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 discovered by sparsifying 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".[1] 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

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

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]For readability of Fig. 2, very low scores were suppressed, making it appear that they do not sum to one though they do.

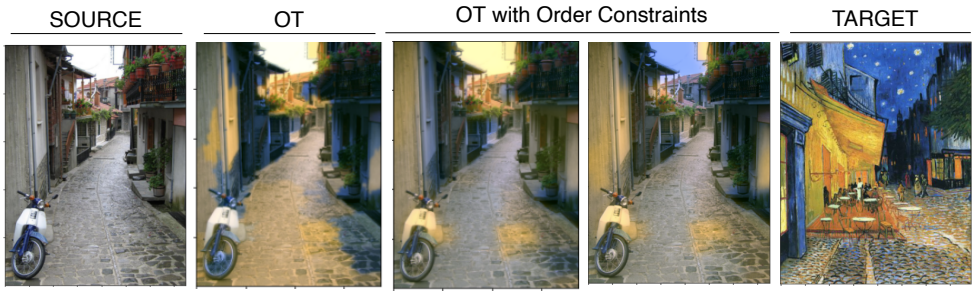

Figure 1: Color transfer using OT with Order Constraints

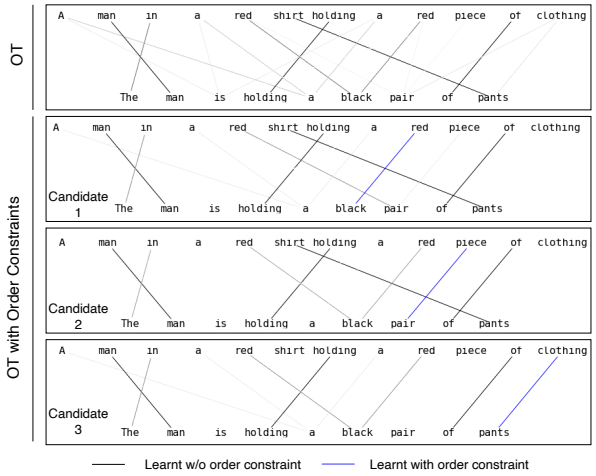

Figure 2: Text matching using OT with Order Constraints

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 explainablity. 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 intepretable 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-

tain 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; Scetbon 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, (Scetbon 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 $[n] := \{1, 2 \dots n\}$ and $[mn] := [m] \times [n]$. Let i, j, k, ℓ, p, q denote vector/matrix indices, $2^{[n]}$ denote the power set of $[n]$, and $\mathbf{1}_n$ and $\mathbf{I}_n$ denote the all-ones vector and identity matrix, resp., of size n. We denote the index sequence $i_1, i_2, \cdots, i_k$ as $i_{[k]}$. $\mathbb{I}_S(\ )$ is the indicator function over set S. The *trace* operator for square matrices is denoted by $\mathrm{tr}\,(\cdot)$ and $\mathrm{rank}\,(X_{ij})$ gives the descending positions over the mn matrix coefficients X_{ij}, or a subset thereof.[2] For a *closed set* $\mathcal{C}$, the Euclidean projector for a matrix $X \in \mathbb{R}^{m\times n}$ is $\mathrm{Proj}_{\mathcal{C}}(X) = \arg\min_{Y\in\mathcal{C}} \|Y - X\|_F^2$ 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:

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

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

$$\min_{\Pi\in U(a,b)} f(\Pi) := \mathrm{tr}\left(D^T\Pi\right), \tag{1}$$

and the optimal value of Π 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 Π_{ij} is semantically or symbolically important, then its rank (Π_{ij}) should reflect that dominance. We thus enforce k order constraints by specifying a sequence of k variates in $[mn]$, $ij_{[k]} := i_1j_1, i_2j_2, \cdots, i_kj_k$ of importance, where the set of variates $V := [mn] \setminus \{i_\ell j_\ell : \ell \in [k]\}$:

$$\Pi_{i_kj_k} \geq \cdots \geq \Pi_{i_1j_1} \geq \Pi_{pq} \ \text{ for } pq \in V. \tag{2}$$

Enforcing order constraints (2) means each variate $i_\ell j_\ell$ is fixed to maintain the $(k-\ell+1)$-th top position $\Pi_{(k-\ell+1)}$ in the ordering $\Pi_{(1)} \geq \Pi_{(2)} \geq \cdots \geq \Pi_{(mn)}$ whilst learning the transport plan Π; in particular i_kj_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:

$$\inf_{\Pi\in U(a,b)} f(\Pi) := \mathrm{tr}\left(D^T\Pi\right), \tag{3}$$

$$\text{s.t} \quad \Pi_{i_kj_k} \geq \cdots \geq \Pi_{i_1j_1}, \ \ \Pi_{i_1j_1} \geq \Pi_{pq} \ \text{ for } pq \in V, \tag{4}$$

for V in (2), and we assume non-negative costs $D \geq 0$ throughout. Let $O_{ij_{[k]}} \subseteq \mathbb{R}_+^{m\times n} =$

$$\{X \in \mathbb{R}_+^{m\times n} : \ \mathrm{rank}\,(X_{i_\ell j_\ell}) = k - \ell + 1, \ \forall \ell \in [k]\}. \tag{5}$$

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

$$\inf_{\Pi\in U(a,b)\cap O_{ij_{[k]}}} f(\Pi). \tag{6}$$

Problem (6) is convex with linear costs and constraints. However, $O_{ij_{[k]}}$ 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.

[2] This is not to be confused with the standard definition of a matrix's rank.

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 $ij_{[k]}$ where $i_{[k]} = i_1, i_2, \cdots, i_k$ (and $j_{[k]}$) do not repeat row (or column) indices; let $\mathbb{I}_{i_{[k]}}(p) = 1$ if $p = i_\ell$ for at most one choice of ℓ, or 0 otherwise, and similarly $\mathbb{I}_{j_{[k]}}(q)$. We next derive conditions for the feasibility of (6) under mild assumptions.

Proposition 2.1. *Suppose $i_{[k]} = i_1, i_2, \cdots, i_k$ (and $j_{[k]}$) do not repeat row (or column) indices. Further suppose $\Pi \in \mathbb{R}^{m\times n}$, where $\Pi_{i_\ell j_\ell} = c_\ell$ for $\ell \in [k]$, and Π_{pq} for $pq \in V$, resp. satisfy:*

$$0 \le \Pi_{i_\ell j_\ell} = c_\ell \le \min(a_{i_\ell}, b_{j_\ell}), \quad (7)$$

$$0 \le \Pi_{pq} = \frac{(a_p - \mathbb{I}_{i_{[k]}}(p)c_p)\cdot(b_q - \mathbb{I}_{j_{[k]}}(q)c_q)}{\alpha(c_1,\cdots,c_\ell)} \quad (8)$$

where $\alpha = \alpha(c_1,\cdots,c_\ell) = 1 - \sum_{\ell\in[k]} c_\ell \ge 0$. Then, Π is feasible w.r.t (6) if:

$$\frac{a_p b_q}{\alpha(c_1,\cdots,c_k)} \le c_k \le \cdots \le c_1 \ \text{ for all } \ pq \in V. \quad (9)$$

Proof. Since $i_{[k]}$ and $j_{[k]}$ do not repeat row/column indices, then (7) and (8) imply that Π satisfies $\Pi\mathbf{1}_n = a$ and $\Pi^T\mathbf{1}_m = b$; and since they also imply $\Pi \ge 0$, thus $\Pi \in U(a,b)$. Furthermore, we also have $\Pi \in O_{ij_{[k]}}$, because (8) and (9) imply for all $pq \in V$, that $\Pi_{pq} \le a_p b_q/\alpha \le c_k$. □

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 Π is feasible w.r.t (6) if $\min(m,n) \ge (1 - k/\max(m,n))^{-1}$.*

Proof. For all $\ell \in [k]$, pick $c_\ell = \min(1/m, 1/n) = c$ for some $c \ge 0$; this choice for c satisfies (7) and (8). Then (9) holds if $a_p b_q/c \le 1 - kc = \alpha$ holds, or equivalently $\min(m,n) \ge (1 - k/\max(m,n))^{-1}$ 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 $1 + \mathcal{O}(k/\max(m,n))$. 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 δ-approximate method for (6) that runs in $\mathcal{O}(\|D\|_\infty/\delta \cdot mn\log mn)$ 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 $U(a,b) \cap O_{ij_{[k]}}$ as the intersection of two closed convex sets:

$$\mathcal{C}_1(a,b) = \{X \in \mathbb{R}^{m\times n} : X\mathbf{1}_n = a, X^T\mathbf{1}_n = b\},$$

and $\mathcal{C}_2 = O_{ij_{[k]}}$, used to equivalently rewrite (6) as:

$$\min_{X\in\mathcal{C}_1(a,b)} f(X) + \mathbb{I}_{\mathcal{C}_2}(Z), \quad (10)$$

$$\text{s.t. } X, Z \in \mathbb{R}^{m\times n}, \ X - Z = \mathbf{0}, \quad (11)$$

where the indicator function $\mathbb{I}_{\mathcal{C}}(X) = 0$ if $X \in \mathcal{C}$, and $\mathbb{I}_{\mathcal{C}}(X) = \infty$ otherwise. For $X \in \mathcal{C}_1(a,b)$, $Z \in \mathcal{C}_2$, 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:

$$\min_{X\in\mathcal{C}_1(a,b)} \left(\text{tr}\left(D^T X\right) + \tfrac{\rho}{2}\|X - Z_t + M_t\|_F^2\right)$$

$$\min_{Z\in\mathbb{R}^{m\times n}} \left(\mathbb{I}_{\mathcal{C}_2}(Z) + \tfrac{\rho}{2}\|X_{t+1} - Z + M_t\|_F^2\right) \quad (12)$$

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 $\mathcal{C}_1(a,b)$ of a point $Z_t - M_t - \rho^{-1}D$; similarly Z_{t+1} is the Euclidean projection onto $\mathcal{C}_2$ of $X_{t+1} + M_t$.

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

Require: Costs D, penalty ρ, initial M_0, Z_0.
1: **for** round $t \ge 1$ until stopping **do**
2: Update $X_{t+1} = \text{Proj}_{\mathcal{C}_1(a,b)}\left(Z_t - M_t - \rho^{-1}D\right)$
3: Update $Z_{t+1} = \text{Proj}_{\mathcal{C}_2}\left(X_{t+1} + M_t\right)$.
4: Update (scaled) dual variable $M_{t+1} = M_t + X_t - Z_t$.
5: Return X_t

We next derive the projections, $\text{Proj}_{\mathcal{C}_1(a,b)}(\cdot)$ and $\text{Proj}_{\mathcal{C}_2}(\cdot)$, required for Alg. 1. Prop. 3.1 gives the Euclidean projector for $\mathcal{C}_1(a,b)$.

Proposition 3.1 (Projection $\mathcal{C}_1(a,b)$)**.** *For $\mathcal{C}_1(a,b)$ consider the Euclidean projector $Proj_{\mathcal{C}_1(a,b)}(X)$. Let P_k be the projection of a vector onto its mean, i.e., $P_k = \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^T$. Define matrices $M := \mathbf{I}_m - \frac{m}{m+n}P_m$ and $N := \mathbf{I}_n - \frac{n}{m+n}P_n$, and matrices*

$$Y_1 := \frac{1}{n}\overbrace{[Ma,\cdots,Ma]^T}^{m \text{ copies (rows)}} + \frac{1}{m}\overbrace{[Nb,\cdots,Nb]}^{n \text{ copies (cols)}},$$

$$Y_2 := M(XP_n) + (P_m X)M. \quad (13)$$

Then the projection $\hat{X} = Proj_{\mathcal{C}_1(a,b)}(X)$ 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 $a = b = \mathbf{1}$. and X symmetric, which does not hold in the optimal transport setting.

Algorithm 2 ePAVA for $\mathcal{C}_2 = O_{ij_{[k]}}$ for $k \in [mn]$

Require: $X \in \mathbb{R}^{m\times n}$. Indices $ij_{[k]}$.
$\ell := 1, B := 1, \texttt{le}[1] := \texttt{ri}[1] := 1, \texttt{val}[1] := T(0)$.
for $\ell \leq k$ **do**
  $B := B+1$, $\ell := \ell+1$, $\texttt{le}[B] := \texttt{ri}[B] := \ell$, $\texttt{val}[B] := X_{i_\ell j_\ell}$.
  for $B \geq 2$ and $\texttt{val}[B] \leq \texttt{val}[B-1]$ **do**
    Let $q = \texttt{ri}[B]$.
    if $B = 2$ **then**
      Solve and store $\tilde{\eta} \geq 0$ satisfying $T(\tilde{\eta}) = \Delta_{2q} + \tilde{\eta}/(q-1)$. Set $\texttt{val}[B-1] := T(\tilde{\eta})$.
    else
      Set $\texttt{val}[B-1] := \Delta_{pq}$ for $p = \texttt{ri}[B-1]$.
    Set $\texttt{ri}[B-1] := \texttt{ri}[B]$. Decrement $B := B-1$.
Return $B, \tilde{\eta}$, le, ri, and val.

Next we address computing $\text{Proj}_{\mathcal{C}_2}(X)$ for $\mathcal{C}_2 = O_{ij_{[k]}}$ 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 $\text{Proj}_{\mathcal{C}_2}(X)$ for $\mathcal{C}_2 = O_{ij_{[mn]}}$. 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 $\mathcal{C}_2 = O_{ij_{[k]}}$ 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 $\texttt{le}[B]$ denote the left and $\texttt{ri}[B]$ the right boundary of B; $\texttt{val}[B]$ is the value of its projection. For any $\eta \geq 0$, let $r(\eta)$ denote the rank $(X_{i_1 j_1} - \eta)$ with respect to decreasing order $X_{(1)} \geq X_{(2)} \geq \cdots \geq X_{(mn-k+1)}$ over all variates in $V \cup \{i_1 j_1\}$, for V in (2). We require averages $\Delta_{pq} := \sum_{\ell=p}^{q} X_{i_\ell j_\ell}/(q-p+1)$ and a threshold function $T(\cdot) := (\tau(t(\eta), \eta))_+$, where $\tau(s, \eta)$ for any $0 \leq s < r(\eta)$ and *integer* $t(\eta) \geq 0$ are defined[3] as:

$$\tau(s,\eta) = \frac{1}{s+1}\left(X_{i_1 j_1} - \eta + \sum_{\ell=1}^{s} X_{(\ell)}\right), \quad (14)$$

$$t(\eta)+1 = \arg\min\left\{s \in [r] : \tau(s,\eta) > X_{(s)}\right\} \quad (15)$$

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

Proposition 3.2 (Projection $\mathcal{C}_2$)**.** *For $\mathcal{C}_2 = O_{ij_{[k]}}$ for any $i_\ell j_\ell \in [mn]$ where $\ell \in [k]$, consider the Euclidean projector $\text{Proj}_{\mathcal{C}_2}(X)$ for any $X \in \mathbb{R}^{m\times n}$. Let $T(\eta) := (\tau(t(\eta),\eta))_+$ for τ, 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 $B, \tilde{\eta}$, le, ri, and val. Furthermore, the projection $\hat{X} = \text{Proj}_{\mathcal{C}_2}(X)$ satisfies i) for $pq \in V$ we have $\hat{X}_{pq} = T(\tilde{\eta}) = \texttt{val}[1]$ if $\text{rank}(X_{pq}) \leq t(\tilde{\eta})$ or $\hat{X}_{pq} = (X_{pq})_+$ otherwise, and ii) for $\ell \in [k]$ we have $\hat{X}_{i_\ell j_\ell} = \texttt{val}[B']$ iff $\texttt{le}[B'] \leq \ell \leq \texttt{ri}[B']$ for some $B' \in [B]$.*

[3] The sum in (14) equals 0 when $s = 0$.

Convergence Analysis Standard ADMM convergence results hold as follows. As $t \to \infty$, the primal residue $\|X_t - Z_t\|_F \to 0$ and dual residue $\rho\|Z_{t-1} - Z_t\|_F \to 0$, 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[4] non-asymptotic result (see Beck, 2017, Thm 15.4). Consider the following.

As (10)-(11) are both linear, if $\mathcal{C}_1(a,b) \cap \mathcal{C}_2 \neq \emptyset$, then an optimal $Y^* = \arg\max_{Y\in\mathbb{R}^{m\times n}} d(Y)$ exists (Boyd & Vandenberghe, 2004) from solving the dual $d(Y) :=$

$$\min_{X\in\mathcal{C}_1(a,b), Z\in\mathcal{C}_2} \text{tr}\left(X^T(D+Y)\right) - \text{tr}\left(Y^T Z\right) \quad (16)$$

Recall Proposition 2.1 is a sufficient condition for $\mathcal{C}_1(a,b) \cap \mathcal{C}_2 \neq \emptyset$.

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 $\bar{X}_t = (t+1)^{-1}\sum_{\ell=1}^{t+1} X_\ell$ and similarly $\bar{Z}_t$. Assume $\mathcal{C}_1(a,b)\cap\mathcal{C}_2 \neq \emptyset$, and let f^* and Π^* denote the optimum value and solution respectively of (6), and $Y^* = \arg\max_{Y\in\mathbb{R}^{m\times n}} d(Y)$ for $d(Y)$ in (16). Then we have the convergence bounds*

$$f(\bar{X}_t) - f^* \leq \frac{\rho c_1}{2(t+1)}, \quad \left\|\bar{X}_t - \bar{Z}_t\right\|_F \leq \frac{\rho c_1}{c_2(t+1)}$$

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 $Z_0 = M_0 = \mathbf{0}$, achieves error $f(\bar{X}_t) - f^* \leq \delta$, in $\mathcal{O}(\|D\|_\infty/\delta)$ iterations.*

Proof. We seek to estimate ρc_1 that bounds the error $f(\bar{X}_t) - f^*$ in Thm. 3.3. First, consider the constant c_1. Setting $Z_0 = M_0 = \mathbf{0}$ 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 ρ, 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

[4] Use Thm 15.4 of Beck (2017) with $\mathbf{G}, \mathbf{Q}$ set to zero, and checking Assump 15.2 holds for $f(X)$ and $\mathcal{C}_1(a,b), \mathcal{C}_2$.

a constant c satisfying $c \geq \|Y^*(D')\|_F$ for any normalized [5] 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 ρc_1 to be in $\mathcal{O}(\|D\|_\infty)$. Thus we conclude the error is at most δ 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 $\mathcal{O}(\|D\|_\infty / \delta \cdot mn \log mn)$.

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

Proof. Line 2 costs $\mathcal{O}(mn)$ for $\mathcal{C}_1(a,b)$ and Line 4 costs $\mathcal{O}(k + (mn-k)\log(mn-k))$ for $\mathcal{C}_2$. The update in Line 2 requires only $\mathcal{O}(mn)$. Note that for $\text{Proj}_{\mathcal{C}_1(a,b)}(\cdot)$ the expressions (13) require only matrix-vector multiplications with $\mathcal{O}(mn)$ 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 $M = \mathbf{I}_m - \frac{m}{m+n} P_m$ and $P_k = \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^T$ for $k = m, n$, we get that $M(XP_n)$ can be expanded (ignoring scaling factors) into two terms $X\mathbf{1}_n\mathbf{1}_m^T$ and $\mathbf{1}_m^T(\mathbf{1}_m X \mathbf{1}_n)\mathbf{1}_n^T$. Hence the claim holds given that $\mathbf{1}_m, \mathbf{1}_n$ are vectors.

For $\mathcal{C}_2$, observe that computing $T(\eta) = \tau(t(\eta), \eta)$ multiple times in Alg. 2 (Line 7), requires a one-time sort of $mn - k + 1$ terms and Lines 5-10 incur an additional $\mathcal{O}(k)$ following arguments in Grotzinger & Witzgall (1984). □

Care should to be taken in comparing our complexity, $\mathcal{O}(\|D\|_\infty / \delta \cdot mn \log mn)$, to that of OT methods that use rounding approximations Altschuler et al. (2017) to ensure feasibility of each iterate. Such methods, which use KL-divergence, are $\mathcal{O}\left(n^2/\delta \cdot \|D\|_\infty\right)$ for $m = n$, (see Guminov et al., 2021; Jambulapati et al., 2019). For order constraints like other structured OT approaches (e.g. Scetbon et al., 2021), a rounding method for $U(a,b) \cap O_{ij_{[k]}}$ is not available. However, we have the bound on $\left\|\bar{X}_t - \bar{Z}_t\right\|_F$ in Thm. 3.3 to quantify the feasibility of the iterates X_t, Z_t from $\mathcal{C}_1(a,b) \cap \mathcal{C}_2$.

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 define the corresponding constraints. With the goal of generating plans from which a 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 $ij \in [mn]$ such that $\Pi_{i_k j_k} \geq \cdots \geq \Pi_{i_1 j_1} \geq \Pi_{ij} \geq \Pi_{pq}$ for $pq \in V$, see (2). This manner of introducing variates one-at-a-time, can be likened to variable selection and formalized by a tree structure. On the tree $\mathcal{T}$, each node $ij_{[k]}$ has children $i'j'_{[k+1]} = ij, i_1 j_1, i_2 j_2, \cdots, i_k j_k$ that share the same top-k constraints; the root node corresponds to unconstrained OT, and has children at level $k = 1$ corresponding to single order constraint variates $ij_{[1]} = ij$. 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 τ_1, τ_2 that lie between 0 and 1. For any given node $ij_{[k]}$, the parameters τ_1, τ_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 $\phi_{ij} = \Pi_{ij} / \min(a_i, b_j)$, to derive

$$\begin{aligned} \left(\phi^s{}_{ij}, \phi^r{}_{ij}, \phi^c{}_{ij}\right) &= \left(\phi_{ij}, \max_{\ell \in [n]:\, \ell \neq i} \phi_{i\ell}, \max_{\ell \in [m]:\, \ell \neq j} \phi_{\ell j}\right) \\ \Phi_{ij} &:= \min(\phi^r{}_{ij}, \phi^c{}_{ij}). \end{aligned} \tag{17}$$

Low saturation values in (17) imply uncertainty in the assignments Π. The thresholds τ_1, τ_2 are used to determine the set $\mathcal{I}(\Pi)$ that filters those variates ij with low self saturation (τ_1) and low neighborhood saturation (τ_2):

$$\mathcal{I}(\Pi) := \{ij \in [mn] : \phi^s{}_{ij} \leq \tau_1, \Phi_{ij} \leq \tau_2\}. \tag{18}$$

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 learning 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 beneath the thresholds τ_1, τ_2. We thus wish to learn the top-k_2 plans given by subtree, $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$, and iden-

[5] 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 αY^* optimizes the same for costs αD.

[6] For the choice $\rho = c_2$, it follows that the error $\left\|\bar{X}_t - \bar{Z}_t\right\|_F$ in Thm. 3.3 also drops linearly with t.

Algorithm 3 Learning subtree $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ of $\mathcal{T}(k_3, \tau_1, \tau_2)$ and top-k_2 candidate plans for linear costs $f(\Pi) = \text{tr}\left(D^T \Pi\right)$.

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 $\hat{\Pi}_1$ using (1). Init $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$.
2: Use $\hat{\Pi}_1$, τ_1, τ_2 in (17) and (18) to obtain $\mathcal{I}$ and Φ_{ij}. Init. $\mathcal{S} = \{(ij, \Phi_{ij}) : ij \in \mathcal{I}\}$. `count=0`.
3: **for** `count` $< k_1$ **do**
4: Pop $ij_{[k]}$ having smallest Φ in $\mathcal{S}$, for some k constraints. Compute $\mathcal{L}$ from right-hand side of (23).
5: **if** $\hat{\Pi}_{k_2}$ is not yet obtained or $\mathcal{L} > f(\hat{\Pi}_{k_2})$ **then**
6: Solve Algorithm 1 with order constraint $O_{ij_{[k]}}$ for new candidate $\hat{\Pi}$. Set `count += 1`.
7: Update top-k_2 candidates $\hat{\Pi}_1, \hat{\Pi}_2, \cdots, \hat{\Pi}_{k_2}$ and $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ using new candidate $\hat{\Pi}$.
8: **if** k equals k_3 **then**
9: Go to line 4.
10: **if** $\hat{\Pi}_{k_2}$ not yet obtained or $f(\hat{\Pi}) < f(\hat{\Pi}_{k_2})$ **then**
11: Use $\hat{\Pi}$, τ_1, τ_2 in (17) and (18) and obtain new variates $ij \in \mathcal{I}(\hat{\Pi})$ and $\{\Phi_{ij}\}_{ij \in \mathcal{I}(\hat{\Pi})}$.
12: **for** variate ij in $\mathcal{I}(\hat{\Pi})$ **do**
13: **if** $i \notin i_{[k]}$ and $j \notin j_{[k]}$ **then**
14: Push $(ij, i_1 j_1, \cdots, i_k j_k, \Phi_{ij})$ onto stack $\mathcal{S}$.
15: Return top k_2 candidates $\hat{\Pi}_1, \hat{\Pi}_2, \cdots, \hat{\Pi}_{k_2}$ and $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$.

tifying the nodes of $\mathcal{T}(k_3, \tau_1, \tau_2)$ that are in 1-to-1 correspondence with those top-k_2 plans. The tree $\mathcal{T}(k_3, \tau_1, \tau_2)$ is constructed dynamically, and avoids redundantly computing plans Π corresponding to variates $ij_{[k]}$ if the k_2-best candidate has cost lower than either: i) a known lower bound $\mathcal{L} \leq f(\Pi)$, or ii) the cost of $ij_{[k]}$ parent's plan. Finally, in the construction of $\mathcal{T}(k_3, \tau_1, \tau_2)$ we only consider variates where $i_{[k]}$ and $j_{[k]}$ do not repeat indices.

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 $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ are the top-k_2 amongst these k_1 plans, as the most uncertain variates 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:

$$\min_{\Pi \in U(a,b) \cap O_{ij_{[k]}}} f(\Pi) \geq \min_{\alpha \leq x \leq \beta} \left[G_{ij_{[k]}} \cdot x + g_{ij_{[k]}}(x, D) \right] \tag{19}$$

for some suitable α, β, where V in (2), and where $G_{ij_{[k]}} = \sum_{\ell \in [k]} D_{i_\ell j_\ell}$, and for coefficients $\varphi \in \mathbb{R}^{m \times n}$,

$$g_{ij_{[k]}}(x, \varphi) := \min_{\Pi \in U(a,b) \cap O_{ij_{[k]}} :\, \Pi_{i_\ell j_\ell} = x,\ \forall \ell \in [k]} \sum_{pq \in V} \Pi_{pq} \varphi_{pq}.$$

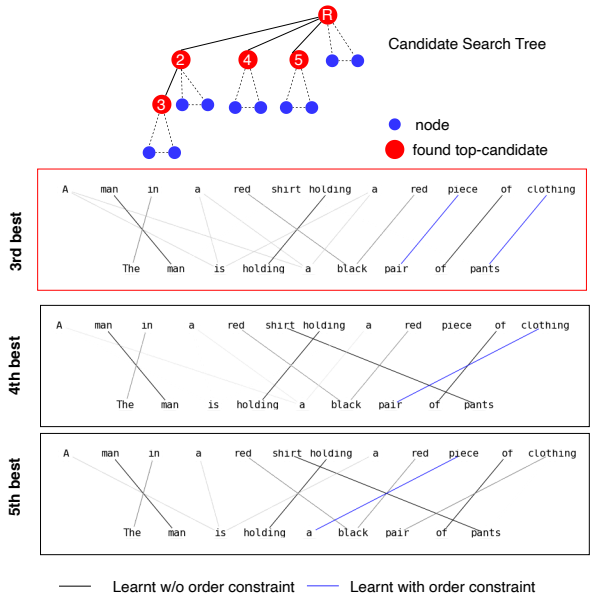

Figure 3: Top $k_2 = 5$ candidates for OT with multiple order constraints for the example in Fig. 2

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

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

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

$$\mu\left(u, \{\varphi\}_{i \in [n]}, \alpha\right) = \texttt{PACKING}\left(\{\varphi\}_{i \in [n]}, u, \alpha\right) \tag{21}$$
$$\nu\left(u, \{\varphi\}_{i \in [n-1]}, \alpha\right) = \texttt{PACKING}\left(\{\varphi\}_{i \in [n-1]}, u, \alpha - u\right).$$

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

$$g_{ij_{[k]}}(x, \varphi) \geq \begin{cases} L_{1ijk}(x, \varphi, a), & \alpha_1 \leq x \leq \beta_1 \\ L_{2ijk}(x, \varphi, b), & \alpha_2 \leq x \leq \beta_2 \end{cases} \tag{22}$$

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

$$\sum_{\ell \in [k]} \nu\left(x, \{\varphi_{i_\ell q}\}_{q \in [n] \setminus \{j_\ell\}}, a_{i_\ell}\right) + \sum_{p \notin i_{[k]}} \mu\left(x, \{\varphi_{pq}\}_{q \in [n]}, a_p\right)$$

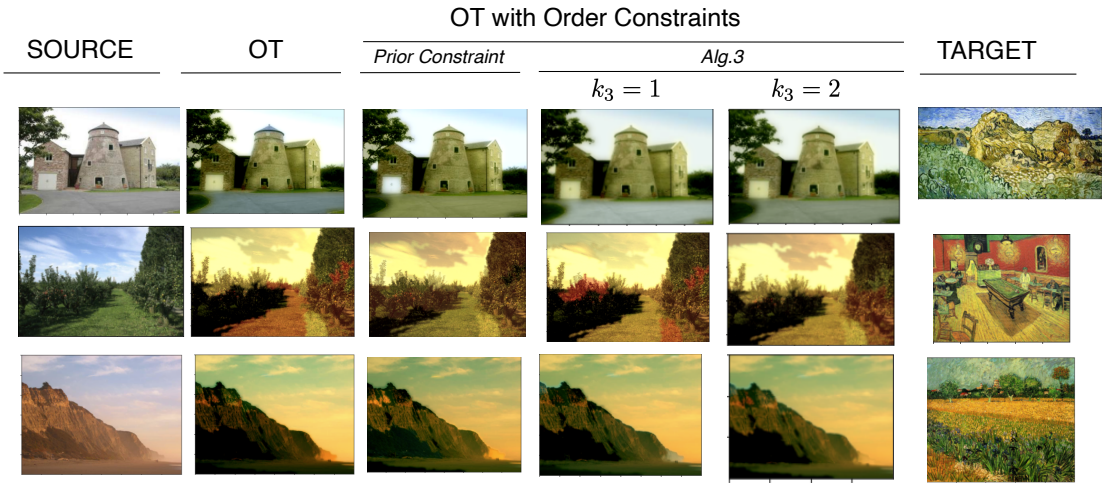

Figure 4: Color transfer candidates obtained using OT with OC (Alg. 3)

Table 1: Task and annotation scores on e-SNLI. `BestF1@n` is the best achievable annotation `F1` score over the set of $k_2 = \mathtt{n}$ best plans, measured by setting coefficients 0 or 1 after determining values under which there is no impact on the the task `F1` (75.1 $\pm$ .3). Standard OT achieves 64.5 $\pm$ .3 annotation `F1`.

Algorithm	BestF1@n		
	n= 2	n= 5	n= 10
Algorithm 3 (ours)	68.1 $\pm$ .2	71.2 $\pm$ .3	73.7 $\pm$ .2
Greedy version	67.9 $\pm$ .3	68.2 $\pm$ .3	68.2 $\pm$ .3

$$\sum_{\ell\in[k]} \nu\left(x, \{\varphi_{pj_\ell}\}_{p\in[m]\setminus\{i_\ell\}}, b_{j_\ell}\right) + \sum_{q\notin j_{[k]}} \mu\left(x, \{\varphi_{pq}\}_{p\in[m]}, b_q\right)$$

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

$$\begin{aligned} &\min_{\Pi\in U(a,b)\cap O_{ij_{[k]}}} f(\Pi) \\ \geq\ &\max\begin{pmatrix} \min_{\alpha_1\leq x\leq\beta_1} G_{ij_{[k]}}\cdot x + L_{1ijk}\left(x, D, a\right) \\ \min_{\alpha_2\leq x\leq\beta_2} G_{ij_{[k]}}\cdot x + L_{2ijk}\left(x, D, b\right) \end{pmatrix} \end{aligned} \tag{23}$$

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 (τ_1, τ_2) that constrain $\mathcal{T}(k_3, \tau_1, \tau_2)$ to be $(\tau_1, \tau_2) = (.5, .5)$. 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 $k_2 = \mathtt{n}$ plans are scored using the `BestF1@n` metric, which reports the score of the best plan in the learnt subtree $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$. 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

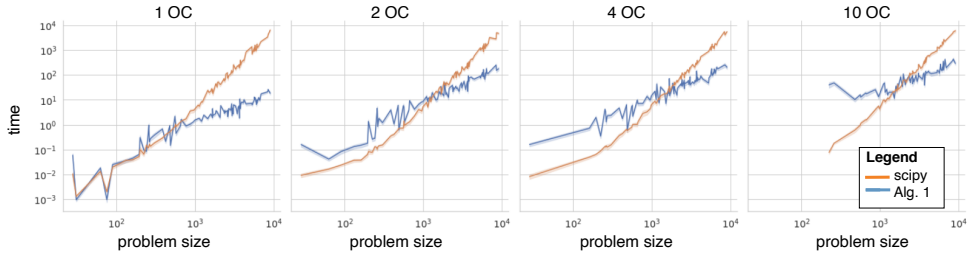

Figure 5: Compute time of Alg. 1 compared with `scipy.optimize`, for various number of constraints.

subtree $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ (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 $(\tau_1, \tau_2) = (.5, 1.)$ 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 $\pm.19$). We use penalty $\rho = 1.0$ after testing a range of ρ'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., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, jan 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

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., Fatras, 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):1–8, 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 brenier 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.

Proceedings of the AAAI Conference on Artificial Intelligence, 34:11990–11997, Apr. 2020. doi: 10.1609/aaai.v34i07.6875. URL https://ojs.aaai.org/index.php/AAAI/article/view/6875.

Solomon, J. Optimal transport on discrete domains. 2018. URL http://arxiv.org/abs/1801.07745.

Su, B. and Hua, G. Order-preserving wasserstein distance for sequence matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2906–2914, 2017. doi: 10.1109/CVPR.2017.310.

Swanson, K., Yu, L., and Lei, T. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5609–5626. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.496. URL https://www.aclweb.org/anthology/2020.acl-main.496.

Tan, W. R., Chan, C. S., Aguirre, H. E., and Tanaka, K. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3703–3707, Phoenix, AZ, USA, September 2016. IEEE. ISBN 978-1-4673-9961-6. doi: 10.1109/ICIP.2016.7533051. URL http://ieeexplore.ieee.org/document/7533051/.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wang, F., Li, P., and Konig, A. C. Learning a bi-stochastic data similarity matrix. In *2010 IEEE International Conference on Data Mining*, pp. 551–560, 2010. doi: 10.1109/ICDM.2010.141.

Wu, L., Yen, I. E.-H., Xu, K., Xu, F., Balakrishnan, A., Chen, P.-Y., Ravikumar, P., and Witbrock, M. J. Word mover's embedding: From Word2Vec to document embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4524–4534, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1482. URL https://www.aclweb.org/anthology/D18-1482.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970. ISSN: 1063-6919.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv:1506.03365 [cs]*, June 2016. URL http://arxiv.org/abs/1506.03365. arXiv: 1506.03365.

Yurochkin, M., Claici, S., Chien, E., Mirzazadeh, F., and Solomon, J. M. Hierarchical optimal transport for document representation. In Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F. d., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 1601–1611. Curran Associates, Inc., 2019.

Zhang, Y., Cheng, X., and Reeves, G. Convergence of gaussian-smoothed optimal transport distance with sub-gamma distributions and dependent samples. In *International Conference on Artificial Intelligence and Statistics*, pp. 2422–2430. PMLR, 2021.

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 explainablity 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 explainabilty 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 `F1`. The highest annotation `F1` score over $\texttt{n} = k_2$ 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 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 `F1`. Using the validation set, we pick the threshold in the region between 0.01 and 2 where the task `F1` 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 Π, 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 `F1` (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 $\mathcal{I}(\Pi)$, 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 Π. 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 (τ_1, τ_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 ϕ^s_{ij} and Φ_{ij} using (18). The polytope $U(a, b)$ constrains the points (ϕ^s_{ij}, Φ_{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 τ_1 and τ_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 `F1` score against the annotations. The top $k_2 = \texttt{n}$ plans are scored using the `BestF1@n` metric, which reports the score of the best plan in the learnt subtree $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$.

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 $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ is initialized. At Line 2 the single order constraint nodes are constructed and pushed to stack $\mathcal{S}$ along

[7] `https://github.com/asappresearch/rationale-alignment`

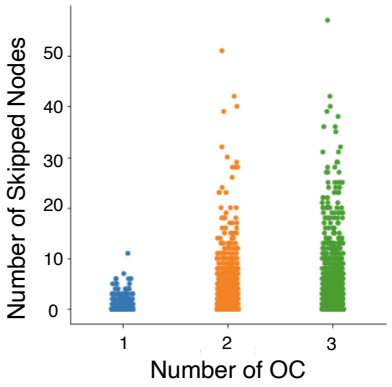

Figure 6: Effectiveness of lower bound (23) in terms of nodes skipped in $\mathcal{T}(k_3, \tau_1, \tau_2)$ for various depths $k_3 = 1, 2, 3$.

with saturations Φ_{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 $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ which now has two nodes. Then at Lines 11-14, new depth-2 nodes $ij_{[2]} = i'j'ij$ are computed from $\hat{\Pi}_2$ and pushed to $\mathcal{S}$ to be considered next in $\mathcal{T}(k_3, \tau_1, \tau_2)$ (e.g., "piece"-"pair" followed by "clothing"-"pants"). Lines 3-14 iterate to update $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ and the candidate set. Once $\widehat{\mathcal{T}}(k_1, k_2, k_3, \tau_1, \tau_2)$ grows to k_2 nodes, the lower bound test at Lines 5 and 11 prune redundant nodes of $\mathcal{T}(k_3, \tau_1, \tau_2)$. 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 $\widehat{\mathcal{T}}(20, 5, 1, \tau_1, \tau_2)$ and $\widehat{\mathcal{T}}(40, 10, 2, \tau_1, \tau_2)$, where in this set of experiments we chose $(\tau_1, \tau_2) = (0.5, 1.0)$. Here we have no labels to tune (τ_1, τ_2) as we did for e-SNLI. The neighbour saturation metric Φ_{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 `numpy`[8] 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 effectives of the lower bound (23), by showing the number of search tree nodes in $\mathcal{T}(k_3, \tau_1, \tau_2)$ 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 $\mathcal{T}(k_3, \tau_1, \tau_2)$ increases with the depth k_3.

[8] https://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 $\text{Proj}_{\mathcal{C}_2 \cap \mathcal{C}_2}(X) = \arg\min_{Y \in \mathcal{C}_2} \|Y - X\|_F^2$ involves $mn - k$ constraints $Y_{pq} \leq Y_{i_1 j_1}$ for $pq \in V$ where V in (2), followed by $mn - k + 1$ constraints $0 \leq Y_{pq}$ for $pq \in V \cup \{i_1 j_1\}$, and $k - 1$ constraints $Y_{i_\ell j_\ell} \leq Y_{i_{\ell+1} j_{\ell+1}}$ for $\ell \in [k-1]$. Write $\lambda_{pq}, \delta_{pq}$ and η_ℓ for their respective Lagrangian variables and the Karush-Kuhn-Tucker (KKT) conditions:

$$Y_{pq} = \begin{cases} X_{pq} - \lambda_{pq} + \delta_{pq} & \text{for all} \quad pq \in V \\ X_{i_1 j_1} - \eta_1 + \sum_{pq \in V} \lambda_{pq} + \delta_{i_1 j_1} & \text{for} \quad pq = i_1 j_1 \\ X_{i_\ell j_\ell} + \eta_{\ell-1} - \eta_\ell & \text{for all} \quad pq = i_\ell j_\ell \ \text{ where } \ 2 \leq \ell \leq k \end{cases} \tag{24}$$

$$Y_{pq} \leq Y_{i_1 j_1}, \quad \text{for all} \quad pq \in V, \quad \text{and} \quad Y_{pq} \geq 0 \quad \text{for all} \quad pq \in V \cup \{i_1 j_1\} \tag{25}$$

$$\lambda_{pq} \geq 0, \quad \text{for all} \quad pq \in V, \quad \text{and} \quad \delta_{pq} \geq 0 \quad \text{for all} \quad pq \in V \cup \{i_1 j_1\} \tag{26}$$

$$\lambda_{pq}(Y_{pq} - Y_{i_1 j_1}) = 0, \quad \text{for all} \quad pq \in V, \quad \text{and} \quad \delta_{pq} Y_{pq} = 0 \quad \text{for all} \quad pq \in V \cup \{i_1 j_1\} \tag{27}$$

$$Y_{i_\ell j_\ell} \leq Y_{i_{\ell+1} j_{\ell+1}}, \quad \text{for all} \quad \ell \in [k-1] \tag{28}$$

$$\eta_\ell \geq 0, \quad \text{for all} \quad \ell \in [k-1] \tag{29}$$

$$\eta_\ell (Y_{i_{\ell+1} j_{\ell+1}} - Y_{i_\ell j_\ell}) = 0, \quad \text{for all} \quad \ell \in [k-1] \tag{30}$$

Lemma B.1. *Given any index $ij = i_1 j_1$ 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*

$$\tau(t, \eta) \geq X_{(t+1)}, \ \text{ and if } \ t > 0 \ \text{ also } \ \tau(t, \eta) \leq X_{(t)} \leq X_{(t-1)} \leq \cdots \leq X_{(1)}. \tag{31}$$

Proof of Lemma B.1. Assume $\eta = 0$ and drop η from τ 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 $ij = i_1 j_1$ 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:*

$$X_{ij} < 0, \ \text{ and } \ \sum_{\ell=1}^{s-1} X_{(\ell)} < -X_{ij}, \tag{32}$$

Proof of Lemma B.2. Assume $\eta = 0$ and drop η from τ 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 $\tau(s) = (s+1)^{-1} \cdot (\sum_{\ell=1}^{s-1} X_{(\ell)} + X_{(s)} \cdot + X_{ij}) \leq X_{(s)}$, where the inequality follows from the minimality of t in (15). Re-arranging, we get $\sum_{\ell=1}^{s-1} X_{(\ell)} + X_{ij} \leq X_{(s)}$. On the other hand, we had assumed $X_{(s)} < 0$, so therefore we must conclude (32). □

Lemma B.3. *Given index $ij = i_1 j_1$ 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 $\text{rank}\,(X_{pq}) \leq t$ or $pq = ij$ 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* λ_{pq} *and* δ_{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 $ij = i_1 j_1$. 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

$$(\text{for all } pq \in V) \qquad Y_{pq} = \begin{cases} (\tau(t))_+ & \text{if rank}\,(X_{pq}) \leq t \\ (X_{pq})_+ & \text{otherwise} \end{cases} \tag{33}$$

$$Y_{ij} = (\tau(t))_+\,, \tag{34}$$

together with (31) and (32) in Lemmatta B.1 and B.2 respectively, establish λ_{pq} and δ_{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 $i_i j_1 = ij$, 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 $0 \leq Y_{pq} \overset{(33)}{=} X_{pq} \leq X_{(t+1)} \overset{(31)}{\leq} \tau(t) \overset{(34)}{=} Y_{ij}$, and whenever $X_{pq} < 0$ we have $Y_{pq} \overset{(33)}{=} (X_{pq})_+ = 0 \leq (\tau(t))_+ = Y_{ij}$. For all other $pq \in V$ where rank $(X_{pq}) \leq t$ we see that $Y_{pq} \overset{(33)}{=} (\tau(t))_+ \overset{(34)}{=} Y_{ij}$. Thus the first set of inequalities in (25) follow. The second set $Y_{pq} \geq 0$ for all $V \cup \{ij\}$ follow by definitions (33) and (34).

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

$$(\text{if rank}\,(X_{pq}) > t) \;\Rightarrow\; \lambda_{pq} = 0, \;\text{ and }\; \delta_{pq} = (X_{pq})_+ - X_{pq}, \tag{35}$$

$$(\text{if rank}\,(X_{pq}) \leq t) \;\Rightarrow\; \begin{cases} \lambda_{pq} = X_{pq} - (\tau(t))_+\,, & \delta_{pq} = 0, & \text{if } \tau(t) \geq 0 \text{ or } X_{pq} \geq 0, \\ \lambda_{pq} = 0, & \delta_{pq} = -X_{pq}, & \text{if } \tau(t) < 0 \text{ and } X_{pq} < 0 \end{cases} \tag{36}$$

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 trivally satisfied for λ_{pq}, and holds for δ_{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 *either* $\tau(t) \geq 0$ or $X_{pq} \geq 0$ holds, since $\tau(t) \overset{(31)}{\leq} X_{pq}$ 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 δ_{pq}, and for rank $(X_{pq}) \leq t$ this is seen given (33) putting $Y_{pq} = (\tau(t))_+$ and choice (36) for δ_{pq}.

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

$$\sum_{pq \in V} \lambda_{pq} \overset{(35)}{=} \sum_{pq \in V:\, \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq} - (\tau(t))_+) \quad = \quad \sum_{pq \in V:\, \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq}) - t \cdot (\tau(t))_+\,.$$

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:

$$\delta_{ij} = \begin{cases} (t+1)\tau(t) - (X_{ij} + \sum_{pq \in V:\, \text{rank}(X_{pq}) \leq t} X_{pq}) \overset{(14)}{=} (t+1)\tau(t) - (t+1)\tau(t) = 0 & \text{if } \tau(t) \geq 0 \\ -X_{ij} - \sum_{pq \in V:\, \text{rank}(X_{pq}) \leq t} (X_{pq} + \delta_{pq}) \overset{(36)}{=} -X_{ij} - \sum_{pq \in V:\, X_{pq} \geq 0} X_{pq} \overset{(32)}{>} 0, & \text{if } \tau(t) < 0 \end{cases}$$

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* $T : \mathbb{R}_+ \mapsto \mathbb{R}$ *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 η 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 $t(\eta)$ changes value, with the final inflection point occurring when $\tau(t(\eta),\eta)$ meets the horizontal axis. □

Lemma B.5 ((Grotzinger & Witzgall, 1984)). *For $p,q,r,s \in [k]$ satisfying $1 \le p \le q \le r \le s \le k$ where $r = q+1$, assume $\Delta_{pq} \le \Delta_{sq}$ for Δ in (37). Let ℓ satisfy $p \le \ell \le s$. Then for all $\ell \le q$ we have $\psi_{p\ell} - (\ell - p + 1)\Delta_{ps} \ge 0$, and for all $\ell > q$ we have we have $\psi_{r\ell} - (\ell - r + 1)\Delta_{rs} \ge 0$.*

Restatement of Proposition 3.2 For $\mathcal{C}_2 = O_{ij_{[k]}}$ for any $i_\ell j_\ell \in [mn]$ where $\ell \in [k]$, consider the Euclidean projector $\text{Proj}_{\mathcal{C}_2}(X)$ for any $X \in \mathbb{R}^{m\times n}$. Let $T(\eta) := (\tau(t(\eta),\eta))_+$ for τ, 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 $B, \tilde{\eta}$, `le`,`ri`, and `val`. Furthermore, the projection $\hat{X} = \text{Proj}_{\mathcal{C}_2}(x)$ satisfies i) for $pq \in V$ we have $\hat{X}_{pq} = T(\tilde{\eta}) = \texttt{val}[1]$ if rank $\left(\hat{X}_{pq}\right) \le t(\tilde{\eta})$ or $\hat{X}_{pq} = X_{pq}$ otherwise, and ii) for $\ell \in [k]$ we have $\hat{X}_{i_\ell j_\ell} = \texttt{val}[B']$ iff $\texttt{le}[B'] \le \ell \le \texttt{ri}[B']$ for some $B' \in [B]$.

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

$$\psi_{pq} = \sum_{\ell=p}^{q} X_{i_\ell j_\ell}, \quad \Delta_{pq} = \frac{\psi_{pq}}{q-p+1}, \quad \text{for } 1 \le p \le q \le k. \tag{37}$$

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 η_1 in the KKT conditions (24) and (29)–(30), that Alg. 2 terminates and when it does with $\hat{X}$, that $\hat{X} = Y$ satifies the KKT. The proof is recursive in η_1 and we initialize $\eta_1 = 0$. We use η_ℓ and η'_ℓ to denote the current, and next iterate, respectively.

We first invoke (24) to fix the relationship between solution and Lagrangian multipliers. For a given η_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 $pq \in V \cup \{i_1 j_1\}$ satisfying the first two lines of (24) and (25)–(27) of the KKT conditions. Likewise for η_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 $Y_{i_\ell j_\ell} = Y_{i_{\ell+1} j_{\ell+1}}$ for values of ℓ that satisfy $\texttt{le}[B'] \le \ell < \ell + 1 \le \texttt{ri}[B']$. On the other hand if ℓ is on the boundary $\ell = \texttt{ri}[B']$ and $\ell + 1 = \texttt{le}[B' + 1]$ then ePAVA results in $Y_{i_\ell j_\ell} \ne Y_{i_{\ell+1} j_{\ell+1}}$. Therefore using ψ_{pq} and Δ_{pq} from (37) and $\nu = \sum_{pq\in V} \lambda_{pq} + \delta_{i_1 j_1}$, we express $Y_{i_\ell j_\ell}$ and η_ℓ for all $\ell \in [k]$:

$$\eta_\ell = \psi_{1\ell} - \ell \cdot T(\eta_1) + \nu, \quad Y_{i_\ell j_\ell} = T(\eta_1), \quad \text{if } 1 \le \ell \le \texttt{ri}[1], \tag{38}$$

$$\eta_\ell = \psi_{p\ell} - (\ell - p + 1)\Delta_{pq}, \quad Y_{i_\ell j_\ell} = \Delta_{pq}, \quad \text{if } p = \texttt{le}[B'] \le \ell \le q = \texttt{ri}[B'] \tag{39}$$

where (39) is satisfied for all $1 < B' \le B$, and simply let $\eta_k = 0$ since this does not contradict (37). Thus it remains to show, as the iterates of η_1 update the values of $Y_{pq}, \lambda_{pq}, \delta_{pq}, \eta_{pq}$ 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 η_1) in Line 7. Specifically, let $q = \texttt{ri}[1]$ be the boundary of block 1. Suppose either a) $\eta_1 = 0$ and $q = 1$, or b) $\psi_{2q} - (q-1)\cdot T(\eta_1) + \eta_1 = 0$ is satisfied for some $\eta_1 > 0$ and $q > 1$. Let $r = q + 1$ and consider ψ_{2s} from (37) for some $s \ge r \ge 2$. We show below that in either case a) or b), that if $\Delta_{rs} \le T(\eta_1)$ holds, we must have that i) there exists a zero η'_1 satisfying $\psi_{2s} - (s-1)\cdot T(\eta'_1) + \eta'_1 = 0$, and ii) $T(\eta'_1) \le T(\eta_1)$ and $\eta'_1 \ge \eta_1$.

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 $T(\eta'_1) - \eta'_1/(s-1) - \Delta_{2s}$ has a zero (divide by $(s-1) \ge 1$ and use (37)). Indeed, the zero exists at some $\eta'_1 \ge 0$ since the non-increasing $T(\eta'_1)$, see Lemma B.4, and an increasing linear function $\eta'_1/(s-1) + \Delta_{2s}$, meets, as the latter starts at a point lower than the former as given by the assumption $\Delta_{2s} \le T(0)$. Furthermore, ii) holds since we showed $\eta'_1 \ge 0 = \eta_1$ and by monotonicity of $T(\cdot)$. In the other case put $\Delta_{2s} = (1-\beta)\Delta_{2q} + \beta\Delta_{rs}$ for $0 \le \beta = (s-q)/(s-1) \le 1$, and use the condition b) above to derive $T(\eta_1) - \Delta_{2s} - \eta_1/(s-1) = \beta \cdot (T(\eta_1) - \Delta_{rs}) \ge 0$ where the inequality follows since we assumed $\Delta_{rs} \le T(\eta_1)$. Then by monotonicity of $T(\cdot)$, see Lemma B.4, the zero exists and occurs at some point $\eta'_1 \ge \eta_1$ 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 $\texttt{ri}[1] = q = r - 1$. Let η'_ℓ for $\ell \in [s]$ equal (38) after coalescing, rewritten into $\eta'_\ell = \psi_{2\ell} - (\ell - 1)T(\eta'_1) + \eta'_1$, and similarly η_ℓ for $\ell \in [q]$ 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 $\eta_1 > 0$. ePAVA coalesces blocks 1 and 2 if $\texttt{val}[1] \leq \texttt{val}[2]$, 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 $\ell \leq q$ we have $\eta'_\ell - \eta_\ell = (\ell - 1)(T(\eta_1) - T(\eta'_1)) + \eta'_1 - \eta_1$, and by the inequality $\eta'_1 - \eta_1 \geq 0$, we conclude $\eta'_\ell - \eta_\ell \geq 0$. Next for $\ell > q$, we express the following for some $\alpha = 1 - (\ell - 1)/(s - 1) \geq 0$:

$$\begin{aligned} \eta'_\ell - \eta_\ell &\overset{(a)}{=} \eta'_1 + \psi_{2q} - (\ell - 1) \cdot T(\eta'_1) + (\ell - q)\Delta_{rs} \overset{(b)}{=} \alpha \cdot (\eta'_1 + \psi_{2q} - (q-1)\Delta_{rs}) \\ &\overset{(c)}{\geq} \alpha(q-1) \cdot (T(\eta_1) - \Delta_{rs}) \geq 0, \end{aligned}$$

where (a) follows from η'_ℓ in (38) and η_ℓ in (39), and (b) follows from expressing $T(\eta'_1) = [\psi_{2q} + (s-q) \cdot \Delta_{rs} + \eta'_1]/(s-1)$ and collecting terms into α, 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 $B' > 1$. ePAVA coalesces block B' and $B' + 1$ only if $\texttt{val}[B' - 1] \leq \texttt{val}[B']$, or equivalently, $\Delta_{pq} \leq \Delta_{rs}$. Consider any $p \leq \ell \leq s$. Eqn. (39) implies that the Lagrangians before and after coalescing are $\eta_\ell = \psi_{p\ell} - (\ell - p + 1)\Delta_{ps}$ and $\eta'_\ell = \psi_{r\ell} - (\ell - r + 1)\Delta_{rs}$, respectively. Supposing $\eta_\ell \geq 0$ and by these expressions for η'_ℓ, η_ℓ, we thus invoke Lemma B.5 to show $\eta'_\ell \geq 0$ for all $\ell \leq q$ 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 $s = \texttt{ri}[B']$ for any $B' \in [B]$ property $\eta_s = 0$; this proves complementary slackness (30) because (38)–(39) shows $Y_{i_\ell j_\ell}$ to differ only across boundaries. Let η_q and η'_s 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 $\eta_s = 0$ by definition (37). For (38) for $B' = 1$, rewrite (38) to get the form $\eta'_s = \psi_{2s} - (s-1) \cdot T(\eta'_1) + \eta'_1$ 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 ePAVA attempts to restore a non-increasing property of $Y_{i_\ell j_\ell}$ for all $\ell \in [k]$. This is only possible if the new values $Y_{i_\ell j_\ell}$ that result from coalescing does not increase beyond the values before coalescing. This is obvious for the case of $B > 2$ by definition (37). ePAVA eventually terminates satisfying (28), since there are finite number of coalescsing steps, and in the worst case arrives at the terminating state $B = 1$ and $\texttt{ri}[1] = k$ is arrived at, which satisfies (28). □

We now turn to the Euclidean projection $\text{Proj}_{\mathcal{C}_1(a,b)}(X)$ of a matrix $X \in \mathbb{R}^{m \times n}$ onto the row- and column-sum constraint set $\mathcal{C}_1(a, b)$ for measures a, b. Let $\otimes$ denote the matrix Kronecker (*i.e.*, tensor) product. Observe that the row-sums of any $X \in \mathbb{R}^{m \times n}$ can also be obtained via Kronecker equivalence $X\mathbf{1}_n = (\mathbf{I}_m \otimes \mathbf{1}_n^T)x$, where $x \in \mathbb{R}^{mn}$ is a length-mn vector formed from $X \in \mathbb{R}^{m \times n}$ by setting $x_{(k-1)n+\ell} = X_{k\ell}$. Also be the same equivalence we obtain column-sums of $X \in \mathbb{R}^{m \times n}$ $X^T\mathbf{1}_m = (\mathbf{1}_m \otimes \mathbf{I}_n)x$ from the same equivalent $x \in \mathbb{R}^{mn}$; in `numpy` terminology[9] one writes $x =$ `ravel` (X). We thus conclude that Euclidean projection $\hat{X} = \text{Proj}_{\mathcal{C}_1(a,b)}(X)$ for any $X \in \mathbb{R}^{m \times n}$, can be equivalently obtained by solving the following optimization over $\mathbb{R}^{mn}$:

$$\hat{x} = \arg\min_{y \in \mathbb{R}^{mn}} \frac{1}{2} \|y - x\|_2^2, \quad \text{s.t.} \quad (I_m \otimes 1_n^T)y = a, \quad (1_m^T \otimes I_n)y = b \tag{40}$$

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

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

[9] `https://numpy.org/doc/stable/reference/generated/numpy.ravel.html`

$\beta \in \mathbb{R}^n$ in a system of linear equations:

$$\begin{bmatrix} \mathbf{I}_{mn} & A^T \\ A & \end{bmatrix} \begin{bmatrix} y \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x \\ a \\ b \end{bmatrix} \tag{41}$$

where here A is an $(m+n) \times mn$ matrix with first m rows equal $I_m \otimes 1_n^T$ and last n rows equal $1_m^T \otimes I_n$. That is to compute $\hat{X} = \text{Proj}_{\mathcal{C}_1(a,b)}(X)$, one may simply solve (41) for y, given $x = \texttt{ravel}(X)$, 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 $(A^TA)^{-1}A^T$, and instead requires the *pseudo-inverse* of the matrix[10] $A \in \mathbb{R}^{(m+n)\times mn}$ in (41):

$$A^\dagger = \begin{bmatrix} \frac{1}{n}\left(\mathbf{I}_m \otimes \mathbf{1}_n - \frac{1}{m+n}\right) & \frac{1}{m}\left(\mathbf{1}_m \otimes \mathbf{I}_n - \frac{1}{m+n}\right) \end{bmatrix}.$$

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

Lemma B.6. *Assume $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ have same means, i.e., $a^T\mathbf{1}_n = b^T\mathbf{1}_n$. 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*

$$y = A^\dagger \begin{bmatrix} a \\ b \end{bmatrix} + \left(\mathbf{I}_{mn} - A^T\left(A^T\right)^\dagger\right)x \tag{42}$$

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 $\nu \in$ 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 $\nu \in$ null (A). Indeed for any $x \in \mathbb{R}^{mn}$, there exists some $\nu \in$ null (A) such that $x - \nu$ equals the projection of x onto span $\left(A^T\right)$, *i.e.*, there exists ν such that $x - \nu = A^T(A^T)^\dagger x$; to check write $\nu = \left(\mathbf{I}_{mn} - A^T(A^T)^\dagger\right)x$ and observe that ν is then the projection of any $x \in \mathbb{R}^{mn}$ onto $(\text{span}\left(A^T\right))^\perp$, and observe that $(\text{span}\left(A^T\right))^\perp = \text{null}\,(A)$, 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 $\nu = \left(\mathbf{I}_{mn} - A^T(A^T)^\dagger\right)x$ 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 $1^Ta = 1^Tb$ 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 $y_1 = \frac{1}{n}[M \otimes \mathbf{1}_n]a + \frac{1}{m}[\mathbf{1}_n \otimes N]b$ 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. □

[10] 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 $g_{ij_{[k]}}(x,\varphi)$ that appears in (19) in the main text. Prop. 4.1 derives this lower bound by decoupling the expression into (m 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 $\texttt{PACKING}\left(\{\varphi_k\}_{k\in[n]},u,\alpha\right)$, see (20) in the main text, where u and α represent the per item x_k capacity and total budget, respectively. We first present Lemmatas B.7–B.8 that show the quantities $L_{1ijk}(x,D,a)$ and $L_{2ijk}(x,D,b)$ (in the lower bound (23), main text) are convex and piecewise-linear. That is the lower bound in Proposition 4.1 can be efficiently evaluated by bisection; it only requires an upfront sorting complexity of $\mathcal{O}(n\log n)$ when parallelized.

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

$$\texttt{PACKING}\left(\{\varphi\}_{k\in[n]},u,\alpha\right)=u\left[\sum_{k=1}^{\ell}\left(\varphi_{(k)}-\varphi_{(\ell+1)}\right)\right]+\alpha\varphi_{(\ell+1)} \tag{43}$$

where $\ell=\lfloor\alpha/u\rfloor<n$, *and* $\varphi_{(k)}$ *is the* k*-th coefficient in increasing order* $\varphi_{(1)}\le\varphi_{(2)}\le\cdots\le\varphi_{(n)}$. *For fixed* α, *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:

$$\texttt{PACKING}\left(\{\phi_k\}_{k\in[n]},u,\alpha\right)=\sum_{k=1}^{\ell}\varphi_{(k)}u+(\alpha-\ell\cdot u)\varphi_{(\ell+1)} \tag{44}$$

where $\ell=\lfloor\alpha/u\rfloor<n$, and clearly $\alpha/n\le u\le\alpha$ is required to have a 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 $0\le\alpha\le 1$, and consider u in the interval $(\alpha/(i+1),\alpha/i]$. For our choice of i we conclude the sequence of inequalities $i+1=\left\lfloor\frac{\alpha}{\alpha/(i+1)}\right\rfloor>\left\lfloor\frac{\alpha}{u}\right\rfloor=\ell\ge\left\lfloor\frac{\alpha}{\alpha/i}\right\rfloor=i$ and so we conclude ℓ in (44) will equal $\ell=i$ 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 $\ell=i+1$):

$$\texttt{PACKING}\left(\{\phi_k\}_{k\in[n]},\frac{\alpha}{i+1},\alpha\right)=\lim_{\frac{\alpha}{i+1}\leftarrow u}\texttt{PACKING}\left(\{\phi_k\}_{k\in[n]},u,\alpha\right)=\frac{\alpha}{i+1}\sum_{k=1}^{i+1}\varphi_{(k)}.$$

From (44) it the gradient is non-positive everywhere due to the non-positivity of the summands $\varphi_{(k)}-\varphi_{(i+1)}\le 0$ since $k\le\ell$, 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 $\ell=i+1$):

$$\left[\sum_{k=1}^{i-1}\left(\varphi_{(k)}-\varphi_{(i)}\right)\right]-\left[\sum_{k=1}^{i}\left(\varphi_{(k)}-\varphi_{(i+1)}\right)\right]=i\cdot\left(\varphi_{(i+1)}-\varphi_{(i)}\right)\ge 0.$$

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

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

$$\texttt{PACKING}\left(\{\varphi_k\}_{k\in[n-1]},u,\alpha-u\right)=u\left[-\varphi_{(\ell)}+\sum_{k=1}^{\ell-1}\left(\varphi_{(k)}-\varphi_{(\ell)}\right)\right]+\alpha\varphi_{(\ell)} \tag{45}$$

where $\ell=\lfloor\alpha/u\rfloor$ *and* $\varphi_{(k)}$ *is the* k*-th coefficient in increasing order* $\varphi_{(1)}\le\varphi_{(2)}\le\cdots\le\varphi_{(n)}$. *For a for a fixed* α, *it is convex piecewise-linear and non-increasing in* u *with the inflection points* $u=\alpha/i$ *for* $i=n-1,n-2,\cdots,1$.

Proof. The proof is identical to that of Lemma B.7; we outline the key differences. The expression (45) is derived similarly as (43) by the same greedy strategy

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

To show the monotonic non-increasing behavior and convexity proceed in the similar manner. The $n-1$ inflection points for $i = n-1, n-2, \cdots 1$ are the same, thus so are the piecewise intervals; evaluating at the left-limit $\alpha/(i+1) \leftarrow u$ of the interval $(\alpha/(i+1), \alpha/i]$ equals $\frac{\alpha}{i+1}\sum_{k=1}^{i}\varphi_{(k)}$, as does the rightmost point $u = \alpha/(i+1)$ of the interval $(\alpha/(i+2), \alpha/(i+1)]$. The gradient is indeed non-decreasing, again as seen by subtracting the gradient in the i-th $(\alpha/(i+1), \alpha/i]$ from that of $(i-1)$-th interval which we obtain $i\cdot\left(\varphi_{(i)} - \varphi_{(i-1)}\right) \geq 0$ for any choice of $i < n$. □

Complexity of computing lower bounds in Proposition 4.1: The lower bound (see (23), main text) suggests to evaluate $L_{1ijk}(x,\varphi,a)$ and $L_{2ijk}(x,\varphi,b)$ at multiple values of $\alpha \leq x \leq \beta$, or equivalently, evaluate μ and ν (see (22), main text) over the same. But μ and ν are defined using $\texttt{PACKING}\left(\{\varphi\}_{i\in[n]}, x, a_k\right)$ and $\texttt{PACKING}\left(\{\varphi\}_{i\in[n-1]}, x, a_i - x\right)$, and by Lemmata B.7–B.8 they are piecewise-linear x with gradients and coefficients determined upfront by an $\mathcal{O}\left(n\log(n)\right)$ sort on the coefficients φ_k. In other words, the cost of $\mathcal{O}\left(n\log(n)\right)$ is only one-time, and once paid, the function (43) can be evaluated for multiple points of x in $\mathcal{O}\left(1\right)$ time. Finally, each μ,ν summand in the expressions for $L_{1ijk}(x,\varphi,a)$ and $L_{2ijk}(x,\varphi,b)$ can be computed independently in parallel.

Restatement of Proposition 4.1 Let $\alpha_1 = \max_{i=1}^{m}\frac{a_i}{n}, \beta_1 = \max_{i=1}^{m} a_i,$, and $\alpha_2 = \max_{j=1}^{n}\frac{b_j}{m}, \beta_2 = \max_{j=1}^{n} b_j$. Then for any $ij_{[k]}$ set that defines the order constrained OT (see (6) in main text) where $i_{[k]} = i_1, i_2, \cdots, i_k$ (and $j_{[k]}$) do not repeat row (or column) indices, the minimum $g_{ij_{[k]}}(x,\varphi)$ (see (19) in main text) is lower-bounded by

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

where $L_{1ijk}\left(x,\varphi,a\right)$ and $L_{2ijk}\left(x,\varphi,b\right)$ resp. equal $\sum_{\ell\in[k]}\nu\left(x, \{\varphi_{i_\ell q}\}_{q\in[n]\setminus\{j_\ell\}}, a_{i_\ell}\right) + \sum_{p\notin i_{[k]}}\mu\left(x, \{\varphi_{pq}\}_{q\in[n]}, a_p\right)$ and $\sum_{\ell\in[k]}\nu\left(x, \{\varphi_{pj_\ell}\}_{p\in[m]\setminus\{i_\ell\}}, b_{j_\ell}\right) + \sum_{q\notin j_{[k]}}\mu\left(x, \{\varphi_{pq}\}_{p\in[m]}, b_q\right)$.

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

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

$$\begin{aligned} &\left\{\Pi\in\mathbb{R}^{m\times n} : \Pi\in U(a,b),\ \Pi\in O_{ij_{[k]}},\ \Pi_{i_1j_1} = \cdots = \Pi_{i_kj_k} = x\right\} \\ \subseteq\ &\bigcap_{p\in[m]}\left\{\Pi\in\mathbb{R}_+^{m\times n} : \sum_{q\in[n]}\Pi_{pq} = a_q,\ \Pi_{i_1j_1} = x, \cdots, \Pi_{i_kj_k} = x,\ \max_{q\in[n]}\Pi_{pq} \leq x\right\}. \end{aligned}$$

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

$$\begin{array}{llll} \min\sum_{q\in[n]}\varphi_{pq}\Pi_{pq} & \text{s.t.} & \sum_{q\in[n]}\Pi_{pq} = a_p,\ \text{ and } 0\leq\Pi_{pq}\leq x, & \text{for } p\notin i_{[k]} \\ \min\sum_{q\in[n]\setminus\{j_\ell\}}\varphi_{i_\ell q}\Pi_{i_\ell q} & \text{s.t.} & \sum_{q\in[n]\setminus\{j_\ell\}}\Pi_{i_\ell q} = a_{i_\ell} - x,\ \text{ and } 0\leq\Pi_{i_\ell q}\leq x, & \text{for } \ell\in[k] \end{array}$$

where the sum of the m optimal costs of the m minimizations, lower bound the quantity $g_{ij_{[k]}}(x,\varphi)$. For any $p\notin i_{[k]}$, the p-th minimization has optimal cost $\mu\left(x, \{\varphi_{pq}\}_{q\in[n]}, a_p\right) = \texttt{PACKING}\left(\{\varphi_{pq}\}_{q\in[n]}, x, a_p\right)$, and similarly for $\ell\in[k]$, the i_ℓ-th row has optimal cost $\nu\left(x, \{\varphi_{pq}\}_{q\in[n]\setminus\{j_\ell\}}, a_{i_\ell}\right) = \texttt{PACKING}\left(\{\varphi_{pq}\}_{q\in[n]\setminus\{j_\ell\}}, x, a_{i_\ell} - x\right)$, 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 $\alpha_1 = \max_{p\in[m]} a_p/n$ and $\beta_1 = \max_{p\in[n]} a_p$.

□