. non-negative) numbers. Let \( m, n \) denote the number of rows and columns in the optimal transport problem. Vectors and matrices are in \( \mathbb{R}^n \) and \( \mathbb{R}^{m \times n} \). Let \( [n] := \{1, 2, \ldots, n\} \) and \( \{mn\} := [m] \times [n] \). Let \( i, j, k, \ell, p, q \) denote vector/matrix indices, \( 2^n \) denote the power set of \( [n] \), and \( 1_n \) and \( 0_n \) denote the all-ones vector and identity matrix, resp., of size \( n \). We denote the index sequence \( i_1, i_2, \ldots, i_k \) as \( i_{\{k\}} \). \( \mathbb{I}_S(\cdot) \) is the indicator function over set \( S \). The trace operator for square matrices is denoted by tr (\( \cdot \)) and rank (\( X_{ij} \)) gives the descending positions over the \( mn \) matrix coefficients \( X_{ij} \), or a subset thereof.² For a closed set \( C \), the Euclidean projector for a matrix \( X \in \mathbb{R}^{m \times n} \) is 

\[
\text{Proj}_C(X) = \arg \min_{X' \in C} \|Y - X'\|^2_F,
\]

where \( \| \cdot \|_F \) is the matrix Frobenius norm. For \( c \in \mathbb{R} \) let \( (c)_+ := c \) if \( c \geq 0 \) and \((c)_+ = 0 \) otherwise, and for \( X \in \mathbb{R}^{m \times n} \) let \( (X)_+ \) equal \( X \) after setting all negative coefficients to zero.

Standard OT Formulation In optimal transport, one optimizes a linear transport cost over a simplex-like polytope, \( U(a, b) \). A (balanced) optimal transport problem is given by two probability vectors \( a \) and \( b \) that each sum to 1. Each being of length \( m, n \), resp., they define a polytope:

\[
U(a, b) = \{ \Pi \in \mathbb{R}^+_{m \times n} : \Pi 1_n = a, \Pi^T 1_m = b \},
\]

and a linear cost \( D \in \mathbb{R}^{m \times n} \), where \( D \geq 0 \) and bounded. Then the optimal transport cost \( f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}_+ \) is the optimal cost of the following optimization:

\[
\min_{\Pi \in U(a, b)} f(\Pi) := \text{tr}(D^T \Pi),
\]

and the optimal value of \( \Pi \) is the transport plan. We propose to extend (1) by including order constraints to represent structure.

Proposed Formulation: OT with Order Constraints

The use of order constraints is analogous to the constraints used in isotonic, or monotonic, regression, (see De Leeuw et al., 2009). Specifically, if a particular variate \( i_{\ell j} \) is semantically or symbolically important, then its rank \( (\Pi_{i_{\ell j}}) \) should reflect that dominance. We thus enforce \( k \) order constraints by specifying a sequence of \( k \) variates in \( [mn] \), \( i_{\ell j[k]} := i_{1\ell_1 j_1}, i_{2\ell_2 j_2}, \ldots, i_{k\ell_k j_k} \) of importance, where the set of variates \( V := [mn] \setminus \{i_{\ell j[k]} : \ell \in [k]\} \):

\[
\Pi_{i_{\ell j[k]}} \geq \cdots \geq \Pi_{i_{\ell j[2]}} \geq \Pi_{i_{\ell j[1]}} \geq \Pi_{pq} \quad \text{for } pq \in V.
\]

Enforcing order constraints (2) means each variate \( i_{\ell j[k]} \) is fixed to maintain the \((k - \ell + 1)\)-th top position \( \Pi_{i_{\ell j[k]}} \) in the ordering \( \Pi_{i_{1j[1]}} \geq \Pi_{i_{2j[2]}} \geq \cdots \geq \Pi_{i_{mn[j]}} \) whilst learning the transport plan \( \Pi \); in particular \( i_{k\ell_k j_k} \) is fixed at the topmost position. Therefore, for order constraints on \( k \) variates of importance and point-to-point costs \( D \in \mathbb{R}^{m \times n} \), we extend the OT problem as follows:

\[
\inf_{\Pi \in U(a, b)} f(\Pi) := \text{tr}(D^T \Pi),
\]

s.t \( \Pi_{i_{\ell j[k]}} \geq \cdots \geq \Pi_{i_{\ell j[2]}} \geq \Pi_{i_{\ell j[1]}} \geq \Pi_{pq} \quad \text{for } pq \in V \),

for \( V \) in (2), and we assume non-negative costs \( D \geq 0 \) throughout. Let \( O_{i_{\ell j[k]}} \subseteq \mathbb{R}^+_{m \times n} = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X_{i_{\ell j[k]}}) = k - \ell + 1, \forall \ell \in [k]\} \).

Thus (3)–(4) can be compactly expressed as:

\[
\inf_{\Pi \in U(a, b) \cap O_{i_{\ell j[k]}}} f(\Pi).
\]

Problem (6) is convex with linear costs and constraints. However, \( O_{i_{\ell j[k]}} \) adds a significant number of constraints to the \( m+n \) in the standard OT formulation (1). The inf in (4) and (6) are replaced by min when feasible; we address such sufficient conditions in the following.
Feasibility of the Proposed OT with OC Problem

$U(a, b)$ is feasible since $ab^T \in U(a, b)$, (see Cuturi, 2013).

In certain instances it may be considered ambiguous to match a single position to multiple positions, consider variates $i[k]$ where $i[k] = i_1, i_2, \cdots, i_k$ (and $j[k]$) do not repeat row (or column) indices; let $I_{i[k]}(p) = 1$ if $p = i_k$ for at most one choice of $\ell$, 0 otherwise, and similarly $I_{j[k]}(q)$. We next derive conditions for the feasibility of (6) under mild assumptions.

**Proposition 2.1.** Suppose $i[k]$ and $j[k]$ do not repeat row (or column) indices. Further suppose $\Pi \in \mathbb{R}^{m \times n}$, where $\Pi_{i[j]} = c_\ell$ for $\ell \in [k]$, and $\Pi_{pq}$ for $pq \in V$, resp. satisfy:

$$0 \leq \Pi_{i[j]} = c_\ell \leq \min(a_{i_\ell}, b_{j_\ell}),$$

$$0 \leq \Pi_{pq} = \frac{(a_p - I_{i[k]}(p)c_p) \cdot (b_q - I_{j[k]}(q)c_q)}{c_\ell} \leq c_k \leq \cdots \leq c_1 \text{ for all } pq \in V.$$  \hspace{1cm} (7) \hspace{1cm} (8)

**Proof.** Since $i[k]$ and $j[k]$ do not repeat row/column indices, then (7) and (8) imply that $\Pi$ satisfies $\Pi_{1,1} = a$ and $\Pi^T 1_m = b$; and since they also imply $\Pi \geq 0$, then $\Pi \in U(a, b)$. Furthermore, we also have $\Pi \in O_{i[k]}$, because (8) and (9) imply for all $pq \in V$, that $\Pi_{pq} \leq a_pb_q/c_\ell \leq c_k$.

**Corollary 2.2.** Suppose $i[k]$ and $j[k]$ do not repeat row (or column) indices. If $a_i = 1/m$ and $b_j = 1/n$, then $\Pi$ is feasible w.r.t (6) if $\min(m, n) \geq (1-k/\max(m,n))^{-1}$.

**Proof.** For all $\ell \in [k]$, pick $c_\ell = \min(1/m, 1/n) = c$ for some $c > 0$; this choice for $c$ satisfies (7) and (8). Then (9) holds if $a_pb_q/c_\ell \leq 1 - kc_\ell = \alpha$ holds, or equivalently $\min(m, n) \geq (1-k/\max(m,n))^{-1}$ holds.

Cor. 2.2 states for uniform constraints $a_i = 1/m$ and $b_j = 1/n$, feasibility typically holds when $k \ll \max(m,n)$, since an upper bound for $(1-k/\max(m,n))^{-1}$ is $1 + O(k/\max(m,n))$. Indeed, such is the case of interest where a sparse $k$ number of OCs are enforced, see (2).

3. Solving OT with Order Constraints

In this section we provide a $\delta$-approximate method for (6) that runs in $O ((|D|_\infty / \delta \cdot m n \log mn)$ time using the alternating direction method of multipliers (ADMM) (Boyd et al., 2011; Beck, 2017). In what follows assume feasibility of (6), and the strategy is to consider the (non-empty) polytope $U(a, b) \cap O_{i[k]}$ as the intersection of two closed convex sets:

$$C_1(a, b) = \{ X \in \mathbb{R}^{m \times n} : X1_n = a, X^T 1_n = b \},$$

and $C_2 = O_{i[k]}$, used to equivalently rewrite (6) as:

$$\min_{X \in C_1(a, b)} f(X) + \ell_{C_2}(Z),$$

s.t. $X, Z \in \mathbb{R}^{m \times n}, X - Z = 0,$

where the indicator function $\ell_{C}(X) = 0$ if $X \in C$, and $\ell_{C}(X) = \infty$ otherwise. For $X \in C_1(a, b), Z \in C_2$, the equality implies $X = Z = \Pi$ for feasible $\Pi \in U(a, b)$. The first-order, iterative ADMM procedure is summarized in Algorithm 1; $\rho > 0$ is a penalty term, and iterates $X_{t+1}, Z_{t+1}$ solve, respectively:

$$\min_{X \in C_1(a, b)} \left( \text{tr} \left( D^T X \right) + \frac{\rho}{2} \| X - Z_t + M_t \|_F^2 \right)$$

$$\min_{Z \in \mathbb{R}^{m \times n}} \left( \ell_{C_2}(Z) + \frac{\rho}{2} \| X_{t+1} - Z + M_t \|_F^2 \right) \hspace{1cm} (12)$$

where $M_t$ is the (scaled) dual iterate, see (Boyd et al., 2011). Simplifying the first min expression in (12) shows that $X_{t+1}$ is nothing but the Euclidean projection onto $C_1(a, b)$ of a point $Z_t - M_t - \rho^{-1}D$; similarly $Z_{t+1}$ is the Euclidean projection onto $C_2$ of $X_{t+1} + M_t$.

**Algorithm 1** Iterative procedure for OT under OC $O_{i[k]}$ with linear costs $f(X) = \text{tr} \left( D^T X \right)$.

**Require:** Costs $D$, penalty $\rho$, initial $M_0, Z_0$.

1: for round $t \geq 1$ until stopping do

2: Update $X_{t+1} = \text{Proj}_{C_1(a,b)} (Z_t - M_t - \rho^{-1}D) \hspace{1cm}$

3: Update $Z_{t+1} = \text{Proj}_{C_2}(X_{t+1} + M_t) \hspace{1cm}$

4: Update (scaled) dual variable $M_{t+1} = M_t + X_{t} - Z_{t} \hspace{1cm}$

5: Return $(scaled)$ dual variable $M_{t+1} = M_t + X_{t} - Z_{t}$

We next derive the projections, $\text{Proj}_{C_1(a,b)} ()$ and $\text{Proj}_{C_2}()$, required for Alg. 1. Prop. 3.1 gives the Euclidean projector for $C_1(a, b)$.

**Proposition 3.1** (Projection $C_1(a, b)$). For $C_1(a, b)$ consider the Euclidean projector $\text{Proj}_{C_1(a,b)}(X)$. Let $P_k$ be the projection of a vector onto its mean, i.e., $P_k = \frac{1}{k} 1_k 1_k^T$. Define matrices $M := I_m - \frac{m}{m+n} P_m$ and $N := I_n - \frac{n}{m+n} P_n$, and matrices

$$Y_1 := \frac{1}{n} \sum_{\ell=1}^{m} [Ma, \cdots, Ma]^2 + \frac{1}{m} \sum_{i=1}^{n} [Nb, \cdots, Nb],$$

$$Y_2 := M(XP_n) + (P_m X)M.$$

Then the projection $\hat{X} = \text{Proj}_{C_1(a,b)}(X)$ satisfies $\hat{X} = Y_1 + (X - Y_2)$. The proof of Prop. 3.1 can be found in the Appendix. Wang et al. (2010) proved a similar property for the simplified setting where $a = b = 1$. Each symmetric, which does not hold in the optimal transport setting.
Algorithm 2 ePAVA for $C_2 = O_{ijkl}$, for $k \leq [mn]$

Require: $X \in \mathbb{R}^{m \times n}$. Indices $i,j,l,k$.
1: $\ell := 1$, $B := 1$, $1 \in [1] := r \in [1] := 1$, $\text{val}[1] := T(0)$.
2: for $\ell \leq k$ do
3: $B := B + 1$, $\ell := \ell + 1$, $1 \in [B] := r \in [B] := \ell$, $\text{val}[B] := X_{\ell_{ij}}$.
4: for $B \geq 2$ and $\text{val}[B] \leq \text{val}[B - 1]$ do
5: Let $q = r \in [B]$.
6: if $B = 2$ then
7: Solve and store $\hat{\eta} \geq 0$ s.t. $T(\hat{\eta}) = \Delta_{2q} + \hat{\eta}/(q - 1)$. Set $\text{val}[B - 1] := T(\hat{\eta})$.
8: else
9: Set $\text{val}[B - 1] := \Delta_{pq}$ for $p = r \in [B - 1]$.
10: Set $r \in [B - 1] := r \in [B]$. Decrement $B := B - 1$.
11: Return $B$, $\hat{\eta}$, le, ri, and val.

Next we address computing $\text{Proj}_{C_2}(X)$ for $C_2 = O_{ijkl}$ for values of $k$ of interest. For the special case where $k = mn$, Grotzinger & Witzgall (1984) defined the so-called Pool Adjacent Violators Algorithm (PAVA) that solves $\text{Proj}_{C_2}(X)$ for $C_2 = O_{ijkl}$.

We cannot directly make use of PAVA for OT with Order Constraints because we require only a small number, $k \ll mn$, of order constraints, and not the full set, since the OT itself should learn the ordering of the other coefficients.

However, the idea behind PAVA is of interest. To this end we extend PAVA, and call this extension ePAVA. ePAVA projects into $C_2 = O_{ijkl}$ for general $k$, and is presented in Alg. 2. ePAVA makes use of two quantities defined below; the proof of correctness is given in the Appendix.

We define a block $B$ to be a partition of indices. Let $1 \in [B]$ denote the left and $r \in [B]$ the right boundary of $B$; $\text{val}[B]$ is the value of its projection. For any $\eta \geq 0$, let $r(\eta)$ denote the rank $X_{\ell_{ij}}$, $\eta$ with respect to decreasing order $X_{(1)} \geq X_{(2)} \geq \cdots \geq X_{(k-1)}$ over all variates in $V$ for $i,j,l,k$.

We require averages $\Delta_{pq} := \sum_{\ell=1}^{p} X_{\ell_{ij}}/(q - p + 1)$ and a threshold function $T(\tau) \geq (\tau(\tau(\eta)), \eta)$, where $\tau(s, \eta)$ for any $0 \leq s < r(\eta)$ and integer $t(\eta) \geq 0$ are defined as:

\begin{align}
\tau(s, \eta) &= \frac{1}{s + 1} \left( X_{\ell_{ij}} - \eta + \sum_{t=1}^{s} X_{\ell_{ij}} \right), \\
t(\eta) + 1 &= \arg \min \{ s \in [r] : \tau(s, \eta) > X_{(s)} \}
\end{align}

or $\tau(\eta) + 1 = r(\eta)$ whenever the set in (15) is empty.

Proposition 3.2 (Projection $C_2$). For $C_2 = O_{ijkl}$, for any $i,j,l,k \in [mn]$ where $\ell \leq [k]$, consider the Euclidean projector $\text{Proj}_{C_2}(X)$ for any $X \in \mathbb{R}^{m \times n}$. Let $T(\eta) := (\tau(\tau(\eta), \eta))_{\ell_{ij}}$ for $\tau, t$ in (14) and (15) and $V$ in (2). Then

for any $X \in \mathbb{R}^{m \times n}$, ePAVA will successfully terminate with some $B, \hat{\eta}, le, ri, and val$. Furthermore, the projection $X = \text{Proj}_{C_2}(X)$ satisfies i) for $pq \in V$ we have $X_{pq} = T(\hat{\eta}) = \text{val}[1]$ if rank $(X_{pq}) \leq t(\hat{\eta})$ or $X_{pq} = (X_{pq})_{\ell_{ij}}$ otherwise, and ii) for $\ell \in [k]$ we have $X_{\ell_{ij}} = \text{val}[B']$ if $\ell \in [B'] \leq \ell \in B'$ for some $B' \in [B]$.

Convergence Analysis Standard ADMM convergence holds as follows. As $t \to \infty$, the primal residue $\|Z_t - Z_0\|_F \to 0$ and dual residue $\rho \|Z_t - Z_0\|_F$ which can be used to show asymptotic convergence of $f(X_t)$ to the optimum (see Boyd, 2010, p. 17). The iteration complexity of Alg. 1 can be obtained from the following $d$-dimensional asymptotic result (see Beck, 2017, Thm 15.4). Consider the following.

As (10)-(11) are both linear, if $C_1(a,b) \cap C_2 = \emptyset$, then an optimal $Y^* = \arg \max_{Y \in \mathbb{R}^{mn \times n}} d(Y)$ exists (Boyd & Vandenberghe, 2004) from solving the dual $d(Y) :=$

\begin{equation}
\min_{X \in C_1(a,b), Z \in C_2} \text{tr}(X^T(D + Y)) - \text{tr}(Y^T Z)
\end{equation}

Recall Proposition 2.1 is a sufficient condition for $C_1(a,b) \cap C_2 = \emptyset$.

Theorem 3.3 (Convergence of Algorithm 1, Beck (2017)). Let $X_t, Z_t, M_t$ be the sequences of iterates from Algorithm 1 with penalty $\rho > 0$, and let $X_t = (t + 1)^{-1}(1 - 1)^{t+1} X_t$ and similarly $Z_t$. Assume $C_1(a,b) \cap C_2 = \emptyset$, and let $f^*$ and $\Pi^*$ denote the optimum value and solution respectively of (6), and $Y^* = \arg \max_{Y \in \mathbb{R}^{mn \times n}} d(Y)$ for $d(Y)$ in (16). Then we have the convergence bounds

\begin{equation}
f(X_t) - f^* \leq \frac{\rho c_1}{2(t + 1)}, \quad \|X_t - Z_t\|_F \leq \frac{\rho c_1}{c_2(t + 1)}
\end{equation}

where $c_1 \geq \|Z_0 - \Pi^*\|_F^2 + (\|M_0\|_F + c_2/\rho)^2$ and $c_2 \geq 2\|Y^*\|_F$.

Proposition 3.4 (Iteration complexity). Algorithm 1, given linear costs $D$ and penalty parameter $\rho \geq 0$, initialized with $Z_0 = M_0 = 0$, achieves error $f(X_t) - f^* \leq \delta$, in $O(\|D\|/\delta)$ iterations.

Proof. We seek to estimate $\rho c_1$ that bounds the error $f(X_t) - f^*$ in Thm. 3.3. First, consider the constant $c_1$. Setting $Z_0 = M_0 = 0$ gives $\|Z_0 - \Pi^*\|_F^2 \leq 1$ and $\|M_0\|_F = 0$, then Thm. 3.3 requires $c_1 \geq 1 + (c_2/\rho)^2$; hence if we set the penalty $\rho = c_2$ then $c_1 \geq 2$. Next, it remains to estimate $\rho$, or equivalently, $c_2$. Thm. 3.3 requires $c_2$ to be at least the largest possible norm that an optimizer of the dual (16) can take. To this end if, writing $Y^*(D')$ for the optimizer of (16) given costs $D'$, it suffices to consider

$3$The sum in (14) equals 0 when $s = 0$. $4$Use Thm 15.4 of Beck (2017) with G, Q set to zero, and checking Assump 15.2 holds for $f(X)$ and $C_1(a,b), C_2$. Order Constraints in Optimal Transport
a constant $c$ satisfying $c \geq \|Y^*(D\')\|_\rho$ for any normalized
\footnote{The dual $d$ in (16) is scale-invariant, in the sense that if $Y^*$
optimizes (16) for costs $D$, then for any $\alpha > 0$, we have $\alpha Y^*$
optimizes the same for costs $\alpha D$.} cost $D\'$ with $\|D\|_\infty = 1$, and then picking $c_2$ such
that $c_2 \geq 2 \|D\|_\infty c$. Hence we arrive at an estimate for $\rho c_1$ to
be in $O(\|D\|_\infty)$. Thus we conclude the error is at most $\delta$
under the claimed number of iterations.

Prop. 3.4 holds for the choice$^6 \rho = c_2$, but however $c_2$ is
unknown in practice; the common adage is to simply set $\rho = 1$, (see Boyd et al., 2011). Invoking both Propositions 3.4 and 3.5 below, we estimate the total operation complexity as $O(\|D\|_\infty / \delta \cdot mn \log mn)$.

**Proposition 3.5.** One round of Algorithm 1 runs in $O(mn \log (mn))$ time.

\textbf{Proof.} Line 2 costs $O(mn)$ for $C_1(a, b)$ and Line 4 costs
$O(k + (mn - k) \log (mn - k))$ for $C_2$. The update in
Line 2 requires only $O(mn)$. Note that for Proj$_C(a, b)$ the
expressions (13) require only matrix-vector multiplications
with $O(mn)$ complexity, as follows. For $Y_1$ this is clear from (13).
For $Y_2$, consider the left term $M(XP_n)$; then the other
term $(P_m X) M$ similarly follows. Putting
$M = I_m - \frac{m}{m+n} P_m$ and $P_k = \frac{1}{k}(1 \cdot k^T)$
for $k = m, n$, we get that
$M(XP_n)$ can be expanded (ignoring scaling factors)
to two terms $X1_m 1^T_m$ and $1^T_m (I_m X_1_n \cdot 1^T_m)$. Hence
the claim holds given that $1_m, 1_n$ are vectors.

For $C_2$, observe that computing $T(\eta) = \tau(t(\eta), \eta)$
multiples times in Alg. 2 (Line 7), requires a one-time sort of
mn - k + 1 terms and Lines 5-10 incur an additional $O(k)$
following arguments in Grotzinger & Witzgall (1984). \hfill \Box

Care should to be taken in comparing our complexity,
$O(\|D\|_\infty / \delta \cdot mn \log mn)$, to that of OT methods that use rounding approximations Altschuler et al. (2017) to ensure feasibility of each iteration. Such methods, which use KL-divergence, are $O(n^2 / \delta \cdot \|D\|_\infty)$ for $m = n$, (see Guminov et al., 2021; Jambulapati et al., 2019). For order constraints like other structured OT approaches (e.g. Scketon et al., 2021), a rounding method for $U(a, b) \cap O_{i\neq j}$ is not available. However, we have the bound on $\|\tilde{X}_i - \tilde{Z}_i\|_F$ in Thm. 3.3 to quantify the feasibility of the iterates $X_i, Z_i$
from $C_1(a, b) \cap C_2$.

\section{4. Explainability via Branch-and-Bound}

We have thus far assumed that the structure captured by the
optimal transport plan via order constraints has been provided externally. In some settings, the structure is not provided and one must estimate the most important variates to
\footnote{For the choice $\rho = c_2$, it follows that the error $\|X_t - Z_t\|_F$ in Thm. 3.3 also drops linearly with $t$.}
define the corresponding constraints. With the goal of generat- ing plans from which one can select, we propose an explainable and efficient approach using branch & bound to compute a diverse set of optimal transport plans.

The approach successively computes a bound on the best
score that can be obtained with the variates currently fixed;
if it cannot improve the best known score, the branch is cut
and the next branch, i.e. the next set of fixed variates, is
explored. Consider the $k$ order constraints (2) enforced
by the variate $ij|k|$, and further consider introducing an
additional variate $ij|k|$ corresponding to single order constraint variates $ij|k|$. We would like the procedure to always select all ancestors of a given node before selecting the node itself. Hence, the root node should be selected first.

The branch selection aimed at increasing diversity of the
plans is guided by two threshold parameters $\tau_1, \tau_2$ that lie
between 0 and 1. For any given node $ij|k|$, the parameters
$\tau_1, \tau_2$ limit its children to those that only deliver reasonably
likely transport plans, as follows.

For example, suppose that $\Pi \in U(a, b)$ is the solution of (6)
obtained from the variate $ij|k|$. To determine the children of
$ij|k|$, first compute saturation levels normalized between 0
and 1 given as $\phi_{ij} = \Pi_{ij} / \min(\alpha_i, b_j)$, to derive

\begin{equation}
\left(\phi^a_{ij}, \phi^r_{ij}, \phi^e_{ij}\right) = \left(\phi_{ij}, \max_{\ell \in [n]; \ell \neq i} \phi_{\ell j}, \max_{\ell \in [m]; \ell \neq j} \phi_{ij}\right)
\end{equation}

\begin{equation}
\Phi_{ij} := \min(\phi^a_{ij}, \phi^e_{ij}).
\end{equation}

Low saturation values in (17) imply uncertainty in the as-
signed nodes II. The thresholds $\tau_1, \tau_2$ are used to determine the
set $\mathcal{I}(\Pi)$ that filters those variates $ij$ with low self sat-
uration ($\tau_1$) and low neighborhood saturation ($\tau_2$):

\begin{equation}
\mathcal{I}(\Pi) := \{ij \in [mn] : \phi^a_{ij} \leq \tau_1, \Phi_{ij} \leq \tau_2\}.
\end{equation}

The children of $ij|k|$ are obtained as $i'j'|k+1 = ij, ij|k|$ for
all $ij \in \mathcal{I}(\Pi)$.

The proposed explainable approach adds diversity by learn-
ing the top plans. Let us define $k_1, k_2, k_3$ as follows:
The number of nodes constructed while learning a subtree is
at most $k_1$, the number of top plans to retain is $k_2$, and $k_3$, as
noted above, limits the tree depth. The branch-and-bound
search proceeds on the reduced tree given by $\mathcal{T}(k_3, \tau_1, \tau_2)$,
of depth $k_3$, containing nodes (18) whose saturation lie
below the thresholds $\tau_1, \tau_2$. We thus wish to learn the top-
most plans given by subtree, $\mathcal{T}(k_1, k_2, k_3, \tau_1, \tau_2)$, and iden-
Algorithm 3 Learning subtree $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ of $T(k_3, \tau_1, \tau_2)$ and top-$k_2$ candidate plans for linear costs $f(\Pi) = \text{tr}(D^T\Pi)$.

Require: Costs $D$, thresholds $0 \leq \tau_1, \tau_2 \leq 1$. Search upper limit $k_1$, number of top candidates $k_2$, and search depth $k_3 \leq \min(m, n)$.

1: Compute $\hat{\Pi}$ using (1). Init $f(\hat{\Pi})$ using (2).
2: Use $\hat{\Pi}$ and $\tau_1, \tau_2$ in (17) and (18) to obtain $T$ and $\Phi_{ij}$.
3: Set $\text{count} = 0$.
4: for $\text{count} < k_2$ do
5:   Pop $ij[k]$ having smallest $\Phi$ in $\mathcal{S}$, for some $k$ constraints. Compute $\mathcal{L}$ from right-hand side of (23).
6:   if $\hat{\Pi}$ is not yet obtained or $\mathcal{L} > f(\hat{\Pi})_k$ then
7:      Solve Algorithm 1 with order constraint $O_{ij[k]}$ for new candidate $\hat{\Pi}$. Set $\text{count} = 1$.
8:   Update top-$k_2$ candidates $\hat{\Pi}_1, \hat{\Pi}_2, \ldots, \hat{\Pi}_{k_2}$ and $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ using new candidate $\hat{\Pi}$.
9: if $k$ equals $k_3$ then
10:   Go to line 4.
11: if $\hat{\Pi}_{k_2}$ not yet obtained or $f(\hat{\Pi}) < f(\hat{\Pi}_{k_2})$ then
12:     Use $\hat{\Pi}$, $\tau_1, \tau_2$ in (17) and (18) and obtain new variates $ij \in T(\hat{\Pi})$ and $\{\Phi_{ij}\}_{ij \in T(\hat{\Pi})}$.
13:     for variate $ij \in T(\hat{\Pi})$ do
14:        if $i \notin i[k]$ and $j \notin j[k]$ then
15:           Push $(ij, i, j_1, \ldots, i_k, j_k, \Phi_{ij})$ onto stack $\mathcal{S}$.
16:           Return top $k_2$ candidates $\hat{\Pi}_1, \hat{\Pi}_2, \ldots, \hat{\Pi}_{k_2}$ and $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$.

Alg. 3 summarizes the branch-and-bound variable selection procedure. Upon termination, a diverse set of at most $k_1$ plans are computed and the $k_2$ plans identified by $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ are the top-$k_2$ amongst these $k_1$ plans, as the most uncertain variables are handled first.

It remains to define the lower bound $\mathcal{L} \leq f(\Pi)$ (Line 5); Kusner et al. (2015) proposed a bound for (1), which we extend to the order constrained case (6) as follows:

$$\min_{\Pi \in U(a,b) \cap O_{ij[k]}} f(\Pi) \geq \min_{\alpha \leq x \leq \beta} \left[ G_{ij[k]}(x) + g_{ij[k]}(x, D) \right]$$

for some suitable $\alpha, \beta$, where $V$ in (2), and where $G_{ij[k]} = \sum_{x \in [k]} \nu(x, \{\phi_{ij[k]}\}_{ij \in \{k\}}, \alpha, \gamma) + \sum_{\beta \in [k]} \mu(x, \{\phi_{ij[k]}\}_{ij \in \{k\}}, \beta, \gamma)$

Figure 3: Top $k_2 = 5$ candidates for OT with multiple order constraints for the example in Fig. 2.

$$\sum_{i \in [k]} D_{ij[k]} \text{ and for coefficients } \varphi \in \mathbb{R}^{m \times n},$$

$$g_{ij[k]}(x, \varphi) := \min_{\Pi \in U(a,b) \cap O_{ij[k]}} \Pi_{ij[k]} = x, \forall \in [k] \sum \Pi_{pq} \varphi_{pq}$$

We decouple the row and column constraints in (20) and solve PACKING $\left( \{\varphi_i\}_{i \in [n]}, u, \alpha \right)$:

$$\min_{x \in \mathbb{R}^n} \sum_{i \in [n]} \varphi_i x_i \text{ s.t. } \sum_{i \in [n]} x_i = \alpha, 0 \leq x_i \leq u. \quad (20)$$

Here $u$ and $\alpha$ in (20) represent the per item $x_i$ capacity and total budget, respectively. Define

$$\mu(u, \{\varphi_i\}_{i \in [n]}, \alpha) = \text{PACKING} \left( \{\varphi_i\}_{i \in [n]}, u, \alpha \right)$$

$$\nu(u, \{\varphi_i\}_{i \in [n-1]}, \alpha) = \text{PACKING} \left( \{\varphi_i\}_{i \in [n-1]}, u, \alpha - u \right)$$

**Proposition 4.1.** Let $\alpha_1 = \max_{i=1}^m a_i$, $\beta_1 = \max_{i=1}^m a_i$, and $\alpha_2 = \max_{j=1}^n b_j$, $\beta_2 = \max_{j=1}^n b_j$. Then for any $ij[k]$ set enforcing (6) where $i[k] = i_1, i_2, \ldots, i_k$ and $j[k]$ do not repeat row or column indices, the quantity $g_{ij[k]}(x, \varphi)$ appearing in (19) is lower-bounded by

$$g_{ij[k]}(x, \varphi) \geq \left\{ \begin{array}{ll} L_{1,ijk}(x, \varphi, a), & \alpha_1 \leq x \leq \beta_1 \ \\
L_{2,ijk}(x, \varphi, b), & \alpha_2 \leq x \leq \beta_2 \end{array} \right.$$ 

where $L_{1,ijk}(x, \varphi, a)$ and $L_{2,ijk}(x, \varphi, b)$ equal, resp.

$$\sum_{t \in [k]} \nu(x, \{\phi_{tq}\}_{t \in [k]}, \alpha, \gamma) + \sum_{\beta \in [k]} \mu(x, \{\phi_{tq}\}_{t \in [k]}, \beta, \gamma)$$
Table 1: Task and annotation scores on e-SNLI. BestF1@n is the best achievable annotation F1 score over the set of $k_2 = n$ best plans, measured by setting coefficients 0 or 1 after determining values under which there is no impact on the task F1 (75.1 ± .3). Standard OT achieves 64.5 ± .3 annotation F1.

| Algorithm            | BestF1@n   |
|----------------------|------------|
|                      | n = 2      | n = 5      | n = 10     |
| Algorithm 3 (ours)   | 68.1 ± .2  | 71.2 ± .3  | 73.7 ± .2  |
| Greedy version       | 67.9 ± .3  | 68.2 ± .3  | 68.2 ± .3  |

The proof is in the Appendix. Set $\varphi = D$ in (22). Now, from (19) and Prop. 4.1 we obtain:

$$\sum_{i \in [k]} \nu \left( x, \{ \varphi_{pj} \}_{p \in [m] \setminus \{i_1 \}}, b_{ji} \right) + \sum_{q \notin [k]} \mu \left( x, \{ \varphi_{pq} \}_{p \in [m]}, b_q \right)$$

The proof is in the Appendix. Set $\varphi = D$ in (22). Now, from (19) and Prop. 4.1 we obtain:

$$\min_{\Pi \in \mathcal{U}(a,b)\cap \Omega(i|k)} f(\Pi) \geq \max \left( \min_{\alpha_1 \leq x \leq \beta_1} G_{ijk} : x + L_{1ijk} (x, D, a) \right)$$

Thus (23) provides the lower bound in Line 4 of Alg. 3.

5. Experimental Results

Sentence Relationship Classification: We use an annotated dataset from the enhanced Stanford Natural Language Inference (e-SNLI) (Camburu et al., 2018; Swanson et al., 2020) that includes English sentence pairs classified as “entailment”, “contradiction” or “neutral”. Annotations denote which words were used by humans to determine the class.

We use Alg. 3 to compute $k_2$ candidate plans which are measured against the annotations after being set to a binary variable; values above a threshold are set to 1, and below to 0. The threshold is determined as the one that maintains the reported task F1 score when the plan weights that lie below the threshold are set to zero. We determine the $(\tau_1, \tau_2)$ that constrain $\mathcal{T}(k_3, \tau_1, \tau_2)$ to be $(\tau_1, \tau_2) = (0.5, 0.5)$. Annotations are provided separately on the source and target in each pair; each candidate plan is marginalized using $\max$ across columns/rows to arrive at a vector of coefficients, and used to compute an annotation F1 score against the annotations (after application of the threshold). The top $k_2 = n$ plans are scored using the BestF1@n metric, which reports the score of the best plan in the learnt subtree $\mathcal{T}(k_1, k_2, k_3, \tau_1, \tau_2)$. More details can be found in the Appendix.

Table 1 shows the results in terms of the classification task and annotation accuracy along with confidence intervals. Using even only a single order constraint, i.e. $k_3 = 1$, the tree search in Alg. 3 achieves significantly improved explainability using a modest $k_1 = 20$ candidates, giving higher annotation scores over the set of plans. As a baseline against which to compare, we implement a greedy algorithm by restricting the set $\mathcal{I}(\Pi)$ of variates in (18) to contain a single variate, i.e., modifying Lines 2 and 11 of Alg. 3. The greedy algorithm retains the variate $ij \in \mathcal{I}(\Pi)$ with the lowest saturation (17), thus corresponding to a single tree path. For $n \geq 5$, the greedy algorithm results in a candidate set lacking diversity, as evidenced by the stagnating BestF1@n scores as compared to Alg. 3.

Fig. 3 shows the search tree $\mathcal{T}(k_3, \tau_1, \tau_2)$ (blue), the learnt
subtree $\mathcal{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ (red) and the corresponding candidate plans for the example of Fig. 2 for the top $k_2 = 5$ candidates and depth increased to $k_3 = 3$. See Supp. Mat. for a walk-through of Alg. 3 using Fig. 3 and a study of the effectiveness of the lower bound using a “swarmplot” of # of nodes skipped in the search tree, $\mathcal{T}(k_3, \tau_1, \tau_2)$ as tree depth $k_3$ increases. The numbers in Fig. 3 indicate the ranking in the top-$k_2$ order upon termination. The diversity provided by Alg. 3 allows the 3rd best solution (in red) to be included in the set of plans, and it is that plan that best explains the contradiction in the two sentences.

**Color Transfer** We use source images from the SUN dataset (Xiao et al., 2010; Yu et al., 2016), and target images from WikiArt (Tan et al., 2016). The images are segmented and the average RGB colors in each segment are used to obtain distributions $a, b$ and costs $D$ in the OT formulations (1) and (3). The thresholds that constrain $\mathcal{T}(k_3, \tau_1, \tau_2)$ are set to $(\tau_1, \tau_2) = (0.5, 1)$ here. Further details on the problem setup can be found in the Appendix. Results are shown in Fig. 4. The first OT with OC plan includes a prior (human-crafted) constraint while the latter two provide the auto-generated constraints from Alg. 3. The OT with OC solutions offer a range of images, all of which improve upon the standard OT solution in diverse ways. In the first row, standard OT transfers the blue sky to the ground and the cylindrical roof, while the prior constraint maps the green grass from the painting to the ground and intensifies the sky with the blue. The auto-generated constraints from Alg. 3, especially the $k_3 = 2$ case, do the same though slightly less. In the second row, standard OT awkwardly transferred the red to the grass path. The OT with OC solution with a prior constraint as well as the auto-generated constraints of Alg. 3, $k_3 = 2$ use the red to create an evening sky effect. In the third row, standard OT results in a very dark cliff with minimal visible features; the OT with OC solutions lead to a well-defined cliff and more effectively use the blue from the painting sky to deepen the sky in the source image. In addition to producing better resulting images, Alg. 3 allows for human judgement to be used to make the final selection from a set of plans.

**Computational Efficiency** Fig. 5 shows how the computation time (in secs) of Alg. 1 scales with problem size. We generate 100 random problems for $m, n \leq 100$ with 1, 2, 4 and 10 order constraints. The iterations are set to terminate at 1e4 rounds or a max projection error of 1e-4, and these settings achieve an average functional approximation of 0.51% error (within ±.19). We use penalty $\rho = 1.0$ after testing a range of $\rho$’s and observing little difference, as it is well-known in the ADMM literature (Boyd, 2010). We compare our method against scipy.optimize.

Fig. 5 shows 95% confidence intervals from 10 independent repetitions. Alg. 1 scales much better than scipy.optimize for large problems by orders of magnitude as the problem size $mn$ increases. Note that Fig. 5 compares python-based algorithms; we also evaluated the C++-based cvxpy, and found that our algorithm performs almost indistinguishably to cvxpy for a single order constraint, which is remarkable given the overhead of python as compared to C++.

6. Conclusions

Our proposed optimal transport with order constraints allows complex structure to be incorporated in optimal transport plans and provides a set of explainable solutions from which a human can select. Future extensions of interest are as follows. Other OT solvers may be extended to incorporate order constraints. Such solvers may offer different sets of plans satisfying varying objectives arising from other loss functions and various forms of regularization. See (Flamary et al., 2021). Nonlinear costs such as the submodular formulation of (Alvarez-Melis et al., 2018) can be useful in some settings and may be extended to include order constraints. It would also be of interest to explore other applications that can benefit from OT and specifically OT with OC. Optimal Transport with Order Constraints can be found in the AI Explainability 360 toolbox, which is part of the IBM Research Trusted AI library (Arya et al., 2019) at https://github.com/Trusted-AI/AIX360.
References

Altschuler, J., Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, pp. 1961–1971, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. Massively scalable sinkhorn distances via the nyström method. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

Altschuler, J. M. and Boix-Adsera, E. Hardness results for Multimarginal Optimal Transport problems. arXiv:2012.05398 [cs, math], December 2020a. arXiv: 2012.05398.

Altschuler, J. M. and Boix-Adsera, E. Polynomial-time algorithms for multimarginal optimal transport problems with structure. 2020b. URL http://arxiv.org/abs/2008.03006.

Alvarez-Melis, D., Jaakkola, T., and Jegelka, S. Structured optimal transport. In Storkey, A. and Perez-Cruz, F. (eds.), Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of Proceedings of Machine Learning Research, pp. 1771–1780. PMLR, 09–11 Apr 2018. URL http://proceedings.mlr.press/v84/alvarez-melis18a.html.

Alvarez-Melis, D., Jegelka, S., and Jaakkola, T. S. Towards optimal transport with global invariances. In Chaudhuri, K. and Sugiyama, M. (eds.), Proceedings of Machine Learning Research, volume 89 of Proceedings of Machine Learning Research, pp. 1870–1879. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/alvarez-melis19a.html.

Alvarez-Melis, D., Mroueh, Y., and Jaakkola, T. Unsupervised hierarchy matching with optimal transport over hyperbolic spaces. In International Conference on Artificial Intelligence and Statistics, pp. 1606–1617. PMLR, 2020.

Arya, V., Bellamy, R. K. E., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., Mourad, S., Pedemonte, P., Raghavendra, R., Richards, J., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K. R., Wei, D., and Zhang, Y. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, 2019. URL https://arxiv.org/abs/1909.03012.

Beck, A. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, October 2017. ISBN 978-1-61197-498-0 978-1-61197-499-7. doi: 10.1137/1.9781611974997. URL http://epubs.siam.org/doi/book/10.1137/1.9781611974997.

Blondel, M., Seguy, V., and Rolet, A. Smooth and sparse optimal transport. In Storkey, A. and Perez-Cruz, F. (eds.), Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of Proceedings of Machine Learning Research, pp. 880–889. PMLR, 09–11 Apr 2018. URL http://proceedings.mlr.press/v84/blondel18a.html.

Boyd, S. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends® in Machine Learning, 3(1):1–122, 2010. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000016. URL http://www.nowpublishers.com/article/Details/MAL-016.

Boyd, S. P. and Vandenberghe, L. Convex optimization. Cambridge University Press, Cambridge, UK ; New York, 2004. ISBN 978-0-521-83378-3.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. e-SNLI: Natural language inference with natural language explanations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(9):1853–1865, 2017. doi: 10.1109/TPAMI.2016.2615921.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), Advances in Neural Information Processing Systems 26, pp. 2292–2300. Curran Associates, Inc., 2013.

De Leeuw, J., Kurt, H., and Mair, P. Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA)
and Active Set Methods. *Journal of Statistical Software*, 32, October 2009. doi: 10.18637/jss.v032.i05.

Di Marino, S., Gerolin, A., and Nenna, L. Optimal transportation theory with repulsive costs. 2015. URL http://arxiv.org/abs/1506.04565.

Felzenszwalb, P. F. and Huttenlocher, D. P. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004. ISSN 0920–5691. doi: 10.1023/B:VISI.0000022288.19776.77. URL http://link.springer.com/10.1023/B:VISI.0000022288.19776.77.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Faturas, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–20, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Forrow, A., Hütter, J.-C., Nitzan, M., Rigollet, P., Schiebinger, G., and Weed, J. Statistical optimal transport via factored couplings. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2454–2465. PMLR, 2019.

Goldfeld, Z. and Greenewald, K. Gaussian-smoothed optimal transport: Metric structure and statistical efficiency. In *International Conference on Artificial Intelligence and Statistics*, pp. 3327–3337. PMLR, 2020.

Grotzinger, S. J. and Witzgall, C. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12(1):247–270, October 1984. ISSN 1432–0606. doi: 10.1007/BF01449044.

Guminov, S., Dvurechensky, P., Tupitsa, N., and Gasnikov, A. On a Combination of Alternating Minimization and Nesterov’s Momentum. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 3886–3898. PMLR, July 2021. URL https://proceedings.mlr.press/v139/guminov21a.html. ISSN: 2640-3498.

Guo, W., Ho, N., and Jordan, M. Fast algorithms for computational optimal transport and wasserstein barycenter. In *International Conference on Artificial Intelligence and Statistics*, pp. 2088–2097. PMLR, 2020.

Jambulapati, A., Sidford, A., and Tian, K. A Direct o(1/epsilon) Iteration Parallel Algorithm for Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. From word embeddings to document distances. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 957–966, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/kusnerb15.html.

Laclau, C., Redko, I., Choudhary, M., and Largeron, C. All of the fairness for edge prediction with optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pp. 1774–1782. PMLR, 2021.

Lin, T., Ho, N., and Jordan, M. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning*, pp. 3982–3991. PMLR, 2019.

Liu, B., Rao, Y., Lu, J., Zhou, J., and Hsieh, C.-J. Multi-proxy wasserstein classifier for image classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):8618–8626, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17045.

Pass, B. Multi-marginal optimal transport: Theory and applications. *ESAIM: M2AN*, 49(6):1771–1790, 2015. doi: 10.1051/m2an/2015020. URL https://doi.org/10.1051/m2an/2015020.

Paty, F.-P., d’Aspremont, A., and Cuturi, M. Regularity as regularization: Smooth and strongly convex breniér potentials in optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pp. 1222–1232. PMLR, 2020.

Peyré, G. and Cuturi, M. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. ISSN 1935-8237. doi: 10.1561/2200000073. URL http://dx.doi.org/10.1561/2200000073.

Scetbon, M., Cuturi, M., and Peyré, G. Low-Rank Sinkhorn Factorization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9344–9354. PMLR, July 2021. URL https://proceedings.mlr.press/v139/scetbon21a.html. ISSN: 2640-3498.

Schmitzer, B. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019. doi: 10.1137/16M1106018.

Shi, Y., Yu, X., Liu, L., Zhang, T., and Li, H. Optimal feature transport for cross-view image geo-localization.
Appendix

The Appendix comprises two sections. Section A includes further details on the Sentence Relationship Classification and Color Transfer experiments and on the Computational Efficiency of our proposed OT with OC method. Section B of the Appendix includes the proofs for the results in the main paper.

A. Details on Experimental Setup and Results

A.1. Sentence Relationship Classification

For the enhanced Stanford Natural Language Inference (e-SNLI) dataset we follow Swanson et al. (2020) to evaluate explainability of an OT scheme, whereby annotation-based scores are measured after applying a threshold to determine the activations. The threshold is taken as the one that balances task and explainability performance; it should be a sufficiently high so that we do not get spurious activations, and sufficiently low so that enough coefficients are preserved to not compromise the task scores. e-SNLI provides annotation labels to determine the explainability performance, quantified by the annotation $F_1$. The highest annotation $F_1$ score over $n = k_2$ candidates of the Algorithm 3 determines the BestF1@n scores. The Greedy version used as a baseline is scored the same way. We used sizes of (100K, 10K, 5K) for train, validation, and test, respectively.

Picking the threshold for the classification task: This is a three class “entailment”, “neutral” and “contradiction” classification task to predict logical relationships between sentence pairs. The task is performed by a shallow neural network that incorporates an OT attention module between the input and output layers\(^7\). The attention layer uses OT to match source and target tokens, and matched tokens are concatenated and output via softmax. The shallow network is trained on the un-thresholded plan. Over the validation and tests sets, all coefficients that lie below $T/(mn)$ are dropped when measuring the task $F_1$. Using the validation set, we pick the threshold in the region between 0.01 and 2 where the task $F_1$ starts to plateau due to the dropped coefficients; we use an activation threshold of 2.

Comparing activations with annotation labels: In e-SNLI, annotations that are provided separately on the source and target sentence. Swanson et al. (2020) proposed to marginalize the transport plan $\Pi \in U(a,b)$ forming two vectors for comparison, by applying max to each row and column; we found that this method overlooks the coupling in the transport plan $\Pi$, and we propose instead taking max only over row/column positions that have annotations labels. On the other hand we cannot then utilize “neutral” examples for measuring annotation $F_1$ (since these examples are missing row annotations), but we do not consider this as a drawback of our proposal because Camburu et al. (2018) indicated that the annotations for the “neutral” examples are curated in a somewhat inconsistent manner from the other example classes, and we thus omit them from the annotation scores.

Hyperparameters: Alg. 3 takes a standard OT baseline $\Pi \in U(a,b)$ to generate the variate set $I(\Pi)$, see (18). We use a regularized OT (the hyper-parameter-less mirror descent version of standard OT (see Scetbon et al., 2021) with 20 iterations) to obtain $\Pi$. The standard OT solution lies on a polytope and has at most $m + n$ non-zeros out of the $mn$ locations (Peyré & Cuturi, 2019). The regularized version produces more non-zeros that give more information for seeking unsaturated locations in (18). For the standard OT attention network we do the same with a lower 5 iterations to speed up training.

We obtain $(\tau_1, \tau_2)$ that constrain $\mathcal{T}(k_3, \tau_1, \tau_2)$, as follows. First compute transport plans $\Pi \in U(a,b)$ by solving (1), using the regularized version see above, and computing $\phi_{ij}^*$ and $\Phi_{ij}$ using (18). The polytope $U(a,b)$ constrains the points $(\phi_{ij}^*, \Phi_{ij})$ to lie in a lower triangular region bounded by the horizontal and vertical axes and a line that runs through $(1,0)$ and $(0,1)$. The plan coefficients that are uncertain will lie in a box region bounded by the axes $\tau_1$ and $\tau_2$. Using SNLI annotations as labels that indicate importance, we chose $(\tau_1, \tau_2) = (.5, .5)$. We only consider variates $ij$ in (18) if both $i, j$ do not correspond to stop-words. The transport plan is marginalized using max across columns/rows to arrive at a vector of coefficients, and we compute an annotation $F_1$ score against the annotations. The top $k_2 = n$ plans are scored using the BestF1@n metric, which reports the score of the best plan in the learnt subtree $\mathcal{T}(k_1, k_2, k_3, \tau_1, \tau_2)$.

Walkthrough of Alg. 3 using Fig. 3: At Line 1, the root node (labeled R) and its candidate plan $\hat{\Pi}_1$ are computed, and $\hat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ is initialized. At Line 2 the single order constraint nodes are constructed and pushed to stack $S$ along

\(^7\)https://github.com/asappresearch/rationale-alignment
with saturations $\Phi_{ij}$ using (17). At Line 4 a variate $ij$ corresponding to a single order constraint (e.g., “piece”-“pair”) is popped off, and at Line 6 a new candidate $\hat{\Pi}_2$ is computed. At Line 7 this candidate is identified by adding $ij$ to $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ which now has two nodes. Then at Lines 11-14, new depth-2 nodes $i'j'ij$ are computed from $\hat{\Pi}_2$ and pushed to $S$ to be considered next in $T(k_3, \tau_1, \tau_2)$ (e.g., “piece”-“pair” followed by “clothing”-“pants”). Lines 3-14 iterate to update $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ and the candidate set. Once $\hat{T}(k_1, k_2, k_3, \tau_1, \tau_2)$ grows to $k_2$ nodes, the lower bound test at Lines 5 and 11 prune redundant nodes of $T(k_3, \tau_1, \tau_2)$. The iterations terminate when count reaches $k_1$.

**Hardware and compute times:** Classifier training was performed on a multi-core Ubuntu virtual machine and on a nVidia Tesla P100-PCIE-16GB GPUs. For the shallow neural nets GPU memory consumption was found $<$ 1GB. Training times for 10 epochs (around 5 million sample runs) were roughly 21 hours. Algorithm 3 took about 16 hours to complete over the test set.

### A.2. Color Transfer

For the source image, we use Felzenszwalb & Huttenlocher (2004) to preserve pixel locality (see Alvarez-Melis et al., 2018), and for the target we perform color RGB segmentation via $k$-means, in the range of 5-10 color clusters. We illustrate the results of Alg. 3 on three examples from learnt subtrees $\hat{T}(20, 5, 1, \tau_1, \tau_2)$ and $\hat{T}(40, 10, 2, \tau_1, \tau_2)$, where in this set of experiments we chose $(\tau_1, \tau_2) = (0.5, 1.0)$. Here we have no labels to tune $(\tau_1, \tau_2)$ as we did for e-SNLI. The neighbour saturation metric $\Phi_{ij}$ (see (17)) had less effect in this application. Therefore, it sufficed here to use non-regularized OT to compute the base-plan $\Pi \in U(a, b)$. We use the 5 largest source image segments and the 2 largest target image segments in terms of $D_{ij}$; that is, we only consider source image segments large enough to be visually significant, and target image segments with contrasting colors. The WikiArt dataset can be found at https://github.com/cs-chan/ArtGAN/tree/master/WikiArt%20Dataset.

### A.3. Computational Efficiencies

**Computation Issues** Each computation run of Alg. 1 is measured on a single Intel x86 64 bit Xeon 2MHZ with 12GB memory per core. Also, since the python library scipy.optimize depends on numpy⁸ and can run numpy with multiple threads, for fair comparison with our (single-threaded) implementation of Alg. 1, we disabled all parallelisms.

**Effectiveness of Lower Bound (23) in Alg. 3** Figure 6 demonstrates the effectiveness of the lower bound (23), by showing the number of search tree nodes in $T(k_3, \tau_1, \tau_2)$ that were skipped while performing the e-SNLI experiment for various depths $k_3 = 1, 2, 3$ and a large candidate size $k_1 = 40$. The bounds show to be effective in skipping more nodes as the size of $T(k_3, \tau_1, \tau_2)$ increases with the depth $k_3$.

⁸https://numpy.org/
B. Proofs

First we provide the proofs of the propositions concerning the projections, Proposition 3.2 and Proposition 3.1. Secondly, we cover the bounds, giving the proof of correctness of Proposition 4.1.

B.1. Projections

The projection $\text{Proj}_{C_1 \cap C_2}(X) = \arg \min_{Y \in C_2} \|Y - X\|_F^2$ involves $mn - k$ constraints $Y_{pq} \leq Y_{ij,j_1}$ for $pq \in V$ where $V$ in (2), followed by $mn - k + 1$ constraints $0 \leq Y_{pq}$ for $pq \in V \cup \{i,j,j_1\}$, and $k - 1$ constraints $Y_{i\ell,j_1} \leq Y_{i\ell+1,j_1+1}$ for $\ell \in [k - 1]$. Write $\lambda_{pq}$, $\delta_{pq}$ and $\eta_{\ell}$ for their respective Lagrangian variables and the Karush-Kuhn-Tucker (KKT) conditions:

$$
Y_{pq} = \begin{cases} 
X_{pq} - \lambda_{pq} + \delta_{pq} & \text{for all } pq \in V \\
X_{i\ell,j_1} - \eta_1 + \sum_{pq \in V} \lambda_{pq} + \delta_{i\ell,j_1} & \text{for all } pq = i,j_1 \\
X_{i\ell,j_1} + \eta_{\ell-1} - \eta_\ell & \text{for all } pq = i,j_\ell \text{ where } 2 \leq \ell \leq k
\end{cases}
$$

(24)

$$
Y_{pq} \leq Y_{i,j_1}, \text{ for all } pq \in V, \text{ and } Y_{pq} \geq 0 \text{ for all } pq \in V \cup \{i,j_1\}
$$

(25)

$$
\lambda_{pq} \geq 0, \text{ for all } pq \in V, \text{ and } \delta_{pq} \geq 0 \text{ for all } pq \in V \cup \{i,j_1\}
$$

(26)

$$
\lambda_{pq}(Y_{pq} - Y_{i,j_1}) = 0, \text{ for all } pq \in V, \text{ and } \delta_{pq}Y_{pq} = 0 \text{ for all } pq \in V \cup \{i,j_1\}
$$

(27)

$$
Y_{i\ell,j_\ell} \leq Y_{i\ell+1,j_{\ell+1}}, \text{ for all } \ell \in [k - 1]
$$

(28)

$$
\eta_{\ell} \geq 0, \text{ for all } \ell \in [k - 1]
$$

(29)

$$
\eta_{i,j_\ell} - Y_{i,j_\ell} = 0, \text{ for all } \ell \in [k - 1]
$$

(30)

Lemma B.1. Given any index $ij = i_1j_1$ and positive $\eta \geq 0$, any $X_{pq} \in \mathbb{R}$ for all $pq \in V \cup \{ij\}$, let $r = r(\eta) = \text{rank}(X_{ij} - \eta)$ over the $mn - k + 1$ variates. Then for both $\tau(\cdot) = \tau(\cdot, \eta)$ and $0 \leq t = t(\eta) < r$ (see both (14) and (15) in main text) we have

$$
\tau(t, \eta) \geq X_{(t+1)}, \text{ and if } t > 0 \text{ also } \tau(t, \eta) \leq X_{(t)} \leq X_{(t-1)} \leq \cdots \leq X_{(1)}.
$$

(31)

Proof of Lemma B.1. Assume $\eta = 0$ and drop $\eta$ from $\tau$ and $t$, since the case $\eta > 0$ is equivalent to the case $X'_{pq} = X_{pq}$ and $X'_{ij} = X_{ij} - \eta$. Then, the statement (31) holds if (i) $X_{(t)} \geq \tau(t)$ and (ii) $\tau(t) \geq X_{(t+1)}$. Fact (i) is relevant only when $t > 0$, and holds by definition (see (15) in main text) because $t + 1$ is minimal in $[r]$ for which $\tau(t + 1) > X_{(t+1)}$ holds. Fact (ii) holds in either of the possible two cases:

Case I: $[t + 1 = r]$. In this case, note $\tau(t)$ is in fact the unweighted average of $X_{(t+1)} = X_{ij}$ and values $X_{pq}$ that have higher rank $(X_{pq}) > r$. Therefore we must conclude $\tau(t) \geq X_{(t+1)}$.

Case II: $[t + 1 < r]$. In this case, note that by definition of $t$ we must have $\tau(t + 1) > X_{(t+1)}$ is satisfied. Together with $(t + 1) \cdot \tau(t + 1) = t \cdot \tau(t) + X_{(t+1)}$ we have $\tau(t) > \tau(t + 1)$. Therefore we conclude $\tau(t) > \tau(t + 1) > X_{(t+1)}$.

Lemma B.2. Given any index $ij = i_1j_1$ and positive $\eta \geq 0$, any $X_{pq} \in \mathbb{R}$ for all $pq \in V \cup \{ij\}$, for where at least one $X_{pq}$ in $V \cup \{ij\}$ is negative. Let $1 \leq s \leq t$ denote the least-ranked element that is negative in the ranking $X_{(1)} \geq \cdots \geq X_{(s-1)} \geq 0 > X_{(s-1)} \geq \cdots \geq X_{(t)}$ for $t = t(\eta)$ satisfying (15) in main text. Then for $\tau(\cdot) = \tau(\cdot, \eta)$ in (14), if $\tau(t) < 0$, we necessarily have:

$$
X_{ij} < 0, \text{ and } \sum_{t=1}^{s-1} X_{(t)} < -X_{ij}.
$$

(32)

Proof of Lemma B.2. Assume $\eta = 0$ and drop $\eta$ from $\tau$ and $t$, since the case $\eta > 0$ is equivalent to the case $X'_{pq} = X_{pq}$ and $X'_{ij} = X_{ij} - \eta$. We get that $\tau(s) = (s + 1)^{-1} \cdot (\sum_{t=1}^{s-1} X_{(t)} + X_{(s)} + X_{ij}) \leq X_{(s)}$, where the inequality follows from the minimality of $t$ in (15). Re-arranging, we get $\sum_{t=1}^{s-1} X_{(t)} + X_{ij} \leq X_{(s)}$. On the other hand, we had assumed $X_{(s)} < 0$, so therefore we must conclude (32).

Lemma B.3. Given index $ij = i_1j_1$ and $\eta \geq 0$, coefficients $X_{pq} \in \mathbb{R}$ for all $pq \in V \cup \{ij\}$, let $r = r(\eta) = \text{rank}(X_{ij} - \eta)$ over the $mn - k + 1$ coefficients. Let $\tau(\cdot) = \tau(\cdot, \eta)$ and $0 \leq t = t(\eta)$ respectively satisfy (14) and (15) in the main text. For all $pq \in V \cup \{ij\}$, set $Y_{pq}$ as: i) if rank $(X_{pq}) \leq t$ or $pq = i,j$ then set $Y_{pq} = (\tau(t, \eta))_+$, and ii) if otherwise then set
\( Y_{pq} = (X_{pq})_+ \). Then the \( mn - k \) and \( mn - k + 1 \) coefficients \( \lambda_{pq} \) and \( \delta_{pq} \) as determined by the first two lines of (24), given this choice for \( Y_{pq} \) and \( X_{pq} \), satisfy (25)-(27) together with the chosen \( Y_{pq} \) and \( X_{pq} \).

**Proof of Lemma B.3.** Fix any \( ij = i_jj_1 \). Assume \( \eta = 0 \), because the general case \( \eta \geq 0 \) is equivalent to the case \( X'_{pq} = X_{pq} \) and \( X'_{ij} = X_{ij} - \eta \), hence we simply write \( \tau(\cdot) = \tau(\cdot, \eta) \) and \( t = t(\eta) \). We show that (33)-(34) below

\[
\begin{aligned}
&\text{(for all } pq \in V) \quad Y_{pq} = \begin{cases} 
(\tau(t))_+ & \text{if rank } (X_{pq}) \leq t \\
(X_{pq})_+ & \text{otherwise}
\end{cases} \quad (33) \\
&Y_{ij} = (\tau(t))_+ , \quad (34)
\end{aligned}
\]

(together with (31) and (32) in Lemma B.1 and B.2 respectively, establish \( \lambda_{pq} \) and \( \delta_{pq} \) that relate \( Y_{pq} \) and \( X_{pq} \) by the first two equations of (24), and satisfy the three KKT conditions (25), (26) and (27) for \( i_jj_1 = ij \), proving the lemma.

**KKT condition (25).** We first show \( Y_{pq} \leq Y_{ij} \) for all \( pq \in V \). Consider rank \( (X_{pq}) > t \). Then wherever \( X_{pq} \geq 0 \) we have \( 0 \leq Y_{pq} \leq X_{pq} \leq X_{(t+1)} \leq \tau(t) \), and whenever \( X_{pq} < 0 \) we have \( Y_{pq} = (X_{pq})_+ = 0 \leq (\tau(t))_+ = Y_{ij} \). For all other \( pq \in V \) where rank \( (X_{pq}) \leq t \) we see that \( Y_{pq} \leq (\tau(t))_+ = Y_{ij} \). Thus the first set of inequalities in (25) follow. The second set \( Y_{pq} \geq 0 \) for all \( V \cup \{ij\} \) by definitions (33) and (34).

We next show (26) and (27) separately for \( pq \in V \); for \( pq \in V \) we choose \( \lambda_{pq} \), \( \delta_{pq} \) satisfying \( X_{pq} + \delta_{pq} = \lambda_{pq} + Y_{pq} \) as

\[
\begin{aligned}
&\text{(if rank } (X_{pq}) > t \text{)} \quad \Rightarrow \quad \lambda_{pq} = 0 , \quad \text{and } \delta_{pq} = (X_{pq})_+ - X_{pq} , \quad (35) \\
&\text{(if rank } (X_{pq}) \leq t \text{)} \quad \Rightarrow \quad \begin{cases} 
\lambda_{pq} = X_{pq} - (\tau(t))_+ , \quad \delta_{pq} = 0 , & \text{if } \tau(t) \geq 0 \text{ or } X_{pq} \geq 0 \\
\lambda_{pq} = 0 , \quad \delta_{pq} = -X_{pq} , & \text{if } \tau(t) < 0 \text{ and } X_{pq} < 0
\end{cases}
\end{aligned}
\]

Verify (35) satisfies (first equation of) (24) by putting \( Y_{pq} = (X_{pq})_+ \), as in (33), and similarly verify (36) satisfies the same by putting \( Y_{pq} = (\tau(t))_+ \) whilst considering both cases \( \tau(t) \geq 0 \) and \( \tau(t) < 0 \). Following that (26) and (27) for the remaining index \( ij \) will be shown after.

**KKT condition (26) for \( pq \in V \).** Consider rank \( (X_{pq}) > t \). From (35), this is trivially satisfied for \( \lambda_{pq} \), and holds for \( \delta_{pq} \) because \( (X_{pq})_+ - X_{pq} \geq 0 \). For rank \( (X_{pq}) \leq t \) we see from (36) the following. We see \( \lambda_{pq} = X_{pq} - (\tau(t))_+ \geq 0 \) holds in the event where \( \tau(t) \geq 0 \) or \( X_{pq} \geq 0 \) holds, since \( \tau(t) \leq X_{pq} \) whenever rank \( (X_{pq}) \leq t \). Also in the event where both \( \tau(t) < 0 \) and \( X_{pq} < 0 \) hold, we have that \( \delta_{pq} = -X_{pq} \geq 0 \) because then we have \( X_{pq} \) to be negative.

**KKT condition (27) for \( pq \in V \).** The first set of equalities in (27) satisfy as follows. Whenever rank \( (X_{pq}) \leq t \), we have \( Y_{pq} = Y_{ij} \), see above discussion on KKT condition (25). For rank \( (X_{pq}) > t \), we had chosen \( \lambda_{pq} = 0 \), see (35). The second set of equalities in (27) satisfy as follows: for case rank \( (X_{pq}) > t \) this is seen given choice (33) putting \( Y_{pq} = (X_{pq})_+ \) and choice (35) for \( \delta_{pq} \), and for rank \( (X_{pq}) \leq t \) this is seen given (33) putting \( Y_{pq} = (\tau(t))_+ \) and choice (36) for \( \delta_{pq} \).

It remains to show KKT conditions \( \delta_{ij} \geq 0 \) in (26) and \( \delta_{ij} Y_{ij} = 0 \) in (27). Derive the following:

\[
\sum_{pq \in V} \lambda_{pq} (X_{pq} + \delta_{pq} - (\tau(t))_+) = \sum_{pq \in V: \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq} - t \cdot (\tau(t))_+ .
\]

Then put \( Y_{ij} = (\tau(t))_+ \) from (34) in \( \delta_{ij} = Y_{ij} - X_{ij} - \sum_{pq \in V} \lambda_{ij} \), and put in the derivation above to get \( \delta_{ij} = (t+1) \cdot (\tau(t))_+ - (X_{ij} + \sum_{pq \in V: \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq})) \), and simplify to:

\[
\delta_{ij} = \begin{cases} 
(t+1)\tau(t) - (X_{ij} + \sum_{pq \in V: \text{rank}(X_{pq}) \leq t} X_{pq}) = (t+1)\tau(t) - (t+1)\tau(t) = 0 & \text{if } \tau(t) \geq 0 \\
-X_{ij} - \sum_{pq \in V: \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq}) = -X_{ij} - \sum_{pq \in V: X_{pq} \geq 0} X_{pq} > 0 , & \text{if } \tau(t) < 0
\end{cases}
\]

showing both (26) and (27) hold, where we had used (32) from Lemma B.2 for the final inequality in the \( \tau(t) < 0 \) case.

**Lemma B.4.** For \( \tau(\cdot, \eta) \) and \( t(\eta) \) in (14) and (15) in the main text, consider the function \( T : \mathbb{R}_+ \to \mathbb{R} \) defined as \( T(\eta) := (\tau(t(\eta), \eta))_+ \) for any \( \eta \geq 0 \). Then \( T \) is piecewise-linear, monotonic non-increasing and convex in \( \eta \geq 0 \).
Proof of Lemma B.4. \( \tau(s, \eta) \) is decreasing linear function of \( \eta \) for fixed \( s \), therefore \( T(\eta) = (\tau(t(\eta), \eta))._+ = \max(\tau(t(\eta), \eta), 0) \) is a convex, piecewise linear function with inflection points whenever \( \ell(\eta) \) changes value, with the final inflection point occurring when \( \tau(t(\eta), \eta) \) meets the horizontal axis. \( \square \)

Lemma B.5 ((Grotzinger & Witzgall, 1984)). For \( p, q, r, s \in [k] \) satisfying \( 1 \leq p \leq q \leq r \leq s \leq k \) where \( r = q + 1 \), assume \( \Delta_{pq} \leq \Delta_{qs} \) for \( \Delta \) in (37). Let \( \ell \) satisfy \( p \leq \ell \leq s \). Then for all \( \ell \leq q \) we have \( \psi_{pq} = (\ell - p + 1) \Delta_{pq} \geq 0 \), and for all \( \ell > q \) we have we have \( \psi_{pq} = (\ell - r + 1) \Delta_{rs} \geq 0 \).

Restatement of Proposition 3.2 For \( C_2 = O_{i,j,k} \) for any \( i, j, k \in \{m,n\} \) where \( \ell \in [k] \), consider the Euclidean projector \( \text{Proj}_{C_2}(X) \) for any \( X \in \mathbb{R}^{m \times n} \). Let \( T(\eta) := (\tau(t(\eta), \eta))_+ \) for \( \tau, t \) in (14) and (15) in the main text, and let \( V \) in (2). Then for any \( X \in \mathbb{R}^{m \times n} \), ePAVA will successfully terminate with some \( B, \eta, v \), and \( \forall \ell \). Furthermore, the projection \( \tilde{X} = \text{Proj}_{C_2}(x) \) satisfies i) for \( pq \in V \) we have \( \tilde{X}_{pq} = T(\tilde{\eta}) = \forall \alpha \{1 \} \) if rank \( (\tilde{X}_{pq}) \leq t(\tilde{\eta}) \) or \( \tilde{X}_{pq} = X_{pq} \) otherwise, and ii) for \( \ell \in [k] \) we have \( \tilde{X}_{i\ell,j\ell} = \forall \alpha \{B'\} \) iff \( \ell \in \{B'\} \leq \ell \leq \ell + 1 \) for some \( B' \in \{B\} \).

Proof. Proving Proposition 3.2 involves exhibiting a \( Y \in \mathbb{R}^{m \times n} \) satisfying (24)–(30). Given variables \( X_{i\ell,j\ell} \) where \( i, j, \ell \notin V \) for all \( \ell \in [k] \), define:

\[
\psi_{pq} = \sum_{\ell=p}^{q} X_{i\ell,j\ell}, \quad \Delta_{pq} = \frac{\psi_{pq}}{q - p + 1}, \quad \text{for } 1 \leq p \leq q \leq k.
\]

(37)

The block of operations between Lines 4 and 10 of Alg. 2 is termed “coalescing” (Grotzinger & Witzgall, 1984). As coalescing occurs during iterates, in the case whenever \( B = 2 \), notice the parameter denoted \( \tilde{\eta} \) in Alg. 2 (Line 7) that updates. We prove that if this parameter is taken to be \( \eta_1 \) in the KKT conditions (24) and (29)–(30), that Alg. 2 terminates and when it does with \( \tilde{X} \), that \( \tilde{X} \) satisfies the KKT. The proof is recursive in \( \eta_1 \) and we initialize \( \eta_1 = 0 \). We use \( \eta_\ell \) and \( \eta_\ell' \) to denote the current, and next iterate, respectively.

We first invoke (24) to fix the relationship between solution and Lagrangian multipliers. For a given \( \eta_1 \) value, put \( \eta_1 = \eta \) in Lemma B.3 to get \( mn - k + 1 \) coefficients \( Y_{pq} \) (and the Lagrangians \( \lambda_{pq}, \delta_{pq} \)) for all \( pq \in V \cup \{i,j,1\} \) satisfying the first two lines of (24) and (25)–(27) of the KKT conditions. Likewise for \( \eta_1 \), we use (24) to derive the \( k - 1 \) Lagrangians that accompany \( Y_{i\ell,j\ell} \) as follows. For any \( B' \in \{B\} \), ePAVA in Alg. 2 sets \( Y_{i\ell,j\ell} = \tilde{Y}_{i\ell+1,j\ell+1} \) for values of \( \ell \) that satisfy \( \{B'\} \leq \ell < \ell + 1 \leq \ell + 1 \) in \( \{B'\} \). On the other hand if \( \ell \) is on the boundary \( \ell = \ell + 1 \) in \( \{B'\} \) then ePAVA results in \( Y_{i\ell,j\ell} \neq \tilde{Y}_{i\ell+1,j\ell+1} \). Therefore using \( \psi_{pq} \) and \( \Delta_{pq} \) from (37) and \( \nu = \sum_{pq \in V} \lambda_{pq} + \delta_{pq} \eta_{1,j1} \), we express \( Y_{i\ell,j\ell} \) and \( \eta_\ell \) for all \( \ell \in [k] \):

\[
\eta_\ell = \psi_{pq} - \ell \cdot T(\eta_1) + \nu, \quad Y_{i\ell,j\ell} = T(\eta_1), \quad \text{if } 1 \leq \ell \leq \ell + 1, \quad (38)
\]

\[
\eta_\ell = \psi_{pq} - (\ell - q + 1) \Delta_{pq}, \quad Y_{i\ell,j\ell} = \Delta_{pq}, \quad \text{if } p = \ell \in \{B'\}, \leq \ell \leq q \leq \ell + 1 \quad (39)
\]

where (39) is satisfied for all \( 1 < B' \leq B \), and simply let \( \eta_0 = 0 \) since this does not contradict (37). Thus it remains to show, as the iterates of \( \eta_1 \) update the values of \( Y_{pq}, \lambda_{pq}, \delta_{pq}, \eta_1 \) by Lemma B.3 and (38)–(39), the remaining KKT conditions (28)–(30) are satisfied upon termination of ePAVA. To this end we must prove the existence and required properties of the zero (i.e., \( \tilde{\eta} \), taken here to be \( \eta_1 \) in Line 7. Specifically, let \( \eta = \eta_1 \) be the boundary of block 1. Suppose either a) \( \eta_1 = 0 \) and \( q = 1 \), or b) \( \psi_{pq} - (q - 1) \cdot T(\eta_1) + \eta_1 = 0 \) is satisfied for some \( \eta_1 > 0 \) and \( q > 1 \). Let \( r = q + 1 \) and consider \( \psi_{2s} \) from (37) for some \( s \geq r \geq 2 \). We show below that in either case a) or b), that if \( \Delta_{rs} \leq T(\eta_1) \) holds, we must have that i) there exists a zero \( \eta_1' \) satisfying \( \psi_{2s} = (s - 1) \cdot T(\eta_1') \) and \( \eta_1' \), and ii) \( T(\eta_1') \leq T(\eta_1) \) and \( \eta_1' \geq \eta_1 \).

The proof of the above properties of the zero follow. In the case \( \eta_1 = 0 \) and \( q = 1 \), then i) follows by equivalently showing if \( T(\eta_1') - \eta_1'/\{s - 1\} - \Delta_{rs} \) has a zero (divide by \( s - 1 \) \( \geq 1 \) and use (37)). Indeed, the zero exists at some \( \eta_1' \geq 0 \) since the non-increasing \( T(\eta_1') \), see Lemma B.4, and an increasing linear function \( \eta_1'/\{s - 1\} + \Delta_{rs} \), meets, as the latter starts at a point lower than the former as given by the assumption \( \Delta_{rs} \leq T(\eta_1) \). Furthermore, ii) holds since we showed \( \eta_1' \geq 0 \) and by monotonicity of \( T(\cdot) \). In the other case put \( \Delta_{rs} = (1 - \beta) \Delta_{2s} + \beta \Delta_{rs} \) for \( 0 \leq \beta = (s - q)/(s - 1) \leq 1 \), and use the condition b) above to derive \( T(\eta_1) - \Delta_{2s} - \eta_1'/\{s - 1\} = \beta \cdot (T(\eta_1') - \Delta_{rs} \geq 0 \) where the inequality follows since we assumed \( \Delta_{rs} \leq T(\eta_1) \). Then by monotonicity of \( T(\cdot) \), see Lemma B.4, the zero exists and occurs at some point \( \eta_1' \geq \eta_1 \) with similar arguments as before showing i) and ii).
KKT condition (29) for block 1, case (38): We now show that for block 1, coalescing recursively maintains non-negativity (29). Suppose 1, q and r, s are boundaries of blocks 1 and 2, where \( r \equiv 1 = q = r = 1 \). Let \( \eta'_l \) for \( l \in [s] \) equal (38) after coalescing, rewritten into \( \eta'_l = \psi_{2q} - (\ell - 1)T(\eta'_1) + \eta'_s \), and similarly \( \eta_l \) for \( l \in [q] \) denotes (38) before coalescing. By recursion assumption either \( \eta_1 = 0 \) or \( \psi_{2q} - (q - 1) \cdot T(\eta_1) + \eta_1 = 0 \) is satisfied for some \( \eta_1 > 0 \). ePVA coalesces blocks 1 and 2 if \( \text{val}[1] \leq \text{val}[2] \), or equivalently, if \( \Delta_{rs} \leq T(\eta_1) \) for value \( \eta_1 \geq 0 \), and therefore by recursion assumption, we conclude a new zero \( \eta'_1 \geq 0 \) of Line 7 with properties i) and ii) above exists. Then for \( l \leq q \) we have \( \eta'_l - \eta_l = (\ell - 1)T(\eta_1') + \eta'_s - \eta_l \), and by the inequality \( \eta'_1 - \eta_l \geq 0 \), we conclude \( \eta'_l - \eta_l \geq 0 \). Next for \( l > q \), we express the following for some \( \alpha = 1 - (\ell - 1)/(s - 1) \geq 0 \):

\[
\eta'_l - \eta_l \overset{(a)}{=} \eta'_1 + \psi_{2q} - (\ell - 1) \cdot T(\eta'_1) + (\ell - q) \cdot \Delta_{rs} \overset{(b)}{=} \alpha \cdot (\eta'_1 + \psi_{2q} - (q - 1) \cdot \Delta_{rs}) \\
\overset{(c)}{\geq} \alpha (q - 1) \cdot (T(\eta_1) - \Delta_{rs}) \geq 0,
\]

where (a) follows from \( \eta'_l \) in (38) and \( \eta_l \) in (39), and (b) follows from expressing \( T(\eta_1') = [\psi_{2q} + (s - q) \cdot \Delta_{rs} + \eta'_s] / (s - 1) \) and collecting terms into \( \alpha \), and (c) follows as \( \eta'_1 + \psi_{2q} \geq \eta_1 + \psi_{2q} = (q - 1) \cdot T(\eta_1) \), and finally the last inequality follows because we only coalesce when \( \Delta_{rs} \leq T(\eta_1) \).

KKT condition (29) for block > 1, case (39): We show the similar non-negativity property for other blocks. Suppose \( p, q \) and \( r, s \) are boundaries of two blocks \( B' \) and \( B' + 1 \), where \( r = q + 1 \) and \( B' + 1 \). ePVA coalesces block \( B' \) and \( B' + 1 \) only if \( \text{val}[B' - 1] \leq \text{val}[B'] \), or equivalently, \( \Delta_{ps} \leq \Delta_{rs} \). Consider any \( p \leq \ell \leq s \). Eqs. (39) implies that the Lagrangians before and after coalescing are \( \eta_l = \psi_{ps} - (\ell - p + 1) \Delta_{ps} \) and \( \eta'_l = \psi_{q} - (\ell - r) \Delta_{rs} \), respectively. Supposing \( \eta_1 \geq 0 \) and by these expressions for \( \eta'_l, \eta_l \), we thus invoke Lemma B.5 to show \( \eta'_l \geq 0 \) for all \( l < q \) hold after coalescing; the other case \( \ell > q \) also holds similarly by Lemma B.5.

KKT condition (30): We show the coalescing update Lines 4-10, maintains the boundary \( s = \Pi_i[B'] \) for any \( B' \in [B] \) property \( \eta_s = 0 \); this proves complementary slackness (30) because (38)–(39) shows \( Y_{i,j} \) differ only across boundaries. Let \( \eta_q \) and \( \eta'_q \) denote boundary Lagrangians for the previous and current coalescing update, respectively; the boundaries \( p, q \) and \( r, s \) for \( r = q + 1 \) are coalesced. For (39) for \( B' + 1 \), we have for \( \eta'_s = \psi_{ps} - (s - p + 1) \Delta_{ps} \) at the boundary \( \ell = s \), and \( \eta_p = 0 \) by definition (37). For (38) for \( B' + 1 \), rewrite (38) to get the form \( \eta'_s = \psi_{ps} - (s - 1) \cdot T(\eta'_1) + \eta'_1 \) resembling the equation with the zero shown above. If the coalescing update is executed for the first time, then \( \eta_1 = 0 \) and \( q = 1 \). Otherwise \( \eta_1 > 0 \) and \( q > 1 \) and by recursion assumption \( \eta_q = \psi_{2q} - (q - 1) \cdot T(\eta_1) + \eta_1 = 0 \). Therefore in either cases, together with the condition \( \Delta_{rs} \leq T(\eta_1) \) that holds for a coalescing step to occur, we conclude by zero property i) that \( \eta'_s = 0 \).

KKT condition (28): Each coalescing step of ePVA attempts to restore a non-increasing property of \( Y_{i,j} \) for all \( \ell \in [k] \). This is only possible if the new values \( Y_{i,j} \) that result from coalescing does not increase beyond the values before coalescing. This is obvious for the case of \( B > 2 \) by definition (37). ePVA eventually terminates satisfying (28), since there are finite number of coalescing steps, and in the worst case arrives at the terminating state \( B = 1 \) and \( \Pi_i[1] = k \) is arrived at, which satisfies (28).

We now turn to the Euclidean projection \( \text{Proj}_{C_1(a,b)}(X) \) of a matrix \( X \in \mathbb{R}^{m \times n} \) onto the row- and column-sum constraint set \( C_1(a,b) \) for measures \( a, b \). Let \( \otimes \) denote the matrix Kronecker (i.e., tensor) product. Observe that the rowsums of any \( X \in \mathbb{R}^{m \times n} \) can also be obtained via Kronecker product \( X_{1_m} = (I_m \otimes 1^T_n)x \), where \( x \in \mathbb{R}^{mn} \) is a length-\( mn \) vector formed from \( X \in \mathbb{R}^{m \times n} \) by setting \( x_{(k-1)n+i} = X_{ki} \). Also be the same equivalence we obtain column-sums of \( X \in \mathbb{R}^{m \times n} \) \( X^T1_m = (1_m \otimes I_n)x \) from the same equivalent \( x \in \mathbb{R}^{mn} \); in numpy terminology one writes \( x = \text{ravel}(X) \). We thus conclude that Euclidean projection \( X = \text{Proj}_{C_1(a,b)}(X) \) for any \( X \in \mathbb{R}^{m \times n} \), can be equivalently obtained by solving the following optimization over \( \mathbb{R}^{mn} \):

\[
\hat{x} = \arg \min_{y \in \mathbb{R}^{mn}} \frac{1}{2} \| y - x \|_2^2, \text{ s.t. } (I_m \otimes 1^T_n)y = a, \quad (1^T_m \otimes I_n)y = b
\]

and then setting \( \hat{X} = \text{ravel}(\hat{x}) \), where \( \text{ravel}(\cdot) \) is the operational inverse of numpy function \( \text{ravel}(\cdot) \).

Converting the problem to vectors (40) results in KKT conditions with \((m + n) \) (Lagrangian) variables \( \alpha \in \mathbb{R}^m \) and

https://numpy.org/doc/stable/reference/generated/numpy.ravel.html
$\beta \in \mathbb{R}^n$ in a system of linear equations:

$$
\begin{bmatrix}
I_{mn} & A^T \\
A & I
\end{bmatrix} \begin{bmatrix}
y \\
\alpha \\
\beta
\end{bmatrix} = \begin{bmatrix}
x \\
\alpha \\
\beta
\end{bmatrix}
$$

(41)

where here $A$ is an $(m+n) \times mn$ matrix with first $m$ rows equal $I_m \otimes 1^T_n$ and last $n$ rows equal $1^T_m \otimes I_n$. That is to compute $\hat{X} = \text{Proj}_{C,(a,b)} (X)$, one may simply solve (41) for $y$, given $x = \text{ravel} (X)$, $a$ and $b$. However we have to pay attention that $A$ in (41) is not full-rank. To compute the exact rank of $A$, we can make use of Remark 3.1 from (Peyré & Cuturi, 2019); specifically, it is $m+n-1$. This implies that the left-inverse of $A$ is not equal to $(A^T A)^{-1} A^T$, and instead requires the pseudo-inverse of the matrix $\mathcal{A} \in \mathbb{R}^{(m+n) \times mn}$ in (41):

$$
A^\dagger = \begin{bmatrix}
\frac{1}{m} (I_m \otimes 1_n - \frac{1}{m+n}) & \frac{1}{m} (I_m \otimes I_n - \frac{1}{m+n})
\end{bmatrix}
$$

Recall $P_k$ is the projection $P_k = \frac{1}{k} 1_k 1_k^T$, and matrices $M := I_m - \frac{m}{m+n} P_m$ and $N := I_n - \frac{n}{m+n} P_n$. Then (42) will prove Prop. 3.1, restated here in the Kronecker equivalent form (40) corresponding to numpy function $x = \text{ravel} (X)$.

**Lemma B.6.** Assume $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ have same means, i.e., $a^T 1_n = b^T 1_n$. Then for any $x \in \mathbb{R}^{m \times n}$, the solution of (41) is given by the the following expression for $y \in \mathbb{R}^{mn}$ for some $\alpha \in \mathbb{R}^m$ and $\beta \in \mathbb{R}^n$, where

$$
y = A^\dagger \begin{bmatrix}
a \\
b
\end{bmatrix} + \left( I_{mn} - A^T (A^T)^\dagger \right) x
$$

(42)

where $A^\dagger$ is the pseudo-inverse of $A \in \mathbb{R}^{(m+n) \times mn}$ in (41).

**Proof of Lemma B.6.** Suppose $\begin{bmatrix}
a \\
b
\end{bmatrix}$ lies in span $(A)$, then our strategy is to show $y \in \mathbb{R}^{mn}$ in (42) satisfies both top (i.e., first $m$ rows) and bottom (i.e., last $n$ rows) of (41), in two steps. From the Moore-Penrose property $AA^\dagger A = A$ of the pseudo-inverse, we conclude that $A^\dagger$ has a left-inverse property over span $(A)$. Thus, we can construct some $y = A^\dagger \begin{bmatrix}
a \\
b
\end{bmatrix} + \nu$ with an unrestricted choice for $\nu \in \text{null}(A)$, and $y$ will satisfy the bottom of (41). Next we want to show our construction for $y$ also satisfies the top of (41). To do this, we show there exists some $\alpha \in \mathbb{R}^m, \beta \in \mathbb{R}^n$ that satisfies $A^T \begin{bmatrix}
\alpha \\
\beta
\end{bmatrix} = x - \nu - A^\dagger \begin{bmatrix}
a \\
b
\end{bmatrix}$, for some specific $\nu \in \text{null}(A)$. Indeed for any $x \in \mathbb{R}^{mn}$, there exists some $\nu \in \text{null}(A)$ such that $x - \nu$ equals the projection of $x$ onto span $(A^T)$, i.e., there exists $\nu$ such that $x - \nu = A^T (A^T)^\dagger x$; to check write $\nu = \left( I_{mn} - A^T (A^T)^\dagger \right) x$ and observe that $\nu$ is then the projection of any $x \in \mathbb{R}^{mn}$ onto $(\text{span}(A^T))^\perp$, and observe that $(\text{span}(A^T))^\perp = \text{null}(A)$, agreeing with our assumption $\nu \in \text{null}(A)$. Thus, we have shown that $y = A^\dagger \begin{bmatrix}
a \\
b
\end{bmatrix} + \nu$ with $\nu = \left( I_{mn} - A^T (A^T)^\dagger \right) x$ satisfies the optimality conditions (41), and this form of $y$ is exactly that of (42).

To conclude we need to prove the starting assumption $\begin{bmatrix}
a \\
b
\end{bmatrix} \in \text{span}(A)$. Because we assumed $1^T a = 1^T b$ we have $\begin{bmatrix}
1_m \\
-1_n
\end{bmatrix}$ to be $\perp$ to $\begin{bmatrix}
a \\
b
\end{bmatrix}$. On the other hand $\begin{bmatrix}
1_m \\
-1_n
\end{bmatrix}$ is in fact in $(\text{span}(A))^\perp$. Therefore $\begin{bmatrix}
a \\
b
\end{bmatrix} \in \text{span}(A)$.

**Restatement of Proposition 3.1** For any $x \in \mathbb{R}^{mn}$, the Euclidean projection $\hat{x}$ solving (40) satisfies $\hat{x} = y_1 + (x - y_2)$, where $y_1 = \frac{1}{n}[M \otimes 1_n] a + \frac{1}{m}[1_n \otimes N] b$ and $y_2 = [M \otimes P_n] x + [P_m \otimes N] x$ for $P_m, P_n, M$ and $N$ defined as before.

**Proof of Proposition 3.1.** Put expression for $A^\dagger$ into the expression for $y$ in (42), set $\hat{x} = y$, and apply basic manipulations. \footnote{This can be verified to satisfy the conditions in p. 257, Golub, Gene H. and Van Loan, Charles F., Matrix Computations, Johns Hopkins University Press, Third Edition, 1996.}
B.2. Bounds

The goal here is to extend the result in Kusner et al. (2015) and obtain a lower bound (see (19) in main text) on the optimal cost (6) with order constraints. This requires a lower bound (see (23) in main text) on the function \(g_{ij}(x, \varphi)\) that appears in (19) in the main text. Prop. 4.1 derives this lower bound by decoupling the expression into \((m\text{ or }n)\) independent minimizations that can be computed in parallel; Kusner et al. (2015) takes the same approach for the simpler problem (1), but here a global constraint (due to the order constraints) still has to be dealt with (see equation below (19) in main text).

Here the independent minimizations are related to \(\text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, u, \alpha\right)\), see (20) in the main text, where \(u\) and \(\alpha\) represent the per item \(x_k\) capacity and total budget, respectively. We first present Lemmata B.7–B.8 that show the convex piecewise-linear and monotone non-increasing in \(u\).

**Lemma B.7.** The solution to \(\text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, u, \alpha\right)\) for \(0 \leq \alpha/n < u \leq \alpha \leq 1\) is given by

\[
\text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, u, \alpha\right) = u \left[ \sum_{k=1}^{\ell} (\varphi(k) - \varphi(\ell+1)) \right] + \alpha \varphi(\ell+1)
\]

(43)

where \(\ell = \lfloor \alpha/u \rfloor < n\), and \(\varphi(k)\) is the \(k\)-th coefficient in increasing order \(\varphi(1) \leq \varphi(2) \leq \cdots \leq \varphi(n)\). For fixed \(\alpha\), it is convex piecewise-linear and monotone non-increasing in \(u\), with inflection points at \(u = \alpha/i\) for \(i = n - 1, n - 2, \cdots, 1\).

**Proof.** The solution (of (20), main text) is by greedy prioritization of lower costs in the ordering \(\varphi(k)\), hence:

\[
\text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, u, \alpha\right) = \sum_{k=1}^{\ell} \varphi(k)u + (\alpha - \ell \cdot u)\varphi(\ell+1)
\]

(44)

where \(\ell = \lfloor \alpha/u \rfloor < n\), and clearly \(\alpha/n \leq u \leq \alpha\) is required to have at least one feasible solution. Gathering coefficients of \(u\) we arrive at (43) from (44), and from (43) it is clearly piecewise-linear with inflection points at values of \(u\) that cause \(\ell = \lfloor \alpha/u \rfloor\) to change value; there are \(n - 1\) of them at values \(u = \alpha/i\) for \(i = n - 1, n - 2, \cdots, 1\) making a total of \(n\) intervals \((\alpha/i + 1, \alpha/i]\). Pick some \(i < n\), fix some \(0 \leq \alpha \leq 1\), and consider \(u\) in the interval \((\alpha/(i + 1), \alpha/i]\)

For our choice of \(i\) we conclude the sequence of inequalities \(i + 1 = \left\lfloor \frac{\alpha}{\alpha/(i+1)} \right\rfloor > \left\lfloor \frac{\alpha}{\alpha/i} \right\rfloor = \ell \geq \left\lfloor \frac{\alpha}{\alpha/i} \right\rfloor = i\) and so we conclude \(\ell = i\) for any \(u \in (\alpha/(i + 1), \alpha/i]\). The function (43) is piecewise-linear because evaluating the left-limit \(\alpha/(i + 1) \leftarrow u\) of interval \((\alpha/(i + 1), \alpha/i]\) (when \(\ell = i\)), equals the value evaluated at the rightmost point of the neighboring interval \((\alpha/(i + 2), \alpha/(i + 1)]\) (when \(\ell = i + 1\)):

\[
\text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, \frac{\alpha}{i + 1}, \alpha\right) = \lim_{x \downarrow \alpha/i \leftarrow u} \text{PACKING}\left(\{\varphi_k\}_{k \in [n]}, u, \alpha\right) = \frac{\alpha}{i + 1} \sum_{k=1}^{i+1} \varphi(k).
\]

From (44) it the gradient is non-positive everywhere due to the non-positivity of the summands \(\varphi(k) - \varphi(i+1)\leq 0\) since \(k \leq \ell\), implying (44) is monotone non-increasing in \(u\) everywhere. A monotone non-increasing function is convex if its gradient is non-decreasing. Indeed, if we subtract the gradient of the \(i\)-th interval (where \(\ell = i\)), to the from that of the \((i+1)\)-th interval to the right (where \(\ell = i+1\)):

\[
\sum_{k=1}^{i-1} (\varphi(k) - \varphi(i)) - \sum_{k=1}^{i} (\varphi(k) - \varphi(i+1)) = i \cdot (\varphi(i+1) - \varphi(i)) \geq 0.
\]

We have thus proved that (43) is piecewise-linear, monotone non-increasing and convex.

**Lemma B.8.** The solution to \(\text{PACKING}\left(\{\varphi_k\}_{k \in [n-1]}, u, \alpha - u\right)\) for \(0 \leq \alpha/n \leq x \leq \alpha \leq 1\) is:

\[
\text{PACKING}\left(\{\varphi_k\}_{k \in [n-1]}, u, \alpha - u\right) = u \left[ -\varphi(\ell) + \sum_{k=1}^{\ell-1} (\varphi(k) - \varphi(\ell)) \right] + \alpha \varphi(\ell)
\]

(45)

where \(\ell = \lfloor \alpha/u \rfloor\) and \(\varphi(k)\) is the \(k\)-th coefficient in increasing order \(\varphi(1) \leq \varphi(2) \leq \cdots \leq \varphi(n)\). For \(\alpha\) for a fixed \(\alpha\), it is convex piecewise-linear and non-increasing in \(u\) with the inflection points \(u = \alpha/i\) for \(i = n - 1, n - 2, \cdots, 1\).
Restatement of Proposition 4.1

For any \( i, j, k \) set that defines the order constrained \( OT \) (see (6) in main text) where \( i[k] = i_1, i_2, \ldots, i_k \) (and \( j[k] \)) do not repeat row (or column) indices, the minimum \( g_{ijk}(x, \varphi) \) (see (19) in main text) is lower-bounded by

\[
g_{ijk}(x, \varphi) \geq \begin{cases} L_{1ijk}(x, \varphi, a) & \text{for } \alpha_1 \leq x \leq \beta_1 \\ L_{2ijk}(x, \varphi, b) & \text{for } \alpha_2 \leq x \leq \beta_2 \end{cases}
\]

where \( L_{1ijk}(x, \varphi, a) \) and \( L_{2ijk}(x, \varphi, b) \) resp. equal \( \sum_{\ell \in R} \nu(x, \{\varphi_{\ell}\} \in [n]\{\{j_{\ell}\}, a_{\ell}\} + \sum_{p \notin i[k]} \mu(x, \{\varphi_{\ell}\} \in [n]\{a_{\ell}\}) \) and \( \sum_{\ell \in R} \nu(x, \{\varphi_{\ell}\} \in [n]\{i_{\ell}\}, b_{\ell}\) + \( \sum_{q \notin j[k]} \mu(x, \{\varphi_{\ell}\} \in [n]\{b_{\ell}\}) \).

Proof of Proposition 4.1. For brevity we only prove the top bound that exists for \( x \) in the range \( \alpha_1 \leq x \leq \beta_1 \) (noted as \( L_{1ijk}(x, \varphi, a) \) in (22) of main text); the other \( L_{2ijk}(x, \varphi, b) \) will follow similarly and is omitted.

The bound (see (22), main text) is obtained by relaxing the constraint set (a la (20), main text):

\[
\{ \Pi \in \mathbb{R}^{m \times n} : \Pi \in U(a, b), \Pi \in \text{OT}_{ijkl} \text{, } \Pi_{i_1j_1} = \cdots = \Pi_{i_kj_k} = x \}
\]

The \( p \)-th set on the RHS only involves coefficients \( \Pi_{pq} \) found in the \( k \)-th row; after relaxation and taking into account the linear form of \( g_{ijk}(x, \varphi) \), and the fact that for each \( \ell \)-th row \( i_{\ell} \), the \( j_{\ell} \)-th column does not repeat in \( j[k] \), we obtain \( m \) independent minimizations of the following form:

\[
\begin{align*}
\min \sum_{q \in [n]} \varphi_{pq} \Pi_{pq} & \quad \text{s.t. } \sum_{q \in [n]} \Pi_{pq} = a_p, \quad 0 \leq \Pi_{pq} \leq x, \quad \text{for } p \notin i[k] \\
\min \sum_{q \in [n]\{j[k]\}} \varphi_{iq} \Pi_{iq} & \quad \text{s.t. } \sum_{q \in [n]\{j[k]\}} \Pi_{iq} = a_i - x, \quad 0 \leq \Pi_{iq} \leq x, \quad \text{for } \ell \in [k]
\end{align*}
\]

where the sum of the \( m \) optimal costs of the \( m \) minimizations, lower bound the quantity \( g_{ijk}(x, \varphi) \). For any \( p \notin i[k] \), the \( p \)-th minimization has optimal cost \( \mu(x, \{\varphi_{pq}\} \in [n]\{a_{pq}\}) = \text{PACKING} \{(\varphi_{pq}) \in [n]\{a_{pq}\}, x, a_{pq}\} \), and similarly for \( \ell \in [k] \), the \( i_{\ell} \)-th row has optimal cost \( \nu(x, \{\varphi_{\ell}\} \in [n]\{i_{\ell}\}, a_{i_{\ell}}) = \text{PACKING} \{(\varphi_{\ell}) \in [n]\{i_{\ell}\}, x, a_{i_{\ell}} - x\) see (20) of main text. Lemmata B.7–B.8 applied to the PACKING problems above, obtains that the minimization for the \( p \)-th rows is only feasible for \( x \geq a_p/n \), and has the same optimal value for \( a_p \leq x \leq 1 \); therefore the global constraint across all minimizations is obtained as \( \alpha_1 = \max_{p \in [n]} a_p/n \) and \( \beta_1 = \max_{p \in [n]} a_p \).

\]