A HYPERGRADIENT APPROACH TO ROBUST REGRESSION WITHOUT CORRESPONDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider a regression problem, where the correspondence between input data and output data is not accessible. Such shuffled data is commonly observed. For example, in flow cytometry, the measuring instruments are unable to preserve the correspondence between the samples and the measurements. Existing works for this problem generally focus on small number of data and/or linear case. In this work, we propose a novel computational framework (ROBOT) for the shuffled regression problem that can handle large-scale data and complex models more effectively than previous methods. To facilitate this, we consider the interaction between the regression parameter and the data correspondence parameter by building an end-to-end training framework using differentiable programming techniques. As an extension, ROBOT can also be applied to the robust correspondence setting, where the input data and the output data are not exactly aligned. Numerical experiments show that ROBOT achieves better performance than existing methods in both linear and nonlinear regression tasks, including real-world applications such as flow cytometry and multi-object tracking.

1 INTRODUCTION

Regression is widely used in various machine learning applications to infer the the relationship between an explanatory random variable (or the input) $x \in \mathbb{R}^d$ and a response random variable (or the output) $y \in \mathbb{R}^o$ (Stanton, 2001). In the standard setting, regression is used on labeled datasets that contain paired samples $\{x^{(i)}, y^{(i)}\}_{i=1}^n$, where $x^{(i)}$, $y^{(i)}$ are realizations of x, y, respectively.

Unfortunately, the input-output correspondence is not accessible in some applications. For example, in certain physical experiments, e.g., flow cytometry, the measuring instruments are unable to preserve the correspondence between the samples and the measurements (Abid & Zou, 2018). Flow cytometry is a technique that measures properties of cells, e.g., affinity to a particular target. In this process, cells are suspended in a fluid and injected into the flow cytometer, where measurements are taken using the scattering of a laser. However, the instruments are unable to differentiate the cells passing through the laser, such that the correspondence between the cell proprieties (i.e., the measurements) and the cells is unknown. Another example is multi-object tracking, where we need to infer the motion of objects given consecutive frames in a video. This requires us to find the correspondence between the objects in the current frame and those in the next frame.

For both of the above examples, there are two datasets $\mathcal{D}_1 = \{x_1^{(i)}, y^{(i)}\}_{i=1}^n$ and $\mathcal{D}_2 = \{x_2^{(j)}\}_{j=1}^n$, where $x_1^{(i)} \in \mathbb{R}^{d_1}$ and $x_2^{(i)} \in \mathbb{R}^{d_2}$. We are interested in inferring a permutation π such that $(x_1^{(i)}, x_2^{(\pi(i))})$ and $y^{(i)}$ are corresponding. Here $\{\pi(i)\}_{i=1}^n$ is an unknown permutation of $\{1, \cdots, n\}$. This problem is known as *shuffled regression*, *unlabeled sensing*, *homomorphic sensing*, and *regression with an unknown permutation* (Unnikrishnan et al., 2018). For the rest of this paper, we refer to the problem as *Regression WithOut Correspondence* (RWOC), and we postulate a model

$$y^{(i)} = f\left([x_1^{(i)}, x_2^{(\pi(i))}]; w\right) + \varepsilon^{(i)},$$

where f is an unknown function parameterized by w, and $\varepsilon^{(i)}$ signifies random noise.

The objective of RWOC is to minimize the sum of squared residuals with respect to the regression model parameter w and the permutation $\pi(\cdot)$, namely we solve

$$\min_{w \in \mathbb{R}^d, \pi} \mathcal{L}(w, \pi(\cdot)) = \sum_{i=1}^n \left\| y^{(i)} - f\left([x_1^{(i)}, x_2^{(\pi(i))}]; w\right) \right\|_2^2. \tag{1}$$

Existing works on RWOC mostly focus on the theoretical feasibility of the recovery of w and $\pi(\cdot)$. In contrast, the development of practical algorithms is quite limited in the following three aspects:

• Most of the works are only applicable to linear regression models.

• Existing algorithms fail to consider the interaction between the two parameters w and $\pi(\cdot)$ (see Section 2). For example, Abid et al. (2017) considers the two parameters separately and applies alternating minimization with respect to them. Due to the strong interaction between w and $\pi(\cdot)$, the optimization landscape is usually ill-conditioned. Therefore, these algorithms are not effective, e.g., they get stuck in local optima easily.

• Most of the works only consider the case where there exists exact one-to-one correspondence between $\mathcal{D}_1$ and $\mathcal{D}_2$. In many scenarios, however, these data are not necessarily well aligned. For example, consider $\mathcal{D}_1$ and $\mathcal{D}_2$ collected from two separate databases, where the users overlap, but are not identical. As a result, there exists only partial one-to-one correspondence. A similar situation also happens to multiple-object tracking: Some objects may leave the scene in one frame and new objects may enter the scene in subsequent frames. Therefore, not all objects in different frames can be perfectly matched. The RWOC problem with partial correspondence is known as robust-RWOC, or rRWOC (Varol & Nejatbakhsh, 2019), and is a problem much less studied.

In this work, we propose a computational framework, ROBOT (Regression withOut correspondence using Bilevel OptimizaTion) to address these issues. Our framework exploits the interaction between the data correspondence (i.e., $\pi(\cdot)$) and the regression model (i.e., w) to alleviate the suboptimality issue. Specifically, we consider the function $\pi^*(\cdot, w) = \arg\min_\pi \mathcal{L}(w, \pi(\cdot))$, and smooth it with differentiable programming techniques (Duchi et al., 2008; Luise et al., 2018). We then substitute π^* into (1) to solve for w. Our model can be optimized end-to-end using efficient first order optimization algorithms, which can easily handle large number of data points and complex models (e.g., massive neural networks). We further derive explicit expressions for the model updates to reduce the time and memory costs of automatic differentiation.

Our framework can be naturally extended to the robust case, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are not necessarily exactly aligned, i.e., some data points in $\mathcal{D}_1$ may not correspond to any data point in $\mathcal{D}_2$. Specifically, we relax the constraints on the permutation $\pi(\cdot)$ (Liero et al., 2018) to automatically match related data points and ignore the unrelated ones.

We conduct numerical experiments to evaluate our proposed methods. For well-aligned RWOC, we test ROBOT on synthetic regression datasets and a gated flow cytometry dataset. In all of these datasets, our model outperforms baseline methods by notable margins. We also formulate the vision-based multiple-object tracking task as a rRWOC problem, and the robust version of ROBOT achieves significant improvement over the baselines.

Related works. Existing works have established theoretical guarantees on the recovery of w and $\pi(\cdot)$ for linear models (Pananjady et al., 2016; 2017b; Abid et al., 2017; Elhami et al., 2017; Hsu et al., 2017; Unnikrishnan et al., 2018; Tsakiris & Peng, 2019). On the computational side, however, much less have been achieved. First, as mentioned, most of the works only considers linear regression models. Moreover, the algorithms are of very high computational complexity, and can only handle small number of data points in low dimensions (Elhami et al., 2017; Pananjady et al., 2017a; Tsakiris et al., 2018; Peng & Tsakiris, 2020). Other works improve the computational efficiency by solving variants of RWOC with additional constraints, e.g., Haghighatshoar & Caire (2017); Rigollet & Weed (2018) assume an isotonic function (note that such an assumption may not hold in practice), and Shi et al. (2018); Slawski & Ben-David (2019); Slawski et al. (2019a;b); Varol & Nejatbakhsh (2019) assume only a small fraction of the correspondence is missing.

Notations. Denote $\|\cdot\|_2$ the ℓ_2 norm of vectors, $\langle \cdot, \cdot \rangle$ the inner product of matrices, i.e., $\langle A, B \rangle = \sum_{i,j} A_{ij} B_{ij}$ for matrices A and B. $a_{i:j}$ are the entries from index i to index j of vector a. Let $\mathbf{1}_n$ denotes an n-dimensional vector of all ones. Denote $\frac{d(\cdot)}{d(\cdot)}$ the gradient of scalars, and $\nabla_{(\cdot)}(\cdot)$ the Jacobian of tensors. We denote $[v_1, v_2]$ the concatenation of two vectors v_1 and v_2. $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2.

2 ROBOT: A HYPERGRADIENT APPROACH FOR RWOC

A straightforward algorithm for solving (1) is to apply Alternating Minimization (AM) with respect to the two parameters w and $\pi(\cdot)$. Concretely, in iteration k, we first solve for the correspondence $\pi^*_{(k)} = \arg\min_\pi \mathcal{L}(w_{(k-1)}, \pi(\cdot))$ given $w_{(k-1)}$. And then we update $w_{(k)}$ with $\pi^*_{(k)}$ using first order algorithms, such as gradient descent. Note that for RWOC, w and $\pi(\cdot)$ can have strong interaction: When w changes, π^* can change significantly. In the AM algorithm, $\pi^*_{(k)}$ is optimized based on the suboptimal $w_{(k-1)}$, such that it can be far away from the optimal correspondence. Consequently, $w_{(k)}$ is updated by the gradient of this inaccurate $\pi^*_{(k)}$. The error caused by the "inaccurate" gradient can lead to significant suboptimality. In other words, the optimization landscape of w is ill-conditioned due to the substantial interaction between w and π, leading to non-effective updates.

We introduce our computational framework, ROBOT. This framework alleviates the "inaccurate" gradient issue by considering the interaction between the two parameters w and $\pi(\cdot)$ with an end-to-end training framework.

2.1 A DIFFERENTIABLE FRAMEWORK WITH INTERACTION BETWEEN w AND S

To employ an end-to-end optimization algorithm, our first step is to remove the discreteness of the feasible set of $\pi(\cdot)$. By simple manipulation, (1) can be rewritten in the following form,

$$\min_w \min_{S\in\mathbb{R}^{n\times n}} \langle C(w), S\rangle, \quad \text{s.t. } S\in\mathcal{P}, \quad \text{with } C_{ij}(w) = \left\| y^{(i)} - f\left([x_1^{(i)}, x_2^{(j)}]; w\right)\right\|_2^2, \tag{2}$$

where $\mathcal{P}$ denotes the set of all $n\times n$ permutation matrices, and $C(w)\in\mathbb{R}^{n\times n}$ is the cost matrix.

Note that the inner minimization problem of (2) is

$$\min_{S\in\mathbb{R}^{n\times n}} \langle C(w), S\rangle, \quad \text{s.t. } S\in\mathcal{P}.$$

The following theorem suggests that relaxing the constraints on S to be a convex set does not affect the optimal solutions.

Theorem 1. Denote $\Pi(a,b) = \{S\in\mathbb{R}^{n\times m} : S\mathbf{1}_m = a, S^\top\mathbf{1}_n = b, S_{ij}\geq 0\}$ for any $a\in\mathbb{R}^n$ and $b\in\mathbb{R}^m$. Then the optimal solutions of the following problem lie in $\mathcal{P}$.

$$\min_{S\in\mathbb{R}^{n\times n}} \langle C(w), S\rangle, \quad \text{s.t. } S\in\Pi(\mathbf{1}_n,\mathbf{1}_n).$$

This is a direct corollary of the Birkhoff–von Neumann theorem (Birkhoff, 1946; Von Neumann, 1953). We provide more details in Appendix A. By relaxing $\mathcal{P}$ to the Birkhoff polytope $\Pi(\mathbf{1}_n,\mathbf{1}_n)$ (Ziegler, 2012), we remove the discreteness of the feasible set of S[1].

Recall that the suboptimality of AM is because w is updated by "inaccurate" gradients. Our second step is to make the updates "accurate". Concretely, we rewrite (2) as a bilevel optimization problem:

$$\min_w \mathcal{L}(w) = \langle C(w), S^*(w)\rangle, \quad \text{s.t. } S^*(w) = \arg\min_{S\in\Pi(\mathbf{1}_n,\mathbf{1}_n)} \langle C(w), S\rangle. \tag{3}$$

Note that here we relax the constraint $S\in\mathcal{P}$ to $S\in\Pi(\mathbf{1}_n,\mathbf{1}_n)$ using Theorem 1. Our bilevel formulation splits the optimization problem into two parts: First we solve S^* as a function of w, and then we substitute $S^*(w)$ into the original problem and solve for w using first order optimization algorithms. In particular, a first-order method updates w by the hypergradient (Grazzi et al., 2020),

$$\nabla_w\mathcal{L}(w) = \frac{\partial\mathcal{L}(w)}{\partial C(w)}\nabla_w C(w) + \frac{\partial\mathcal{L}(w)}{\partial S^*(w)}\nabla_w S^*(w).$$

Note that a conventional AM only considers the first half of $\nabla_w\mathcal{L}(w)$. Here, we can naturally model the interaction between S and w: The update of w also considers $\partial\mathcal{L}(w)/\partial S^*(w)$, which represents how "inaccurate" $S^*(w)$ is. This interaction term enables a more effective algorithm[2].

However, $S^*(w)$ is not suitable for applying gradient based optimization algorithms, because the Jacobian matrix of $S^*(w)$ with respect to w is either zero or does not exist. This is because $S^*(w)$ is the optimal solution of a linear optimization problem, therefore in most cases $S^*(w)$ is on a vertex of $\Pi(\mathbf{1}_n,\mathbf{1}_n)$. And when w changes, $S^*(w)$ either stays unchanged, or jumps to another vertex.

[1] This is a common practice in integer programming (Marcus & Ree, 1959).

[2] The bilevel formulation can also be viewed as a Stackelberg game, see Appendix B.2 for more discussion.

To resolve this issue, our third step is to smooth $S^*(w)$ by adding an entropy regularization term to the lower level optimization in (3),

$$\begin{aligned} &\min_w \mathcal{L}_\epsilon(w) = \langle C(w), S_\epsilon^*(w)\rangle, \\ &\text{s.t. } S_\epsilon^*(w) = \arg\min_{S\in\Pi(\mathbf{1}_n,\mathbf{1}_n)} \langle C(w), S\rangle + \epsilon H(S), \end{aligned} \tag{4}$$

where $H(S) = \langle \log S, S\rangle$ is the entropy of S. In this way, $S_\epsilon^*(w)$ is always differentiable with respect to w, as indicated by the following theorem.

Theorem 2. For any $\epsilon > 0$, $S_\epsilon^*(w)$ is differentiable, as long as the cost $C(w)$ is differentiable with respect to w. As a result, the objective $\mathcal{L}_\epsilon(w) = \langle C(w), S_\epsilon^*(w)\rangle$ is also differentiable.

The proof is straightforward and analogous to a result in Xie et al. (2020). We provide more details in Appendix C. The intuition is that because of the entropy regularization, $S_\epsilon^*(w)$ is always in the interior of $\Pi(\mathbf{1}_n, \mathbf{1}_n)$, so that $S_\epsilon^*(w)$ changes smoothly with respect to w. With the entropy regularization, we can now optimize w using gradient based algorithms.

An Illustrative Example: Bilevel Optimization for Quadratic Minimization. We use an example to show why the bilevel optimization formulation in (3) can have a better optimization landscape. Consider a quadratic function $F(a_1, a_2) = a^\top P a + b^\top a$, where $a_1 \in \mathbb{R}^{d_1}$, $a_2 \in \mathbb{R}^{d_2}$, $a = [a_1, a_2]$, $P \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$, $b \in \mathbb{R}^{d_1+d_2}$. To further simplify the discussion, we assume $P = \rho \mathbf{1}_{d_1+d_2}\mathbf{1}_{d_1+d_2}^\top + (1-\rho) I_{d_1+d_2}$, where $I_{d_1+d_2}$ is the identity matrix, and ρ is a constant.

Proposition 1. For the optimization problem $\min_{a_1,a_2} F(a_1, a_2)$, the condition number of the bilevel optimization formulation (ROBOT) is $1 + \frac{d_1\rho}{d_2\rho-\rho+1}$, while that of AM is $1 + \frac{d_1\rho}{1-\rho}$.

The proof can be found in Appendix B.1. As demonstrated, ROBOT is much better-conditioned than AM in high dimensional cases, which is consistent with our intuition earlier. Moreover, since the iteration complexity of gradient descent is proportional to the condition number (Bottou et al., 2018), ROBOT needs fewer iterations to converge than AM.

2.2 Regression Model Update with Explicit Hypergradient

In this section, we present update rules for the model.

First we solve the lower level optimization problem in (4):

$$S_\epsilon^* = \arg\min_{S\in\Pi(\mathbf{1}_n,\mathbf{1}_n)} \langle C(w), S\rangle + \epsilon H(S).$$

This is achieved once we establish the similarity between this problem and the Entropic Optimal Transport (EOT) problem (Monge, 1781; Kantorovich, 1960).

Algorithm 1 ROBOT

Initialize w
while not converged **do**
  p^*, G, q^* =Sinkhorn($C(w); \epsilon$)
  $S_{\epsilon,ij}^*(w) = n p_i^* G_{ij} q_j^*$
  Compute $\nabla_w \mathcal{L}_\epsilon$ using (6)
  $w = w - \beta \nabla_w \mathcal{L}_\epsilon$
end while

The EOT problem considers two discrete distributions $\mathbb{P}_1$, $\mathbb{P}_2$ defined on supports $\mathcal{A} = \{a_i\}_{i=1}^n$, $\mathcal{B} = \{b_j\}_{j=1}^m$, respectively. Denote $\mathbb{P}_1(\{a_i\}) = \mu_i$, $\mathbb{P}_2(\{b_j\}) = \nu_j$, and let $\mu = [\mu_1, \ldots, \mu_n]^\top$, $\nu = [\nu_1, \ldots, \nu_m]^\top$. Further denote $M \in \mathbb{R}^{n\times m}$ the cost matrix with M_{ij} the transport cost. The EOT problem solves the following entropy regularized optimization problem,

$$\begin{aligned} &\Gamma^* = \arg\min_{\Gamma\in\Pi(\mu,\nu)} \langle M, \Gamma\rangle + \epsilon H(\Gamma), \\ &\text{with } \Pi(\mu,\nu) = \{\Gamma \in \mathbb{R}^{n\times m} : \Gamma \mathbf{1}_m = \mu, \Gamma^\top \mathbf{1}_n = \nu, \Gamma_{ij} \geq 0\}. \end{aligned} \tag{5}$$

Notice that problem (5) is a scaled lower problem of (4), i.e., $S_\epsilon^* = n\Gamma^*$, once we set $\mathcal{A} = \mathcal{D}_1$, $\mathcal{B} = \mathcal{D}_2$, $M = C(w)$, and $\mu = \nu = \mathbf{1}_n/n$. We adopt the Sinkhorn algorithm (Cuturi, 2013), which is commonly used to solve the EOT problem, to solve for S_ϵ^*. Specifically, we run iterative Bregman projections (Benamou et al., 2015)

$$p^{(\ell+1)} = \frac{\mu}{Gq^{(\ell)}}, \quad q^{(\ell+1)} = \frac{\nu}{G^\top p^{(\ell+1)}}, \quad \text{where } q^{(0)} = \frac{1}{n}\mathbf{1}_n,\ G_{ij} = \exp\left(\frac{-C_{ij}(w)}{\epsilon}\right).$$

Here the division is entrywise and $G \in \mathbb{R}^{n\times n}$. Denote p^* and q^* the stationary points of the Bregman projections. Then we obtain $S_{\epsilon,ij}^* = n p_i^* G_{ij} q_j^*$.

After obtaining $S_\epsilon^*(w)$, we solve the upper level problem in (4)

$$\min_w \mathcal{L}_\epsilon(w) = \langle C(w), S_\epsilon^*(w)\rangle$$

using first order optimization methods. The update rules can be obtained with the following theorem.

Theorem 3. Let $\mathcal{L}_\epsilon = \langle C(w), S_\epsilon^*(w)\rangle$, the gradient of $\mathcal{L}_\epsilon$ with respect to w is

$$\nabla_w \mathcal{L}_\epsilon = \frac{1}{\epsilon}\sum_{i,j=1}^{n,n}\left((1-C_{ij})S_{\epsilon,ij}^* + \sum_{h,\ell=1}^{n,n} C_{h\ell}S_{\epsilon,h\ell}^* P_{hij} + \sum_{h,\ell=1}^{n,n} C_{h\ell}S_{\epsilon,h\ell}^* Q_{\ell ij}\right)\nabla_w C_{ij}. \quad (6)$$

We defer the proof and the definition of P and Q to Appendix C due to the space limit. The complete algorithm is summarized in Algorithm 1.

3 ROBOT FOR ROBUST CORRESPONDENCE

We now introduce our robust formulation to solve rRWOC (Varol & Nejatbakhsh, 2019). In problem (4), the constraint $S \in \Pi(\mathbf{1}_n, \mathbf{1}_n)$ implies a one-to-one matching between $\mathcal{D}_1$ and $\mathcal{D}_2$. For rRWOC, such a matching may not exist, e.g., $|\mathcal{D}_1| \neq |\mathcal{D}_2|$. We resolve this by relaxing the constraint on S.

Concretely, instead of S_ϵ^* in (4), we solve for a robust correspondence[3] $S_{\mathrm{r}}^*(w)$, namely

$$S_{\mathrm{r}}^*(w), \bar{\mu}^*, \bar{\nu}^* = \arg\min_{S\in\Pi(\bar{\mu},\bar{\nu})} \langle C(w), S\rangle + \epsilon H(S), \quad (7)$$
$$\text{s.t. } \bar{\mu}^\top \mathbf{1}_n = n,\ \bar{\nu}^\top \mathbf{1}_m = m,\ \mathcal{F}(\bar{\mu}, \mathbf{1}_n) \leq \rho_1,\ \mathcal{F}(\bar{\nu}, \mathbf{1}_m) \leq \rho_2.$$

Here $n = |\mathcal{D}_1|$, $m = |\mathcal{D}_2|$, $S_{\mathrm{r}}^*(w) \in \mathbb{R}^{n\times m}$, and $\mathcal{F}$ is a divergence. We take $\mathcal{F}(\cdot,\cdot)$ as the squared Euclidean distance (more discussions in Appendix F). Problem (7) relaxes the marginal constraints $\Pi(\mathbf{1},\mathbf{1})$ in the original problem to $\Pi(\bar{\mu},\bar{\nu})$, where $\bar{\mu},\bar{\nu}$ are picked such that they do not deviate too much from $\mathbf{1}$. Note that we no longer require the number of data points in $\mathcal{D}_1$ and $\mathcal{D}_2$ to be the same.

Figure 1 illustrates different kinds of alignments. Suppose we obtain w, and correspondingly, $\widehat{y}^{(j)} = f(x^{(j)}; w)$, and we want to solve for S. In Figure 1, the scalars above the upper black line are $\widehat{y}^{(j)}$, while those below the lower black line are $y^{(i)}$ (the real outputs). The grey lines represent the non-negative values in $S_\epsilon^*(w)$ or $S_{\mathrm{r}}^*(w)$. Notice that there are two points far away from others, i.e., $10.87, 10.92$. The original formulation in (4) aligns all the scalars, while the robust formulation (7) automatically ignores the outliers.

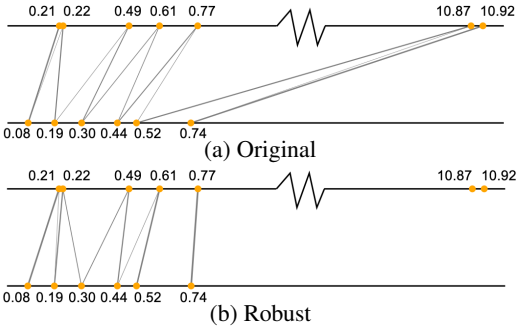


Figure 1: *Illustration of alignments.*

In practice, we run a Sinkhorn iteration and a projected gradient step iteratively (see Appendix D) to solve (7). After obtaining $S_{\mathrm{r}}^*(w)$, we solve the upper level optimization in (4) to obtain w^*, i.e.,

$$w^* = \arg\min_w \langle C(w), S_{\mathrm{r}}^*(w)\rangle.$$

Similar to the previous section, we use a first-order algorithm to solve this problem, and we derive explicit expressions for the update rules. See Appendix E for details.

Since our robust formulation does not require one-to-one correspondence between the datasets, we can run gradient based optimization methods in a stochastic manner. As a result, our proposed framework can scale to handle very large number of data points. We refer ROBOT for rRWOC as *ROBOT-robust* for the remaining of this paper.

4 EXPERIMENT

We evaluate ROBOT and ROBOT-robust on both synthetic and real-world datasets, including flow cytometry and multi-object tracking. We first present numerical results and then we provide insights in the discussion section. Experiment details and auxiliary results can be found in Appendix G.

4.1 UNLABELED SENSING

Data Generation. We follow the unlabeled sensing setting (Tsakiris & Peng, 2019) and generate $n = 1000$ data points $\{(y^{(i)}, x_2^{(i)})\}_{i=1}^n$, where $x_2^{(i)} \in \mathbb{R}^{d_2}$. Note here we take $d_1 = 0$, i.e., we only

[3]The idea is inspired by the marginal relaxation of optimal transport, first independently proposed by Kondratyev et al. (2016) and Chizat et al. (2018a), and later developed by Chizat et al. (2018c); Liero et al. (2018). Chizat et al. (2018b) share the same formulation as ours.

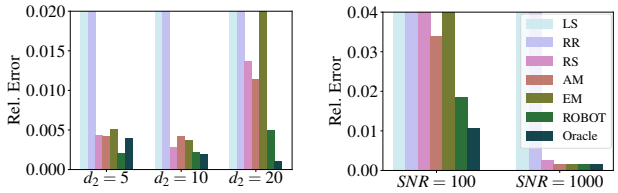
have the response in $\mathcal{D}_1$. We first generate $x_2^{(i)}, w \sim \mathcal{N}(\mathbf{0}_{d_2}, \boldsymbol{I}_{d_2})$, and $\varepsilon^{(i)} \sim \mathcal{N}(0, \rho^2_{\text{noise}})$. Then we compute $y^{(i)} = (x_2^{(i)})^\top w + \varepsilon^{(i)}$. We randomly permute the order of 50% of $x_2^{(i)}$ so that we lose the x-to-y correspondence. We generate the test set in the same way, only without permutation.

Baselines and Training. We consider the following scalable methods:

1. *Oracle*: Standard linear regression where no data are permuted.
2. *Least Squares* (*LS*): Standard linear regression, i.e., treating the data as if they are not permuted.
3. *Alternating Minimization* (*AM*, Abid et al. (2017)): We iteratively solve the correspondence given w, and update w using gradient descent with the correspondence.
4. *Stochastic EM* (Abid & Zou, 2018): A stochastic EM approach to recover the permutation.
5. *Robust Regression* (*RR*, Slawski & Ben-David (2019); Slawski et al. (2019a)). A two-stage block coordinate descent approach to discard outliers and fit regression models.
6. *Random Sample* (*RS*, Varol & Nejatbakhsh (2019)): A random sample consensus (RANSAC) approach to estimate w.

We initialize AM, EM and ROBOT using the output of RS with multi-start. We adopt a linear model $f(x; w) = x^\top w$. Models are evaluated by the relative error on the test set, i.e., error $= \sum_i (\widehat{y}^{(i)} - y^{(i)})^2 / \sum_i (y^{(i)} - \bar{y})^2$, where $\widehat{y}^{(i)}$ is the predicted label, and $\bar{y}$ is the mean of $\{y^{(i)}\}$.

Results. We visualize the results in Figure 2. In all the experiments, ROBOT achieves better results than the baselines. Note that the relative error is larger for all methods except Oracle as the dimension and the noise increase. For low dimensional data, e.g., $d_2 = 5$, our model achieves even better performance than Oracle.



(a) $\rho^2_{\text{noise}} = 0.1$ (b) $d_2 = 10$

Figure 2: *Unlabeled sensing. Results are the mean over 10 runs.* SNR$= \|w\|_2^2 / \rho^2_{\text{noise}}$ *is the signal-to-noise ratio.*

4.2 Nonlinear Regression

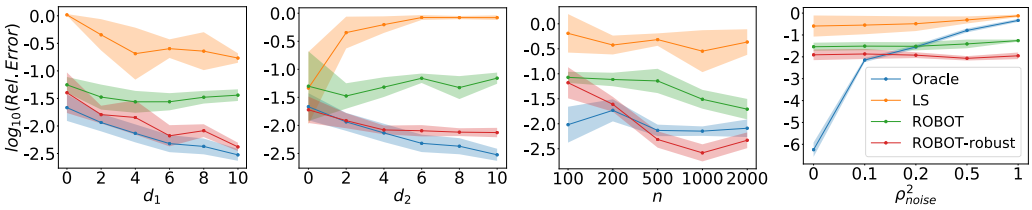


Figure 3: *Nonlinear regression. We use* $n = 1000$, $d_1 = 2$, $d_2 = 3$, $\rho^2_{\text{noise}} = 0.1$ *as defaults.*

Data Generation. We mimic the scenario where the dataset is collected from different platforms. Specifically, we generate n data points $\{(y^{(i)}, [x_1^{(i)}, x_2^{(i)}])\}_{i=1}^n$, where $x_1^{(i)} \in \mathbb{R}^{d_1}$ and $x_2^{(i)} \in \mathbb{R}^{d_2}$. We first generate $x_1^{(i)} \sim \mathcal{N}(\mathbf{0}_{d_1}, \boldsymbol{I}_{d_1})$, $x_2^{(i)} \sim \mathcal{N}(\mathbf{0}_{d_2}, \boldsymbol{I}_{d_2})$, $w \sim \mathcal{N}(\mathbf{0}_{d_1+d_1}, \boldsymbol{I}_{d_1+d_2})$, and $\varepsilon^{(i)} \sim \mathcal{N}(0, \rho^2_{\text{noise}})$. Then we compute $y^{(i)} = f([x_1^{(i)}, x_2^{(i)}]; w) + \varepsilon^{(i)}$. Next, we randomly permute the order of $\{x_2^{(i)}\}$ so that we lose the data correspondence. Here, $\mathcal{D}_1 = \{(x_1^{(i)}, y^{(i)})\}$ and $\mathcal{D}_2 = \{x_2^{(j)}\}$ mimic two parts of data collected from two separate platforms. Since we are interested in the response on platform one, we treat all data from platform two, i.e., $\mathcal{D}_2$, as well as 80% of data in $\mathcal{D}_1$ as the training data. The remaining data from $\mathcal{D}_1$ are the test data. Notice that we have different number of data on $\mathcal{D}_1$ and $\mathcal{D}_2$, i.e., the correspondence is not exactly one-to-one.

Baselines and Training. We consider a nonlinear function $f(x^{(i)}; w) = \sum_{k=1}^d \sin(x_k^{(i)} w_k)$. In this case, we consider only two baselines — Oracle and LS, since the other baselines in the previous section are designed for linear models. We evaluate the regression models by the transport cost divided by $\sum_i (y^{(i)} - \bar{y})^2$ on the test set.

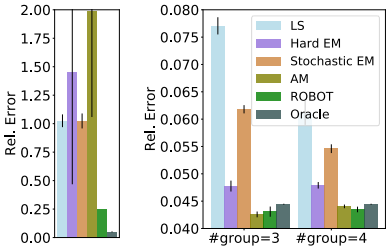
Results. As shown in Figure 3, ROBOT-robust consistently outperforms ROBOT and LS, demonstrating the effectiveness of our robust formulation. Moreover, ROBOT-robust achieves better performance than Oracle when the number of training data is large or when the noise level is high.

Table 1: *Experiment results on MOT.*

Data	Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MOT17 (train)	ROBOT	**48.3**	82.6	55.3	407	553	22,443	149,988	1,811
	w/o ROBOT	44.0	81.3	49.9	404	550	36,187	149,131	3,204
MOT17 (dev)	ROBOT	**48.2**	76.6	43.4	455	904	29,419	259,714	3,228
	w/o ROBOT	42.1	75.0	36.8	414	890	61,210	259,318	6,138
	SORT	43.1	77.8	39.8	295	997	28,398	287,582	4,852
MOT20 (train)	ROBOT	**56.2**	84.9	47.6	805	288	113,752	377,247	5,888
	w/o ROBOT	48.8	81.5	40.2	769	290	186,245	384,562	10,153
MOT20 (dev)	ROBOT	**45.0**	76.9	34.0	394	257	70,416	210,425	3,683
	w/o ROBOT	38.5	75.1	27.0	383	233	104,958	207,627	5,696
	SORT	42.7	78.5	45.1	208	326	27,521	264,694	4,470

4.3 Flow Cytometry

In flow cytometry (FC), a sample containing particles is suspended in a fluid and injected into the flow cytometer, but the measuring instruments are unable to preserve the correspondence between the particles and the measurements. Different from FC, gated flow cytometry (GFC) uses "gates" to sort the particles into one of many bins, which provides partial ordering information since the measurements are provided individually for each bin. In practice, there are usually 3 or 4 bins.



(a) *FC* (b) *GFC*

Figure 4: *Relative error.*

Settings. We adopt the dataset from Knight et al. (2009). Following Abid et al. (2017), the outputs $y^{(i)}$'s are normalized, and we select the top 20 significant features by a linear regression on the top 1400 items in the dataset. We use 90% of the data as the training data, and the remaining as test data. For ordinary FC, we randomly shuffle all the labels in the training set. For GFC, the training set is first sorted by the labels, and then divided into equal-sized groups, mimicking the sorting by gates process. The labels in each group are then randomly shuffled. To simulate gating error, 1% of the data are shuffled across the groups. We compare ROBOT with Oracle, LS, Hard EM (a variant of Stochastic EM proposed in Abid & Zou (2018)), Stochastic EM, and AM. We use relative error on the test set as the evaluation metric.

Results. As shown in Figure 4, while AM achieves good performance on GFC when the number of groups is 3, it behaves poorly on the FC task. ROBOT, on the other hand, is efficient on both tasks.

4.4 Multi-Object Tracking

In this section we extend our method to vision-based Multi-Object Tracking (MOT), a task with broad applications in mobile robotics and autonomous driving. Given a video and the current frame, the goal of MOT is to predict the locations of the objects in the next frame. Specifically, object detectors (Felzenszwalb et al., 2009; Ren et al., 2015) first provide us the potential locations of the objects by their bounding boxes. Then, MOT aims to assign the bounding boxes to trajectories that describe the path of individual objects over time. Here, we formulate the current frame and the objects' locations in the current frame as $\mathcal{D}_2 = \{x_2^{(j)}\}$, while we treat the next frame and the locations in the next frame as $\mathcal{D}_1 = \{(x_1^{(i)}, y^{(i)})\}$.

Figure 5: *One frame in* MOT20 *with detected bounding boxes in yellow.*

Existing deep learning based MOT algorithms require large amounts of annotated data, i.e., the ground truth of the correspondence, during training. Different from them, our algorithm does not require the correspondence between $\mathcal{D}_1$ and $\mathcal{D}_2$, and all we need is the video. This task is referred to as *unsupervised MOT* (He et al., 2019).

Related Works. To the best of our knowledge, the only method that accomplishes unsupervised end-to-end learning of MOT is He et al. (2019). However, it targets tracking with low densities, e.g., `Sprites-MOT`, which is different from our focus.

Settings. We adopt the `MOT17` (Milan et al., 2016) and the `MOT20` (Dendorfer et al., 2020) datasets. Scene densities of the two datasets are 31.8 and 170.9, respectively, which means the scenes are pretty crowded as we illustrated in Figure 5. We adopt the DPM detector (Felzenszwalb et al., 2009) on `MOT17` and the Faster-RCNN detector (Ren et al., 2015) on `MOT20` to provide us the bounding boxes. Inspired by Xu et al. (2019b), the cost matrix is computed as the average of the Euclidean center-point distance and the Jaccard distance between the bounding boxes,

$$C_{ij}(w) = \frac{1}{2}\left(\frac{\|c(f(x_2^{(i)};w)) - c(y^{(j)})\|_2}{\sqrt{H^2+W^2}} + \mathcal{J}(f(x_2^{(i)};w), y^{(j)})\right),$$

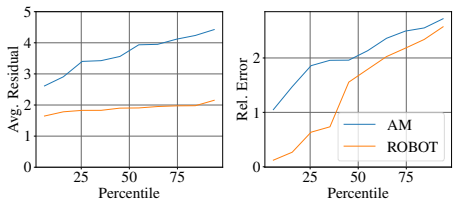
where $c(\cdot)$ is the location of the box center, H and W are the height and the width of the video frame, and $\mathcal{J}(\cdot,\cdot)$ is the Jaccard distance defined as 1-IoU (Intersection-over-Union). We utilize the single-object tracking model SiamRPN[4] (Li et al., 2018) as our regression model f. We apply ROBOT-robust with $\rho_1 = \rho_2 = 10^{-3}$. See Appendix G for more detailed settings.

Results. We demonstrate the experiment results in Table 1, where the evaluation metrics follow Ristani et al. (2016). In the table, $\uparrow$ represents the higher the better, and $\downarrow$ represents the lower the better. ROBOT signifies the model trained by ROBOT-robust, and w/o ROBOT means the pretrained model in Li et al. (2018). The scores are improved significantly after training with ROBOT-robust.

We also include the scores of the SORT model (Bewley et al., 2016) obtained from the dataset platform. Different from SiamRPN and SiamRPN+ROBOT, SORT is a supervised learning model. As shown, our unsupervised training framework achieves comparable or even better performance.

5 Discussion

Sensitivity to initialization. ROBOT is less sensitive to initialization than existing methods. As stated in Pananjady et al. (2017b), obtaining the global optima of (1) is in general an NP-hard problem. Therefore, most of the existing works either have exponential complexity, e.g., Elhami et al. (2017), or are sensitive to initialization. Our model is computationally efficient and greatly relieves the sensitivity issue.

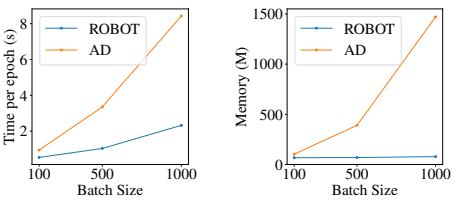


(a) Training residual (b) Test error

Figure 6: *Different initialization.*

We run AM and ROBOT with 10 different initialization, and then we sort the results based on (a) the averaged residual on the training set, and (b) the relative prediction error on the test set. We plot the percentiles in Figure 6. Here we use fully shuffled data under the unlabeled sensing setting, and we set $n = 1000$, $d = 5$, $\rho^2_{\text{noise}} = 0.1$, and $\epsilon = 10^{-2}$. We can see that ROBOT is able to find good solutions in 30% of the cases, but AM is more sensitive to the initialization and cannot find any reasonable solutions.

ROBOT v.s. Automatic Differentiation (AD). In our algorithm, we compute the Jacobian matrix directly for both ROBOT and ROBOT-robust. For the RWOC problem, this can also be done by AD. However, in order to apply AD, we need to store all the intermediate states for all the iterations when running the Sinkhorn algorithm. This imposes a memory requirement proportional to the number of iterations. Applying our explicit expression for the backward pass is more memory-efficient. Moreover, AD is more time-consuming than our method. The time and memory comparisons are shown in Figure 7, where we set $n = 1000$. For robust RWOC, AD cannot compute the Jacobian, since some steps in the forward pass are not differentiable.



(a) Time (b) Memory

Figure 7: *Time and memory comparison.*

[4]The initial weights of f are obtained from `https://github.com/foolwood/DaSiamRPN`.

REFERENCES

Abubakar Abid and James Zou. A stochastic expectation-maximization approach to shuffled linear regression. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 470–477. IEEE, 2018.

Abubakar Abid, Ada Poon, and James Zou. Linear regression with shuffled labels. *arXiv preprint arXiv:1705.01342*, 2017.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468. IEEE, 2016.

Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and fisher–rao metrics. *Foundations of Computational Mathematics*, 18(1):1–44, 2018a.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018b.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018c.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pp. 2292–2300, 2013.

P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, March 2020. URL http://arxiv.org/abs/1906.04567. arXiv: 2003.09003.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, 2008.

Golnooshsadat Elhami, Adam James Benjamin, Bejar Haro, and Martin Vetterli. Unlabeled sensing: Reconstruction algorithm and theoretical guarantees. In *2017 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. Convergence of learning dynamics in stackelberg games. *arXiv preprint arXiv:1906.01217*, 2019.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. *arXiv preprint arXiv:2006.16218*, 2020.

Saeid Haghighatshoar and Giuseppe Caire. Signal recovery from unlabeled samples. *IEEE Transactions on Signal Processing*, 66(5):1242–1257, 2017.

Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1318–1327, 2019.

Daniel J Hsu, Kevin Shi, and Xiaorui Sun. Linear regression without correspondence. In *Advances in Neural Information Processing Systems*, pp. 1531–1540, 2017.

Leonid Vitaliyevich Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

Christopher G Knight, Mark Platt, William Rowe, David C Wedge, Farid Khan, Philip JR Day, Andy McShea, Joshua Knowles, and Douglas B Kell. Array-based evolution of dna aptamers allows modelling of an explicit sequence-fitness landscape. *Nucleic acids research*, 37(1):e6–e6, 2009.

Stanislav Kondratyev, Léonard Monsaingeon, Dmitry Vorotnikov, et al. A new optimal transport distance on the space of finite radon measures. *Advances in Differential Equations*, 21(11/12): 1117–1164, 2016.

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, 2018.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211 (3):969–1117, 2018.

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. In *Advances in Neural Information Processing Systems*, pp. 5859–5870, 2018.

Marvin Marcus and Rimhak Ree. Diagonals of doubly stochastic matrices. *The Quarterly Journal of Mathematics*, 10(1):296–302, 1959.

A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. URL http://arxiv.org/abs/1603.00831. arXiv: 1603.00831.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with an unknown permutation: Statistical and computational limits. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing*, pp. 417—-424, 2016.

Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Denoising linear models with permuted data. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 446–450. IEEE, 2017a.

Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *IEEE Transactions on Information Theory*, 64(5):3286–3300, 2017b.

Liangzu Peng and Manolis C Tsakiris. Linear regression without correspondences via concave minimization. *arXiv preprint arXiv:2003.07706*, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

P. Rigollet and J. Weed. Uncoupled isotonic regression via minimum wasserstein deconvolution. *arXiv preprint arXiv:1806.10648*, 2018.

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pp. 17–35. Springer, 2016.

X. Shi, X. Lu, and T. Cai. Spherical regresion under mismatch corruption with application to automated knowledge translation. *arXiv preprint arXiv:1810.05679*, 2018.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

M. Slawski and E. Ben-David. Linear regression with sparsely permuted data. *Electronic Journal of Statistics*, 2019.

M. Slawski, E. Ben-David, and P. Li. A two-stage approach to multivariate linear regression with sparsely mismatched data. *arXiv preprint arXiv:1907.07148*, 2019a.

Martin Slawski, Mostafa Rahmani, and Ping Li. A sparse representation-based approach to linear regression with partially shuffled labels. In *35th Conference on Uncertainty in Artificial Intelligence, UAI 2019*, 2019b.

Jeffrey M Stanton. Galton, pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education*, 9(3), 2001.

Manolis Tsakiris and Liangzu Peng. Homomorphic sensing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6335–6344, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/tsakiris19a.html.

Manolis C Tsakiris, Liangzu Peng, Aldo Conca, Laurent Kneip, Yuanming Shi, Hayoung Choi, et al. An algebraic-geometric approach to shuffled linear regression. *arXiv preprint arXiv:1810.05440*, 2018.

Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing with random linear measurements. *IEEE Transactions on Information Theory*, 64(5):3237–3253, 2018.

Erdem Varol and Amin Nejatbakhsh. Robust approximate linear regression without correspondence. *arXiv preprint arXiv:1906.00273*, 2019.

Titouan Vayer, Rémi Flamary, Romain Tavenard, Laetitia Chapel, and Nicolas Courty. Sliced gromov-wasserstein. *arXiv preprint arXiv:1905.10124*, 2019.

John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2(0):5–12, 1953.

Mengdi Wang, Yichen Chen, Jialin Liu, and Yuantao Gu. Random multi-constraint projection: Stochastic gradient methods for convex optimization with many constraints. *arXiv preprint arXiv:1511.03760*, 2015.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k operator with optimal transport. *arXiv preprint arXiv:2002.06504*, 2020.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin. Gromov-wasserstein learning for graph matching and node embedding. *arXiv preprint arXiv:1901.06003*, 2019a.

Yihong Xu, Yutong Ban, Xavier Alameda-Pineda, and Radu Horaud. Deepmot: A differentiable framework for training multiple object trackers. *arXiv preprint arXiv:1906.06618*, 2019b.

Günter M Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012.