A Specialized Semismooth Newton Method for Kernel-Based Optimal Transport

Tianyi Lin
MIT

Marco Cuturi
Apple

Michael. I. Jordan
UC Berkeley

Abstract

Kernel-based optimal transport (OT) estimators offer an alternative, functional estimation procedure to address OT problems from samples. Recent works suggest that these estimators are more statistically efficient than plug-in (linear programming-based) OT estimators when comparing probability measures in high-dimensions (Vacher et al., 2021). Unfortunately, that statistical benefit comes at a very steep computational price: because their computation relies on the short-step interior-point method (SSIPM), which comes with a large iteration count in practice, these estimators quickly become intractable w.r.t. sample size n. To scale these estimators to larger n, we propose a nonsmooth fixed-point model for the kernel-based OT problem, and show that it can be efficiently solved via a specialized semismooth Newton (SSN) method: We show, exploring the problem's structure, that the per-iteration cost of performing one SSN step can be significantly reduced in practice. We prove that our SSN method achieves a global convergence rate of $O(1/\sqrt{k})$, and a local quadratic convergence rate under standard regularity conditions. We show substantial speedups over SSIPM on both synthetic and real datasets.

1 Introduction

Optimal transport (OT) theory (Santambrogio, 2015) provides a principled framework to compare probability distributions. OT has been used extensively in machine learning and related fields, notably for generative modeling (Montavon et al., 2016; Arjovsky et al., 2017; Genevay et al., 2018; Salimans et al., 2018; Tolstikhin et al., 2018), classification and clustering (Frogner et al., 2015; Srivastava et al., 2015; Ho et al., 2017), or domain adaptation (Courty et al., 2016, 2017; Redko et al., 2019), see Peyré and Cuturi (2019). OT is also impactful in applied areas such as neuroimaging (Janati et al., 2020) and cell trajectory prediction (Schiebinger et al., 2019; Yang et al., 2020).

Curse of Dimensionality. In most applications, the OT problem is seeded with the squared Euclidean distance as the ground cost, and instantiated with n samples. In that regime, OT estimation is known to suffer from the curse of dimensionality (Dudley, 1969; Fournier and Guillin, 2015; Weed and Bach, 2019): The standard plug-in estimator for the OT objective, which runs a linear program on those samples, converges to its population value at a rate of $O(n^{-2/d})$ (Chizat et al., 2020), hindering the adoption of OT in machine learning. Practitioners are aware of such limitations and use alternative computational schemes that improve computational complexity while carrying out statistical regularization.

Regularization. Quite a few works propose to regularize the OT problem: using entropy (Cuturi, 2013; Genevay et al., 2019; Mena and Niles-Weed, 2019), low-dimensional projections (Rabin et al., 2011; Bonneel et al., 2015; Paty and Cuturi, 2019; Kolouri et al., 2019; Nadjahi et al., 2020; Lin et al., 2020, 2021; Niles-Weed and Rigollet, 2022), bootstrap Sommerfeld and Munk (2018); Fatras et al. (2020) or neural networks (Amos et al., 2017; Makkuva et al., 2020; Korotin et al., 2021). The sample complexity of entropic OT is bounded by $O(\varepsilon^{-d/2}n^{-1/2})$ for a regularization strength $\varepsilon > 0$, while that of projected OT is bounded by $O(n^{-1/k})$ for projection dimension $k \leq d$. Although these bounds may seem dimension-free w.r.t. n, they deteriorate when η is small or k is large, losing relevance to the original OT problem (Chizat et al., 2020). Minibatch approaches are mostly used as fitting loss, while neural approaches are used with no guarantees.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Leveraging Smoothness. Alternative approaches build on strong smoothness assumptions on potentials or maps, such as *wavelet*-based estimators (Weed and Berthet, 2019; Hütter and Rigollet, 2021; Deb et al., 2021; Manole et al., 2021), which are minimax optimal but algorithmically intractable. These approaches stand in contrast to, e.g., entropic map estimators (Pooladian and Niles-Weed, 2021) which are cheap but still suffer from the curse of dimensionality. Recently, Vacher et al. (2021) closed this statistical-computational gap by designing an estimator that relies on kernel sums-of-squares, showing that it can be computed using a short-step interior-point method (SSIPM), with polynomial-time complexity guarantee. Unfortunately, the SSIPM is known to be ineffective Potra and Wright (2000), requiring a large number of iterations as sample size grows. This issue was specifically pointed out in (Vacher et al., 2021, p.11-12), and we do observe it experimentally (see Fig. 3).

Scaling up Kernel-based OT. While Vacher et al.'s method holds several promises on the statistical front, it does lack an efficient implementation. Such an implementation is needed if one wants to show that these theoretical benefits do translate into practical advantages. Muzellec et al. (2021) proposed to improve this computational outlook with an additional relaxation. Their mollified problem can be solved with simple gradient-based methods, but presents, however, a significant departure from the original kernel-based estimator and its guarantees. We follow in their footsteps but focus directly on improving the computational efficiency of Vacher et al.'s estimator. We address Vacher et al.'s original problem using the semismooth Newton (SSN) method (Mifflin, 1977; Qi and Sun, 1993, 1999; Ulbrich, 2011). Our contribution is therefore purely *computational*: Since our approach targets the same optimization problem, our estimators inherit the statistical guarantees proved in (Vacher et al., 2021). Note that SSN methods were recently used in an OT context by Liu et al. (2022), but in the unrelated setting of solving the multiscale min-cost-flow problem on grids.

Contributions. We propose a nonsmooth equation model for kernel-based OT problems. We use it to devise a specialized SSN method to compute kernel-based OT estimators, and prove a global rate of $O(1/\sqrt{k})$ (Theorem 4.7) and a local quadratic rate under standard regularity conditions (Theorem 4.8). We show how to significantly reduce the per-iteration cost of our algorithm by exploiting structure. Finally, we validate experimentally that SSN is substantially faster than SSIPM on both synthetic and real data, and use our estimators to produce OT (Monge) map estimators, benchmarked on single-cell data.

Organization. The remainder is organized as follows. In Section 3, we present the nonsmooth equation model for computing the kernel-based OT estimators and define the optimality notion based on the residual map. In Section 4, we propose and analyze the specialized SSN algorithm for computing the kernel-based OT estimators and prove that our algorithm achieves the global and local convergence rate guarantees. In Section 5, we conduct the experiments on both synthetic and real datasets, demonstrating that our algorithm can effectively compute the kernel-based OT estimators and is more efficient than short-step interior-point methods. In Section 6, we conclude this paper. In the supplementary material, we provide additional experimental results, and missing proofs for key results.

2 Further Related Works

Semismooth Newton (SSN) methods (Ulbrich, 2011) are a class of powerful and versatile algorithms for solving constrained optimization problems with PDEs, and variational inequalities (VIs). The notion of semismoothness was introduced by Mifflin (1977) for real-valued functions and then extended to vector-valued mappings by Qi and Sun (1993). A pioneering work on the SSN method was due to Solodov and Svaiter (1999), in which the authors proposed a globally convergent Newton method by exploiting the structure of monotonicity and established a local superlinear convergence rate under the conditions that the generalized Jacobian is semismooth and nonsingular at the global optimal solution. The convergence rate guarantee was later extended in Zhou and Toh (2005) to the setting where the generalized Jacobian is not nonsingular.

The SSN methods have received significant amount of attention due to its wide success in solving several structured convex problems to a high accuracy. In particular, such approach has been successfully applied to solving large-scale SDPs (Zhao et al., 2010; Yang et al., 2015), LASSO (Li et al., 2018), nearest correlation matrix estimation (Qi and Sun, 2011), clustering (Wang et al., 2010), sparse inverse covariance selection (Yang et al., 2013) and composite convex minimization (Xiao et al., 2018). The closest works to ours is Liu et al. (2022), who developed a fast SSN method to compute the plug-in OT estimator by exploring the sparsity and multiscale structure of its linear programming (LP) formulation. All of their experiments are run on 2D image grids. In contrast, our methods uses SSN to target a regularized, dual RKHS (functional) formulation, useful in higher dimensions. To our knowledge, this paper is the first to apply the SSN method to computing the kernel-based OT estimator and prove the

convergence rate guarantees.

3 Background: Kernel-Based OT

We formally define the OT problem and review the kernel-based OT estimator proposed by Vacher et al. (2021). Let X and Y be two bounded domains in $\mathbb{R}^d$ and let $\mathscr{P}(X)$ and $\mathscr{P}(Y)$ be the set of Borel probability measures in X and Y. Suppose that $\mu \in \mathscr{P}(X)$, $\nu \in \mathscr{P}(Y)$ and $\Pi(\mu,\nu)$ is the set of couplings between μ and ν, the primal OT problem is:

$$\mathrm{OT}(\mu,\nu) := \tfrac{1}{2}\left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{X\times Y} \|x-y\|^2 \, d\pi(x,y)\right),$$

while its dual formulation is stated as follows,

$$\begin{array}{ll} \sup\limits_{u,v\in C(\mathbb{R}^d)} & \int_X u(x)d\mu(x) + \int_Y v(y)d\nu(y), \\ \text{s.t.} & \frac{1}{2}\|x-y\|^2 \geq u(x)+v(y), \forall (x,y)\in X\times Y, \end{array}$$

where $C(\mathbb{R}^d)$ is the set of continuous functions on $\mathbb{R}^d$. Note that the supremum can be attained and the corresponding optimal dual functions $u_\star$ and $v_\star$ are referred to as the Kantorovich potentials (Santambrogio, 2015). This problem has a continuous constraint set, since $\frac{1}{2}\|x-y\|^2 \geq u(x)+v(y)$ must be satisfied on $X\times Y$. A natural approach is to take n points $\{(\tilde{x}_1,\tilde{y}_1),\ldots,(\tilde{x}_n,\tilde{y}_n)\} \subseteq X\times Y$ and consider the constraints $\frac{1}{2}\|\tilde{x}_i-\tilde{y}_i\|^2 \geq u(\tilde{x}_i)+v(\tilde{y}_i)$ for all $1\leq i\leq n$. However, it can not leverage the smoothness of potentials (Aubin-Frankowski and Szabó, 2020), yielding an error of $\Omega(n^{-1/d})$. Vacher et al. overcome this difficulty by replacing the inequality constraints with equivalent equality constraints, and considering these constraints over n points. Following their work, we use the following assumptions on the support sets X, Y and the densities of μ and ν.

Assumption 3.1 *Let $d \geq 1$ be the dimension and let $m > 2d+2$ be the order of smoothness. Then, we assume that (i) the support sets X, Y are convex, bounded, and open with Lipschitz boundaries; (ii) the densities of μ,ν are finite, bounded away from zero and m-times differentiable.*

Assumption 3.1 guarantees that the potentials $u_\star$ and $v_\star$ have a similar order of differentiability (De Philippis and Figalli, 2014), leading to an effective way to represent u and v via a *reproducing Kernel Hilbert space* (RKHS). In particular, we define $H^s(Z) := \{f \in L^2(Z) \mid \|f\|_{H^s(Z)} := \sum_{|\alpha|\leq s} \|D^\alpha f\|_{L^2(Z)} < +\infty\}$ and remark that $H^s(Z) \subseteq C^k(Z)$ for any $s > \frac{d}{2}+k$, where $k \geq 0$ is integer-valued. This guarantees that $H^{m+1}(X)$, $H^{m+1}(Y)$ and $H^m(X\times Y)$ are RKHS under Assumption 3.1 (Paulsen and Raghupathi, 2016) and they are associated with three bounded continuous feature maps $\phi_X : X \mapsto H^{m+1}(X)$, $\phi_Y : Y \mapsto H^{m+1}(Y)$ and $\phi_{XY} : X\times Y \mapsto H^m(X\times Y)$. For simplicity, we let $H_X = H^{m+1}(X)$, $H_Y = H^{m+1}(Y)$ and $H_{XY} = H^m(X\times Y)$. Vacher et al. (2021, Corollary 7) shows that (i) $u_\star \in H_X$ and $v_\star \in H_Y$ with

$$\int_X u(x)d\mu(x) = \langle u, w_\mu\rangle_{H_X}, \int_X v(y)d\nu(y) = \langle v, w_\nu\rangle_{H_Y},$$

where $w_\mu = \int_X \phi_X(x)d\mu(x)$ and $w_\nu = \int_Y \phi_Y(y)d\nu(y)$ are *kernel mean embeddings*; (ii) $A_\star \in \mathbb{S}^+(H_{XY})$[1] exists and satisfies the equality constraint as follows:

$$\tfrac{1}{2}\|x-y\|^2 - u_\star(x) - v_\star(y) = \langle \phi_{XY}(x,y), A_\star \phi_{XY}(x,y)\rangle_{H_{XY}}.$$

Putting these pieces yields a representation theorem for estimating the OT distance. Indeed, under Assumption 3.1, the dual OT problem is equivalent to the RKHS-based problem given by

$$\begin{array}{ll} \max\limits_{u,v,A} & \langle u, w_\mu\rangle_{H_X} + \langle v, w_\nu\rangle_{H_Y}, \\ \text{s.t.} & \frac{1}{2}\|x-y\|^2 - u(x) - v(y) \\ & = \langle \phi_{XY}(x,y), A\phi_{XY}(x,y)\rangle_{H_{XY}}. \end{array} \tag{3.1}$$

The above equation offers two advantages: (i) The equality constraint can be well approximated under Assumption 3.1; (ii) RKHSs allow the kernel trick: computing parameters are expressed in terms of *kernel functions* that correspond to

$$\begin{array}{lcl} k_X(x,x') & = & \langle \phi_X(x), \phi_X(x')\rangle_{H_X}, \\ k_Y(y,y') & = & \langle \phi_Y(y), \phi_Y(y')\rangle_{H_Y}, \\ k_{XY}((x,y),(x',y')) & = & \langle \phi_{XY}(x,y), \phi_{XY}(x',y')\rangle_{H_{XY}}, \end{array}$$

where the kernel functions are explicit and can be computed in $O(d)$ given the samples. The final step is to approximate Eq. (3.1) using the data $x_1,\ldots,x_{n_{\text{sample}}} \sim \mu$ and $y_1,\ldots,y_{n_{\text{sample}}} \sim \nu$, and the filling points $\{(\tilde{x}_1,\tilde{y}_1),\ldots,(\tilde{x}_n,\tilde{y}_n)\} \subseteq X\times Y$. Indeed, we define $\hat{\mu} = \frac{1}{n_{\text{sample}}}\sum_{i=1}^{n_{\text{sample}}} \delta_{x_i}$ and $\hat{\nu} = \frac{1}{n_{\text{sample}}}\sum_{i=1}^{n_{\text{sample}}} \delta_{y_i}$, and use $\langle u, w_{\hat{\mu}}\rangle_{H_X} + \langle v, w_{\hat{\nu}}\rangle_{H_Y}$ instead of $\langle u, w_\mu\rangle_{H_X} + \langle v, w_\nu\rangle_{H_Y}$ where $w_{\hat{\mu}} = \frac{1}{n_{\text{sample}}}\sum_{i=1}^{n_{\text{sample}}} \phi_X(x_i)$ and $w_{\hat{\nu}} = \frac{1}{n_{\text{sample}}}\sum_{i=1}^{n_{\text{sample}}} \phi_Y(y_i)$. We also impose *the penalization terms* for u, v, and A to alleviate the error induced by sampling the corresponding equality constraints. Then, the resulting problem with regularization parameters $\lambda_1, \lambda_2 > 0$ is summarized as follows:

$$\begin{array}{ll} \max\limits_{u,v,A} & \langle u, w_{\hat{\mu}}\rangle_{H_X} + \langle v, w_{\hat{\nu}}\rangle_{H_Y} \\ & -\lambda_1 \mathrm{Tr}(A) - \lambda_2(\|u\|^2_{H_X} + \|v\|^2_{H_Y}), \\ \text{s.t.} & \frac{1}{2}\|\tilde{x}_i-\tilde{y}_i\|^2 - u(\tilde{x}_i) - v(\tilde{y}_i) \\ & = \langle \phi_{XY}(\tilde{x}_i,\tilde{y}_i), A\phi_{XY}(\tilde{x}_i,\tilde{y}_i)\rangle_{H_{XY}}. \end{array} \tag{3.2}$$

[1] We refer to $\mathbb{S}^+(H_{XY})$ as the set of linear, positive and self-adjoint operators on H_{XY}.

Focusing on the case of $n_{\text{sample}} = \Theta(n)$, we let $\hat{u}_\star$ and $\hat{v}_\star$ be the unique maximizers of Eq. (3.2). Then, the estimator for $\text{OT}(\mu, \nu)$ we consider corresponds to

$$\widehat{\text{OT}}^n = \langle \hat{u}_\star, w_{\hat{\mu}} \rangle_{H_X} + \langle \hat{v}_\star, w_{\hat{\nu}} \rangle_{H_Y}. \tag{3.3}$$

Remark 3.2 *It follows from Vacher et al. (2021, Corollary 3) that the norm of empirical potentials can be controlled using $\lambda_1 = \tilde{\Theta}(n^{-1/2})$ and $\lambda_2 = \tilde{\Theta}(n^{-1/2})$ in high probability and the statistical rate is $\tilde{O}(n^{-1/2})$. Compared with plug-in OT estimators, the kernel-based OT estimators are better when sample size n is small (estimator is still tractable) and dimension d is large (statistical rates are $O(n^{-2/d})$ and $\tilde{O}(n^{-1/2})$ for plug-in and kernel-based estimators, respectively).*

Remark 3.3 *The entropic OT estimators achieve the rate of $\tilde{O}(n^{-1/2})$ for fixed ε (Genevay et al., 2019). Such a rate blows up exponentially fast to infinity as $\varepsilon \to 0$ if one wants to approximate non-regularized OT. Hence, entropic OT estimators are only statistically efficient for fixed, and fairly large, values of ε. In contrast, kernel-based OT estimators do not suffer from such a blow-up. While the constants depend exponentially in d, they are fixed, and the rate of $\tilde{O}(n^{-1/2})$ is valid for approximating non-regularized OT.*

Eq. (3.2) is an infinite-dimensional problem and is thus difficult to solve. Thanks to Vacher et al. (2021, Theorem 15), we have that the dual problem of Eq. (3.2) can be presented in a finite-dimensional space and strong duality holds true. Indeed, we define $Q \in \mathbb{R}^{n\times n}$ with $Q_{ij} = k_X(\tilde{x}_i, \tilde{x}_j) + k_Y(\tilde{y}_i, \tilde{y}_j)$, and $z \in \mathbb{R}^n$ with $z_i = w_{\hat{\mu}}(\tilde{x}_i) + w_{\hat{\nu}}(\tilde{y}_i) - \lambda_2 \|\tilde{x}_i - \tilde{y}_i\|^2$, and $q^2 = \|w_{\hat{\mu}}\|^2_{H_X} + \|w_{\hat{\nu}}\|_{H_Y}$, where we have

$$\begin{aligned}
w_{\hat{\mu}}(\tilde{x}_i) &= \tfrac{1}{n_{\text{sample}}} \textstyle\sum_{j=1}^{n_{\text{sample}}} k_X(x_j, \tilde{x}_i),\\
w_{\hat{\nu}}(\tilde{y}_i) &= \tfrac{1}{n_{\text{sample}}} \textstyle\sum_{j=1}^{n_{\text{sample}}} k_Y(y_j, \tilde{y}_i),\\
\|w_{\hat{\mu}}\|^2_{H_X} &= \tfrac{1}{n^2_{\text{sample}}} \textstyle\sum_{1\le i,j\le n_{\text{sample}}} k_X(x_i, x_j),\\
\|w_{\hat{\nu}}\|^2_{H_Y} &= \tfrac{1}{n^2_{\text{sample}}} \textstyle\sum_{1\le i,j\le n_{\text{sample}}} k_Y(y_i, y_j).
\end{aligned}$$

We define $K \in \mathbb{R}^{n\times n}$ with $K_{ij} = k_{XY}((\tilde{x}_i, \tilde{y}_i), (\tilde{x}_j, \tilde{y}_j))$ and R as an upper triangular matrix for the Cholesky decomposition of K. We let Φ_i be the i^{th} column of R. Then, the dual problem of Eq. (3.2) reads:

$$\begin{aligned}
\min_{\gamma\in\mathbb{R}^n} \quad & \tfrac{1}{4\lambda_2}\gamma^\top Q\gamma - \tfrac{1}{2\lambda_2}\gamma^\top z + \tfrac{q^2}{4\lambda_2},\\
\text{s.t.} \quad & \textstyle\sum_{i=1}^n \gamma_i \Phi_i \Phi_i^\top + \lambda_1 I \succeq 0.
\end{aligned} \tag{3.4}$$

Suppose that $\hat{\gamma}$ is one such minimizer, we have

$$\widehat{\text{OT}}^n = \tfrac{q^2}{2\lambda_2} - \tfrac{1}{2\lambda_2} \sum_{i=1}^n \hat{\gamma}_i (w_{\hat{\mu}}(\tilde{x}_i) + w_{\hat{\nu}}(\tilde{y}_i)).$$

To the best of our knowledge, the only method proposed to solve Eq. (3.4) is the SSIPM, for which the required number of iterations is known to grow as n grows. To avoid this issue, Muzellec et al. (2021) proposed solving an unconstrained relaxation model, which allows for the application of gradient-based methods. However, the estimators obtained from solving such relaxations lack any statistical guarantee.

4 Method and Analysis

In this section, we derive our algorithm and provide a convergence rate analysis. We define first a suitable root function that is optimized by kernel-based OT, and apply the regularized SSN method. We improve the computation of each SSN step by exploring the special structure of the generalized Jacobian of that function. We also safeguard the regularized SSN method using a min-max method to achieve a global rate.

4.1 A nonsmooth equation model for kernel-based OT

We define the operator $\Phi : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^n$ and its adjoint $\Phi^\star : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ as

$$\Phi(X) = \begin{pmatrix} \langle X, \Phi_1 \Phi_1^T \rangle \\ \vdots \\ \langle X, \Phi_n \Phi_n^T \rangle \end{pmatrix}, \quad \Phi^\star(\gamma) = \sum_{i=1}^n \gamma_i \Phi_i \Phi_i^T.$$

Clearly, Eq. (3.4) can be reformulated as the following optimization problem given by

$$\begin{aligned}
\min_{\gamma\in\mathbb{R}^n} \max_{X\in\mathcal{S}^n_+} \quad & \tfrac{1}{4\lambda_2}\gamma^T Q\gamma - \tfrac{1}{2\lambda_2}\gamma^T z + \tfrac{q^2}{4\lambda_2}\\
& -\langle X, \Phi^\star(\gamma) + \lambda_1 I\rangle.
\end{aligned} \tag{4.1}$$

We denote $w = (\gamma, X)$ as a vector-matrix pair and let $R : \mathbb{R}^n \times \mathbb{R}^{n\times n} \to \mathbb{R}^n \times \mathbb{R}^{n\times n}$ be given by

$$R(w) = \begin{pmatrix} \frac{1}{2\lambda_2} Q\gamma - \frac{1}{2\lambda_2} z - \Phi(X) \\ X - \text{proj}_{\mathcal{S}^n_+}(X - (\Phi^\star(\gamma) + \lambda_1 I)) \end{pmatrix}. \tag{4.2}$$

where $\mathcal{S}^n_+ = \{X \in \mathbb{R}^{n\times n} : X \succeq 0, X^T = X\}$. Then, we measure the optimality of w by monitoring $\|R(w)\|$, as supported by the following proposition linking R to minimizers of Eq. (3.4).

Proposition 4.1 *A point $\hat{\gamma}$ is an optimal solution of Eq. (3.4) if and only if $\hat{w} = (\hat{\gamma}, \hat{X})$ satisfies $R(\hat{w}) = 0$ for some $\hat{X} \in \mathcal{S}^n_+$.*

Proposition 4.1 shows that we can recover a kernel-based OT estimator by solving the nonsmooth equation model $R(w) = 0$.

Regularized SSN method. Since R is Lipschitz, Rademacher's theorem guarantess that R is almost

everywhere differentiable. We introduce generalized Jacobians (Clarke, 1990).

Definition 4.1 *Suppose R is Lipschitz and D_R is the set of differentiable points of R. The B-subdifferential at w is $\partial_B R(w) := \{\lim_{k\to+\infty} \nabla F(w^k) \mid w^k \in D_R, w^k \to w\}$ and the generalized Jacobian at w is $\partial R(w) = \mathrm{conv}(\partial_B R(w))$ where* conv *is the convex hull.*

The regularized SSN method for solving $R(w) = 0$ is as follows: Having the vector w_k, we compute $w_{k+1} = w_k + \Delta w_k$ where Δw_k is obtained by solving

$$(\mathcal{J}_k + \mu_k \mathcal{I})[\Delta w_k] = -r_k, \tag{4.3}$$

where $\mathcal{J}_k \in \partial R(w_k)$, $r_k = R(w_k)$ and $\mathcal{I}$ is the identity. The parameter is chosen as $\mu_k = \theta_k \|r_k\|$ to stabilize the SSN method in practice. If R is continuously differentiable and $\theta_k = 0$, the regularized SSN method reduces to the classical regularized Newton method which attains a local quadratic rate. Although the regularized SSN method is divergent in general (Kummer, 1988), its local superlinear rate has been proved if R is strongly semi-smooth (Qi and Sun, 1993; Zhou and Toh, 2005; Xiao et al., 2018).

4.2 Properties of the nonsmooth map R

Generalized Jacobian. Let us focus on the structure of the generalized Jacobian of $R(w)$. Using the definition of $\mathcal{S}_+^n$, one has $\mathrm{proj}_{\mathcal{S}_+^n}(Z) = P_\alpha \Sigma_\alpha P_\alpha^T$ where

$$Z = P\Sigma P^T = \begin{pmatrix} P_\alpha & P_{\bar\alpha} \end{pmatrix} \begin{pmatrix} \Sigma_\alpha & 0 \\ 0 & \Sigma_{\bar\alpha} \end{pmatrix} \begin{pmatrix} P_\alpha^T \\ P_{\bar\alpha}^T \end{pmatrix}, \tag{4.4}$$

with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, with the sets of indices of positive and nonpositive eigenvalues of Z written $\alpha = \{i \mid \sigma_i > 0\}$ and $\bar\alpha = \{1, \ldots, n\} \setminus \alpha$.

We define a generalized operator $\mathcal{M}(Z) \in \partial \mathrm{proj}_{\mathcal{S}_+^n}(Z)$ using its application to an $n \times n$ matrix S:

$$\mathcal{M}(Z)[S] = P(\Omega \circ (P^T S P))P^T \text{ for all } S \in \mathcal{S}_+^n,$$

where the $\circ$ symbol denotes a Hadamard product and $\Omega = \begin{pmatrix} E_{\alpha\alpha} & \eta_{\alpha\bar\alpha} \\ \eta_{\alpha\bar\alpha}^T & 0 \end{pmatrix}$ with $E_{\alpha\alpha}$ being a matrix of ones and $\eta_{ij} = \frac{\sigma_i}{\sigma_i - \sigma_j}$ for all $(i,j) \in \alpha \times \bar\alpha$. Note that all entries of Ω lie in the interval $(0, 1]$. In general, it is nontrivial to characterize the generalized Jacobian $\partial R(w)$ exactly but we can compute an element $\mathcal{J}(w) \in \partial R(w)$ using $\mathcal{M}(\cdot)$ as defined before.

We next introduce the definition of the (strong) semismoothness of an operator.

Definition 4.2 *Suppose that R is Lipschitz, we say it is (strongly) semismooth at w if (i) R is directionally differentiable at w; and (ii) for any $\mathcal{J} \in \partial R(w + \Delta w)$, we let $\Delta w \to 0$ and have*

$$\begin{aligned} \text{(semismooth)} \quad & \tfrac{\|R(w+\Delta w) - R(w) - \mathcal{J}[\Delta w]\|}{\|\Delta w\|} \to 0, \\ \text{(strongly semismooth)} \quad & \tfrac{\|R(w+\Delta w) - R(w) - \mathcal{J}[\Delta w]\|}{\|\Delta w\|^2} \le C. \end{aligned}$$

Algorithm 1 Solving Eq. (4.3) where $r_k = (r_k^1, r_k^2)$

1: $a^1 = -r_k^1 - \frac{1}{\mu_k + 1}(\Phi(r_k^2 + \mathcal{T}_k[r_k^2]))$ and $a^2 = -r_k^2$.
2: Solve $(\frac{1}{2\lambda_2}\mathcal{Q} + \mu_k \mathcal{I} + \Phi \mathcal{T}_k \Phi^\star)^{-1} \tilde a^1 = a^1$ inexactly and compute $\tilde a^2 = \frac{1}{\mu_k + 1}(a^2 + \mathcal{T}_k[a^2])$, where $\mathcal{T}_k[\cdot]$ is computed using the trick (Zhao et al., 2010).
3: Compute the direction $\Delta w_k = (\Delta w_k^1, \Delta w_k^2)$ by $\Delta w_k^1 = \tilde a^1$ and $\Delta w_k^2 = \tilde a^2 - \mathcal{T}_k[\Phi^\star(\tilde a^1)]$.

The following proposition characterizes the residual map given in Eq. (4.2) and guarantees that the SSN method is suitable to solve $R(w) = 0$.

Proposition 4.2 *The residual map R in Eq. (4.2) is strongly semismooth.*

4.3 Newton updates

We discuss how to compute the Newton direction Δw_k efficiently. From a computational point of view, it is not practical to solve the linear system in Eq. (4.3) exactly. Thus, we seek an approximation step Δw_k by solving Eq. (4.3) approximately such that

$$\|(\mathcal{J}_k + \mu_k \mathcal{I})[\Delta w_k] + r_k\| \le \tau \min\{1, \kappa \|r_k\| \|\Delta w_k\|\}, \tag{4.5}$$

where $0 < \tau, \kappa < 1$ are some positive constants and $\|\cdot\|$ is defined for a vector-matrix pair $w = (\gamma, X)$ (i.e., $\|w\| = \|\gamma\|_2 + \|X\|_F$ where $\|\cdot\|_2$ is Euclidean norm and $\|\cdot\|_F$ is Frobenius norm). Since $\mathcal{J}_k$ in Eq. (4.3) is nonsymmetric and its dimension is large, we use the Schur complement formula to transform Eq. (4.3) into a smaller symmetric system. If we vectorize the vector-matrix pair[2] Δw, the operators $\mathcal{M}(Z)$ and Φ can be expressed as matrices:

$$M(Z) = \tilde P \Gamma \tilde P^T \in \mathbb{R}^{n^2 \times n^2}, \; A = \begin{pmatrix} \Phi_1^T \otimes \Phi_1^T \\ \vdots \\ \Phi_n^T \otimes \Phi_n^T \end{pmatrix} \in \mathbb{R}^{n \times n^2},$$

where $\tilde P = P \otimes P$ and $\Gamma = \mathrm{diag}(\mathrm{vec}(\Omega))$.

We next provide a key lemma on the matrix form of $\mathcal{J}_k + \mu_k I$ at a given iterate $w_k = (\gamma_k, X_k)$.

Lemma 4.3 *Given $w_k = (\gamma_k, X_k)$, we compute $Z_k = X_k - (\Phi^\star(\gamma_k) + \lambda_1 I)$ and use Eq. (4.4) to obtain P_k, Σ_k, α_k and $\bar\alpha_k$. We then obtain Ω_k, $\tilde P_k = P_k \otimes P_k$*

[2] If $w = (\gamma, X)$ is a vector-matrix pair, we define $\mathrm{vec}(w) = (\gamma; \mathrm{vec}(X))$ as its vectorization.

and $\Gamma_k = \mathrm{diag}(\mathrm{vec}(\Omega_k))$. *Then, the matrix form of* $\mathcal{J}_k + \mu_k I$ *is given by*

$$(J_k + \mu_k I)^{-1} = C_1 B C_2, \text{ where}$$
$$C_1 = \begin{pmatrix} I & 0 \\ -T_k A^T & I \end{pmatrix}, \quad C_2 = \begin{pmatrix} I & \frac{1}{\mu_k+1}(A + AT_k) \\ 0 & I \end{pmatrix},$$
$$B = \begin{pmatrix} (\frac{1}{2\lambda_2}Q + \mu_k I + AT_k A^T)^{-1} & 0 \\ 0 & \frac{1}{\mu_k+1}(I + T_k) \end{pmatrix},$$

with $T_k = \tilde{P}_k L_k \tilde{P}_k^T$ *where* L_k *is a diagonal matrix with* $(L_k)_{ii} = \frac{(\Gamma_k)_{ii}}{\mu_k+1-(\Gamma_k)_{ii}}$ *and* $(\Gamma_k)_{ii} \in (0, 1]$ *is then denoted as the* i^{th} *diagonal entry of* Γ_k.

As a consequence of Lemma 4.3, the solution of Eq. (4.3) can be obtained by solving one certain symmetric linear system with the matrix $\frac{1}{2\lambda_2}Q + \mu_k I + AT_k A^T$. We remark that this system is well-defined since both Q and $AT_k A^T$ are positive semidefinite and the coefficient μ_k is chosen such that $\frac{1}{2\lambda_2}Q + \mu_k I + AT_k A^T$ is invertible. This also shows that Eq. (4.3) is well-defined.

We define $\mathcal{T}_k$ and $\mathcal{Q}$ as the operator form of $T_k = \tilde{P}_k L_k \tilde{P}_k^T$ and Q and write $r_k = (r_k^1, r_k^2)$ explicitly where $r_k^1 \in \mathbb{R}^n$ and $r_k^2 \in \mathbb{R}^{n\times n}$. Then, we have

$$\mathrm{vec}(a) = -\begin{pmatrix} I & \frac{1}{\mu_k+1}(A + AT) \\ 0 & I \end{pmatrix} \mathrm{vec}(r_k)$$
$$\Longrightarrow \begin{cases} a^1 = -r_k^1 - \frac{1}{\mu_k+1}(\Phi(r_k^2 + \mathcal{T}_k[r_k^2])), \\ a^2 = -r_k^2. \end{cases}$$

The next step consists in solving a new symmetric linear system and is given by

$$\mathrm{vec}(\tilde{a}) = \begin{pmatrix} \left(\frac{\mathcal{Q}}{2\lambda_2} + \mu_k I + AT_k A^T\right)^{-1} & 0 \\ 0 & \frac{1}{1+\mu_k}(I + T_k) \end{pmatrix} \mathrm{vec}(a),$$

which leads to

$$\begin{cases} \tilde{a}^1 = (\frac{1}{2\lambda_2}\mathcal{Q} + \mu_k \mathcal{I} + \Phi\mathcal{T}_k\Phi^\star)^{-1}a^1, \\ \tilde{a}^2 = \frac{1}{\mu_k+1}(a^2 + \mathcal{T}_k[a^2]). \end{cases}$$

Compared to Eq. (4.3) whose matrix form has size $(n^2+n)\times(n^2+n)$, we remark that the one in the step above is smaller with the size of $n \times n$ and can be efficiently solved using conjugate gradient (CG) or symmetric quasi-minimal residual (QMR) methods (Kelley, 1995; Saad, 2003). The final step is to compute the Newton direction $\Delta w_k = (\Delta w_k^1, \Delta w_k^2)$ as follows,

$$\mathrm{vec}(\Delta w_k) = \begin{pmatrix} I & 0 \\ -TA^T & I \end{pmatrix} \mathrm{vec}(\tilde{a})$$
$$\Longrightarrow \begin{cases} \Delta w_k^1 = \tilde{a}^1, \\ \Delta w_k^2 = \tilde{a}^2 - \mathcal{T}_k[\Phi^\star(\tilde{a}^1)]. \end{cases}$$

Algorithm 2 Our specialized SSN method

1: **Input:** τ, κ, $\alpha_2 \geq \alpha_1 > 0$, $\beta_0, \beta_1 < 1$, $\beta_2 > 1$ and $\underline{\theta}, \overline{\theta} > 0$.
2: **Initialization:** $v_0 = w_0 \in \mathbb{R}^n \times \mathcal{S}_+^n$ and $\theta_0 > 0$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Update v_{k+1} from v_k using one-step EG.
5: Select $\mathcal{J}_k \in \partial R(w_k)$.
6: Solve the linear system in Eq. (4.3) approximately such that Δw_k satisfies Eq. (4.5).
7: Compute $\tilde{w}_{k+1} = w_k + \Delta w_k$.
8: Update θ_{k+1} in the adaptive manner.
9: Set $w_{k+1} = \tilde{w}_{k+1}$ if $\|R(\tilde{w}_{k+1})\| \leq \|R(v_{k+1})\|$ is satisfied. Otherwise, set $w_{k+1} = v_{k+1}$.
10: **end for**

It remains to provide an efficient manner to compute $\mathcal{T}_k[\cdot]$. Since $\mathcal{T}_k$ is defined as the operator form of $T = \tilde{P}_k L_k \tilde{P}_k^T$, we have

$$\mathcal{T}_k[S] = P_k(\Psi_k \circ (P_k^T S P_k))P_k^T,$$

where Ψ_k is determined by μ_k and Ω_k: Indeed,

$$\Omega_k = \begin{pmatrix} E_{\alpha_k\alpha_k} & \eta_{\alpha_k\bar{\alpha}_k} \\ \eta_{\alpha_k\bar{\alpha}_k}^T & 0 \end{pmatrix} \Rightarrow \Psi_k = \begin{pmatrix} \frac{1}{\mu_k}E_{\alpha_k\alpha_k} & \xi_{\alpha_k\bar{\alpha}_k} \\ \xi_{\alpha_k\bar{\alpha}_k}^T & 0 \end{pmatrix},$$

where we have $\xi_{ij} = \frac{\eta_{ij}}{\mu_k+1-\eta_{ij}}$ for any $(i, j) \in \alpha_k \times \bar{\alpha}_k$. Following Zhao et al. (2010), we use the decomposition $\mathcal{T}_k[S] = G + G^T$ where $U = P_k(:, \alpha_k)^T S$ and

$$\begin{aligned} G &= P_k(:, \alpha_k)(\tfrac{1}{2\mu_k}(UP_k(:, \alpha_k))P_k(:, \alpha_k)^T \\ &\quad + \xi_{\alpha_k\bar{\alpha}_k} \circ (UP_k(:, \bar{\alpha}_k))P_k(:, \bar{\alpha}_k)^T). \end{aligned}$$

The number of flops for computing $\mathcal{T}_k[S]$ is $8|\alpha_k|n^2$. If $|\alpha_k| > \bar{\alpha}_k$, we use $\mathcal{T}_k[S] = \frac{1}{\mu_k}S - P_k((\frac{1}{\mu_k}E - \Psi_k) \circ (P_k^T S P_k))P_k^T$ with $8|\bar{\alpha}_k|n^2$ flops. Thus, we obtain an inexact solution of Eq. (4.3) efficiently whenever $|\alpha_k|$ or $|\bar{\alpha}_k|$ is small. We present the scheme for computing an inexact SSN direction in Algorithm 1. We propose a rule for updating θ_k where $\mu_k = \theta_k\|r_k\|$ will be used in Eq. (4.3). It is summarized as follows:

$$\theta_{k+1} = \begin{cases} \max\{\underline{\theta}, \beta_0\theta_k\}, & \text{if } \rho_k \geq \alpha_2\|\Delta w_k\|^2, \\ \beta_1\theta_k, & \text{if } \alpha_1\|\Delta w_k\|^2 \leq \rho_k < \alpha_2\|\Delta w_k\|^2, \\ \min\{\overline{\theta}, \beta_2\theta_k\}, & \text{otherwise.} \end{cases}$$

where $\rho_k = -\langle R(w_k + \Delta w_k), \Delta w_k\rangle$, $\beta_0 < 1, \beta_1, \beta_2 > 1$ and $\underline{\theta}, \overline{\theta} > 0$. Intuitively, θ_k can control the quality of Δw_k and the larger value of θ_k gives a slow yet stable convergence. The small value of $\frac{\rho_k}{\|\Delta w_k\|^2}$ implies that Δw_k is bad and we shall increase θ_k.

Remark 4.4 *We see that the per-iteration cost is significantly reduced since we have shown that solving the linear system in Eq.* (4.3) *whose matrix form has size* $(n^2+n)\times(n^2+n)$ *can be equivalently reduced to solving a much smaller linear system whose matrix form has size* $n \times n$. *Such equivalent reduction is based on Lemma 4.3 whose proof is summarized in Appendix C. This is one key contribution of our paper.*

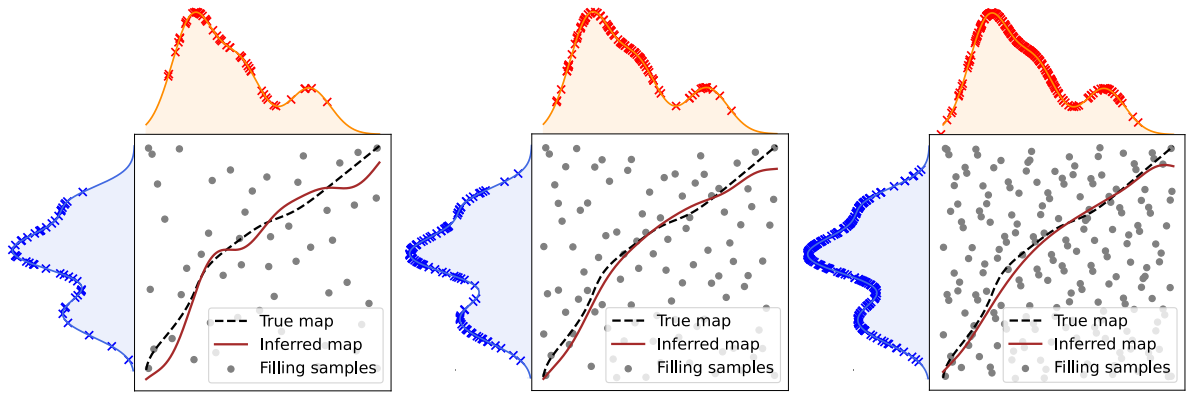

Figure 1: Visualization of the OT map with $n_{\text{sample}} = n \in \{50, 100, 200\}$.

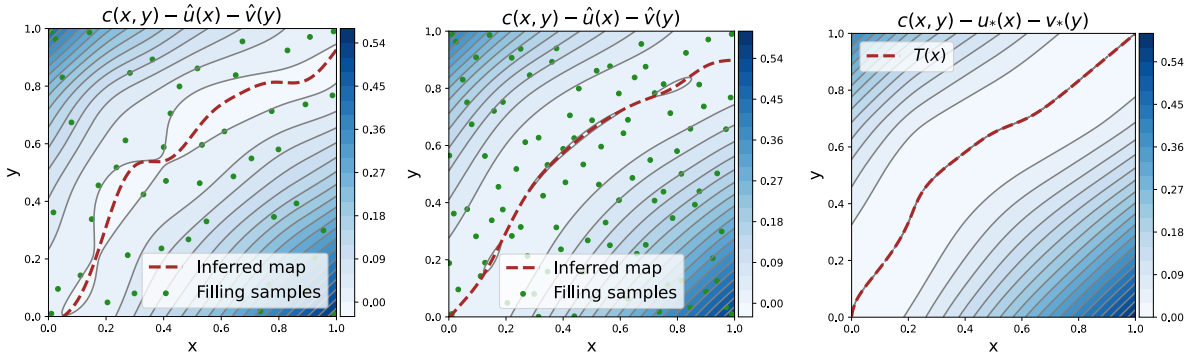

Figure 2: Visualization of the constraint: *(left, middle)* estimated, with $n_{\text{sample}} = n \in \{50, 100\}$, *(right)* ground truth.

4.4 Algorithm

We summarize our approach in Algorithm 2. We generate a sequence of iterates by alternating between extragradient (EG) method (Facchinei and Pang, 2007; Cai et al., 2022) and the regularized SSN method.

We maintain an auxiliary sequence of iterates $\{v_k\}_{k\geq 0}$. This sequence is directly generated by the EG method for solving the min-max problem in Eq. (4.1) and is used to safeguard the regularized SSN method to achieve a global convergence rate. In particular, we start with $v_0 = w_0 \in \mathbb{R}^n \times \mathcal{S}_+^n$ and perform the k^{th} iteration as follows,

1. Update v_{k+1} from v_k via 1-step EG.
2. Update $\tilde{w}_{k+1}$ from w_k via 1-step regularized SSN.
3. Set $w_{k+1} = \tilde{w}_{k+1}$ if $\|R(\tilde{w}_{k+1})\| \leq \|R(v_{k+1})\|$ and $w_{k+1} = v_{k+1}$ otherwise.

Remark 4.5 *The per-iteration cost would be $O(n^3)$ at worst case but it can be much cheaper in practice. Indeed, the $O(n^3)$ cost comes from exactly solving the $n \times n$ linear system. In our experiment, we use CG to approximately solve this linear system and set the maximum iteration number as 20. We can see that the average number of CG steps is less than 5. Also, our implementation can be improved by exploring the potentially sparsity of Q, A and T_k. In contrast, the linear system at each IPM step becomes severely ill-conditioned as the barrier parameter decreases and the matrix factorization has to be done exactly to achieve high precision. Therefore, our method suffers from the same per-iteration cost as IPM at worst case but can be more flexible and efficient from a practical viewpoint.*

Remark 4.6 *Although computing such auxiliary sequence results in extra cost, we can argue that it is not an issue in both theory and practice. Indeed, Theorem 4.8 guarantees the existence of a local region where 1-step regularized SSN can reduce the residue norm at a quadratic rate. This implies that $\|R(\tilde{w}_{k+1})\| \leq \|R(v_{k+1})\|$ will always hold when k is sufficiently large and $w_{k+1} = v_{k+1}$ will not never hold. This encourages us to stop computing the auxiliary sequence after the iterates enter the local region and only perform the regularized SSN steps. In our experiment, we also find that the iterates are mostly generated by regularized SSN steps. However, it is tricky to implement such strategy since it is hard to check if the generated iterates enter the local region. If we stop computing such auxiliary sequence too early, our algorithm is likely to diverge. To show the power of regularized SSN steps, we also compare our algorithm with pure EG steps in Appendix F (see Figure 5).*

4.5 Convergence Analysis

We establish the convergence guarantee of Algorithm 2 in the following theorems.

Theorem 4.7 *Suppose that $\{w_k\}_{k\geq 0}$ is a sequence of iterates generated by Algorithm 2. Then, the residuals of $\{w_k\}_{k\geq 0}$ converge to 0 at a rate of $1/\sqrt{k}$, i.e., $\|R(w_k)\| = O(1/\sqrt{k})$.*

In the context of constrained convex-concave min-max optimization, Cai et al. (2022) has proved the $O(1/\sqrt{k})$ last-iterate convergence rate of the EG, matching the lower bounds (Golowich et al., 2020b,a). Since the kernel-based OT estimation can be solved as a min-max problem, the global convergence rate in Theorem 4.7 demonstrates the efficiency of Algorithm 2. It remains unclear whether or not we can improve these results by exploring special structure of Eq. (4.1).

Moreover, such global rate depends on the smoothness parameter of Eq. (4.1) rather than the condition number of original formulation of Eq. (3.4). The explicit dependence on λ_1 and λ_2 is unknown since the results of Cai et al. (2022) does not provide the dependence on these problem parameters. Yet, our experiment has shown that our method behaves well when the sample size is medium ($\sim$1000) which is sufficient for kernel-based OT estimation in most cases.

Theorem 4.8 *Suppose that $\{w_k\}_{k\geq 0}$ is a sequence of iterates generated by Algorithm 2. Then, the residual norm at $\{w_k\}_{k\geq 0}$ converge to 0 at a quadratic rate, i.e., $\|R(w_{k+1})\| \leq C\|R(w_k)\|^2$ for some constant $C > 0$, if the initial point w_0 is sufficiently close to $w^\star$ with $R(w^\star) = 0$ and every element of $\partial R(w^\star)$ is invertible.*

Similar to classical Newton methods which are key ingredients for IPM, the regularized SSN methods enjoy the weak dependence on problem conditioning; see Qi and Sun (1993) for the details.

Remark 4.9 *Our algorithm becomes inefficient when ϵ is small but has better dependence on n than IPM. This is more desirable since the large n is necessary to ensure good statistical approximation (see Muzellec et al. (2021, Page 11-12) for details). Such trade-off between n and $1/\epsilon$ has occurred in the computation of plug-in estimators: despite worse dependence on $1/\epsilon$, the Sinkhorn method is recognized as more efficient than IPM in practice since many applications require low-accurate solution ($\epsilon \sim 10^{-2}$) when the sample size n is large. In addition, we remark that our algorithm does not downgrade the value of IPM since the latter one is more suitable when ϵ is small.*

5 Experiments

We present experimental results for kernel-based OT estimators run with our SSN algorithm. The baseline approach is the SSIPM (Vacher et al., 2021); we exclude the gradient-based method (Muzellec et al., 2021) from our experiment since it solves a different relaxation model. All experiments were conducted on a MacBook Pro with an Intel Core i9 2.4GHz and 16GB memory. For Algorithm 2, we set $\alpha_1 = 10^{-6}$, $\alpha_2 = 1.0$, $\beta_0 = 0.5$, $\beta_1 = 1.2$ and $\beta_2 = 5$.

SSIPM vs SSN on Synthetic data. Following the setup in Vacher et al. (2021), we draw n_{sample} samples from μ and n_{sample} samples from ν, where μ is a mixture of 3 d-dimensional Gaussian distributions and ν is a mixture of 5 d-dimensional Gaussian distributions. Then, we sample n filling samples from a $2d$ Sobol sequence. We also set the bandwidth $\sigma^2 = 0.005$ and parameters $\lambda_1 = \frac{1}{n}$ and $\lambda_2 = \frac{1}{\sqrt{n_{\text{sample}}}}$. Focusing on 1-dimensional setting, we report the visualization results in Figure 1 and 2 and verify that the inferred OT map gets closer to the true OT map as the number of filling points and data samples increase.

By varying the dimension $d \in \{2, 5, 10\}$, we report the computation efficiency results in Figure 3. It indicates that the our new algorithm is more efficient than the IPM as the number of filling points increases, with smaller variance in computation time (seconds). Here, we used the residue norm $\|R(w)\|$ as the measurement and terminated IPM and our method when $\|R(w)\|$ is below than the threshold 0.005. Although our method can scale to the case of 1000 samples which is relatively small compared to entropic OT methods, these results do start to open up some possibilities.

Entropic OT vs Kernel OT on Single-cell data. Comparing kernel-based OT estimators with plug-in OT estimators on synthetic data has been conducted in Vacher et al. (2021); Muzellec et al. (2021) and the results show that the kernel-based OT estimators behave better when the number of samples is small. We validate this claim using the real-world 4i datasets from Bunne et al. (2021), which track unaligned populations of cells *before* and *after* perturbations. Our experiments are conducted on 15 datasets with different drug perturbations. We consider as a baseline approach Pooladian and Niles-Weed's entropic map estimator, as implemented in the `OTT` package (Cuturi et al., 2022). We use their default implementation, which relies on an adaptive choice for the entropic regularization parameter ε.

Due to space limits, we only present the results on 6 datasets in Figure 4 and defer the results on other

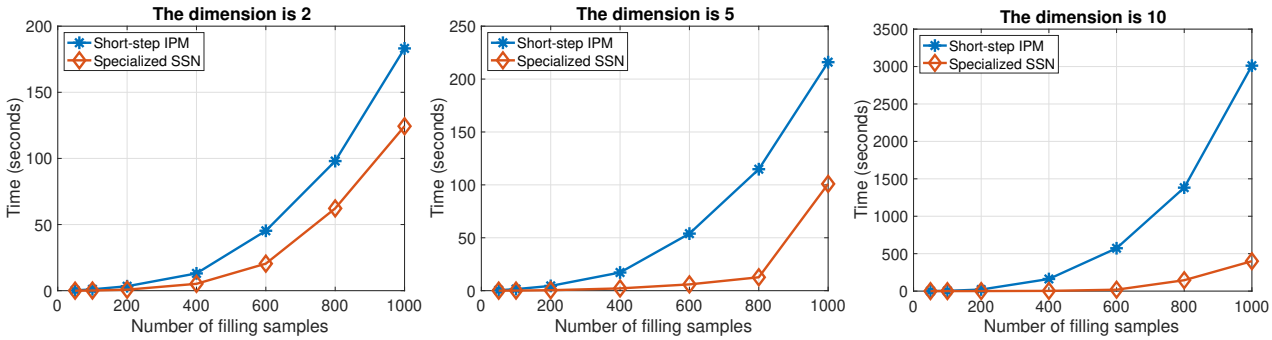

Figure 3: Comparisons of mean computation time of IPM vs. our algorithm (SSN) on CPU time.

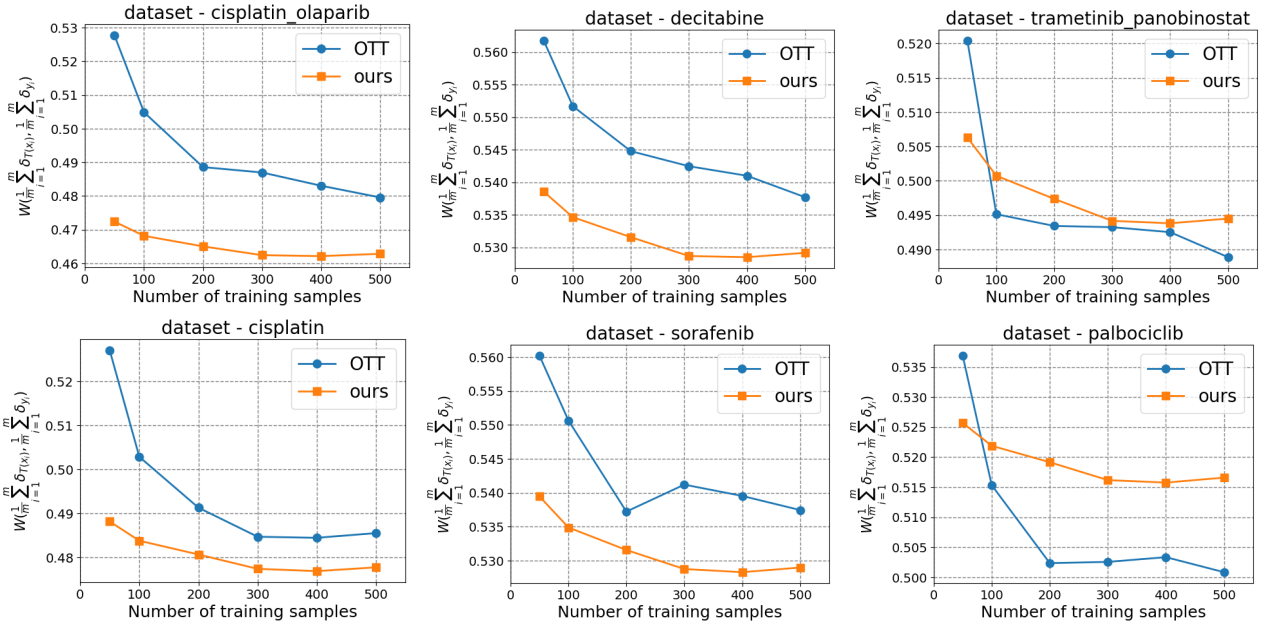

Figure 4: Performance of entropic map (using `OTT`) vs. kernel-based OT estimators computed with the SSN algorithm on 6 drug perturbation datasets. X-axis represent the number of training samples and Y-axis represents the error induced by OT map T on test samples in terms of OT distance.

datasets to Appendix F. We observe that kernel-based OT estimators (computed using our SSN method) achieve satisfactory performance and behave better when the number of training samples is small; indeed, they are better on 6 datasets, comparable on 5 datasets and worse on 4 datasets. While we do expect that the entropic estimator will eventually scale, and outperform our algorithm as the number of training samples increases, these experiments show that kernel-based OT estimation provides a fairly effective alternative when the number of training samples is small, which is consistent with previous observations on synthetic data (Vacher et al., 2021; Muzellec et al., 2021). These results therefore validate the sample efficiency of our algorithm for computing kernel-based OT Monge map estimators in small n large d regimes. Note that the performance drop on the palbocilib dataset for large sample sizes agrees with this. To speculate, the larger gap might be because of a low-rank structure within the palbocilib data, which can be better exploited by entropic regularized methods.

6 Concluding Remarks

In this paper, we propose a nonsmooth equation model for computing kernel-based OT estimators and show that its special problem structure allows it to be solved in an efficient manner using a SSN method. Specifically, we propose a specialized SSN method that achieves low per-iteration cost by exploiting such structure, and prove a global sublinear rate and a local quadratic rate under standard regularity conditions. Experimental results on synthetic data show that our algorithm is more efficient than the short-step IPM (Vacher et al., 2021), and the results on real data demonstrate its effectiveness. We hope this progress can motivate further improvements and/or modifications of kernel-based OT approaches.

References

B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *ICML*, pages 146–155. PMLR, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

P-C. Aubin-Frankowski and Z. Szabó. Hard shape-constrained kernel machines. In *NeurIPS*, pages 384–395, 2020.

N. Bonneel, J. Rabin, G. Peyré, and H. Pfister. Sliced and radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1): 22–45, 2015.

C. Bunne, S. G. Stark, G. Gut, J. S. del Castillo, K-V. Lehmann, L. Pelkmans, A. Krause, and G. Ratsch. Learning single-cell perturbation responses using neural optimal transport. *BioRxiv*, 2021.

Y. Cai, A. Oikonomou, and W. Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *NeurIPS*, pages 33904–33919, 2022.

L. Chizat, P. Roussillon, F. Léger, F-X. Vialard, and G. Peyré. Faster Wasserstein distance estimation with the Sinkhorn divergence. In *NeurIPS*, pages 2257–2269, 2020.

F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, 1990.

N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2016.

N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *NIPS*, pages 3733–3742, 2017.

M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013.

M. Cuturi, L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul. Optimal transport tools (OTT): A jax toolbox for all things wasserstein. *ArXiv Preprint: 2201.12324*, 2022.

G. De Philippis and A. Figalli. The Monge-Ampère equation and its link to optimal transportation. *Bulletin of the American Mathematical Society*, 51(4): 527–580, 2014.

N. Deb, P. Ghosal, and B. Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. In *NeurIPS*, pages 29736–29753, 2021.

R. M. Dudley. The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50, 1969.

F. Facchinei and J-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Science & Business Media, 2007.

K. Fatras, Y. Zine, R. Flamary, R. Gribonval, and N. Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. In *AISTATS*, pages 2131–2141. PMLR, 2020.

N. Fournier and A. Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3): 707–738, 2015.

C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio. Learning with a Wasserstein loss. In *NIPS*, pages 2053–2061, 2015.

A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with Sinkhorn divergences. In *AISTATS*, pages 1608–1617, 2018.

A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. Sample complexity of Sinkhorn divergences. In *AISTATS*, pages 1574–1583. PMLR, 2019.

N. Golowich, S. Pattathil, and C. Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In *NeurIPS*, pages 20766–20778, 2020a.

N. Golowich, S. Pattathil, C. Daskalakis, and A. Ozdaglar. Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems. In *COLT*, pages 1758–1784. PMLR, 2020b.

N. Ho, X. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh, and D. Phung. Multilevel clustering via Wasserstein means. In *ICML*, pages 1501–1509. PMLR, 2017.

J-C. Hütter and P. Rigollet. Minimax estimation of smooth optimal transport maps. *The Annals of Statistics*, 49(2):1166–1194, 2021.

H. Janati, T. Bazeille, B. Thirion, M. Cuturi, and A. Gramfort. Multi-subject MEG/EEG source imaging with sparse multi-task regression. *NeuroImage*, 220:116847, 2020.

C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.

S. Kolouri, K. Nadjahi, U. Şimşekli, R. Badeau, and G. K. Rohde. Generalized sliced Wasserstein distances. In *NIPS*, pages 261–272, 2019.

A. Korotin, V. Egiazarian, A. Asadulaev, A. Safin, and E. Burnaev. Wasserstein-2 generative networks. In *ICLR*, 2021. URL `https://openreview.net/forum?id=bEoxzW_EXsa`.

B. Kummer. Newton's method for non-differentiable functions. *Advances in Mathematical Optimization*, 45(1988):114–125, 1988.

X. Li, D. Sun, and K-C. Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM Journal on Optimization*, 28(1):433–458, 2018.

T. Lin, C. Fan, N. Ho, M. Cuturi, and M. I. Jordan. Projection robust Wasserstein distance and Riemannian optimization. In *NeurIPS*, pages 9383–9397, 2020.

T. Lin, Z. Zheng, E. Chen, M. Cuturi, and M. I. Jordan. On projection robust optimal transport: Sample complexity and model misspecification. In *AISTATS*, pages 262–270. PMLR, 2021.

Y. Liu, Z. Wen, and W. Yin. A multiscale semismooth Newton method for optimal transport. *Journal of Scientific Computing*, 91(2):1–29, 2022.

A. Makkuva, A. Taghvaei, S. Oh, and J. Lee. Optimal transport mapping via input convex neural networks. In *ICML*, pages 6672–6681. PMLR, 2020.

T. Manole, S. Balakrishnan, J. Niles-Weed, and L. Wasserman. Plugin estimation of smooth optimal transport maps. *ArXiv Preprint: 2107.12364*, 2021.

G. Mena and J. Niles-Weed. Statistical bounds for entropic optimal transport: Sample complexity and the central limit theorem. In *NIPS*, pages 4541–4551, 2019.

R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15(6):959–972, 1977.

G. Montavon, K-R. Müller, and M. Cuturi. Wasserstein training of restricted Boltzmann machines. In *NIPS*, pages 3718–3726, 2016.

B. Muzellec, A. Vacher, F. Bach, F-X. Vialard, and A. Rudi. Near-optimal estimation of smooth transport maps with kernel sums-of-squares. *ArXiv Preprint: 2112.01907*, 2021.

K. Nadjahi, A. Durmus, L. Chizat, S. Kolouri, S. Shahrampour, and U. Şimşekli. Statistical and topological properties of sliced probability divergences. In *NeurIPS*, pages 20802–20812, 2020.

J. Niles-Weed and P. Rigollet. Estimation of Wasserstein distances in the spiked transport model. *Bernoulli*, 28(4):2663–2688, 2022.

F-P. Paty and M. Cuturi. Subspace robust Wasserstein distances. In *ICML*, pages 5072–5081. PMLR, 2019.

V. I. Paulsen and M. Raghupathi. *An Introduction to The Theory of Reproducing Kernel Hilbert Spaces*, volume 152. Cambridge University Press, 2016.

G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607, 2019.

A-A. Pooladian and J. Niles-Weed. Entropic estimation of optimal transport maps. *ArXiv Preprint: 2109.12004*, 2021.

F. A. Potra and S. J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

H. Qi and D. Sun. An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis*, 31(2):491–511, 2011.

L. Qi and D. Sun. A survey of some nonsmooth equations and smoothing Newton methods. In *Progress in Optimization*, pages 121–146. Springer, 1999.

L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1): 353–367, 1993.

J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011.

I. Redko, N. Courty, R. Flamary, and D. Tuia. Optimal transport for multi-source domain adaptation under target shift. In *AISTATS*, pages 849–858. PMLR, 2019.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

T. Salimans, H. Zhang, A. Radford, and D. Metaxas. Improving GANs using optimal transport. In *ICLR*, 2018. URL `https://openreview.net/forum?id=rkQkBnJAb`.

F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, volume 87. Birkhäuser, 2015.

G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, and P. Berube. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

M. V. Solodov and B. F. Svaiter. A globally convergent inexact Newton method for systems of monotone equations. *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pages 355–369, 1999.

M. Sommerfeld and A. Munk. Inference for empirical Wasserstein distances on finite spaces. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(1):219–238, 2018.

S. Srivastava, V. Cevher, Q. Dinh, and D. Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *AISTATS*, pages 912–920. PMLR, 2015.

D. Sun and J. Sun. Semismooth matrix-valued functions. *Mathematics of Operations Research*, 27(1): 150–169, 2002.

I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. In *ICLR*, 2018.

M. Ulbrich. *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*. SIAM, 2011.

A. Vacher, B. Muzellec, A. Rudi, F. Bach, and F-X. Vialard. A dimension-free computational upper-bound for smooth optimal transport estimation. In *COLT*, pages 4143–4173. PMLR, 2021.

C. Wang, D. Sun, and K-C. Toh. Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM Journal on Optimization*, 20(6):2994–3013, 2010.

J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019.

J. Weed and Q. Berthet. Estimation of smooth densities in Wasserstein distance. In *COLT*, pages 3118–3119. PMLR, 2019.

X. Xiao, Y. Li, Z. Wen, and L. Zhang. A regularized semismooth Newton method with projection steps for composite convex programs. *Journal of Scientific Computing*, 76(1):364–389, 2018.

J. Yang, D. Sun, and K-C. Toh. A proximal point algorithm for log-determinant optimization with group Lasso regularization. *SIAM Journal on Optimization*, 23(2):857–893, 2013.

K. D. Yang, K. Damodaran, S. Venkatachalapathy, A. C. Soylemezoglu, G. V. Shivashankar, and C. Uhler. Predicting cell lineages using autoencoders and optimal transport. *PLoS Computational Biology*, 16 (4):e1007828, 2020.

L. Yang, D. Sun, and K-C. Toh. SDPNAL++: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3):331–366, 2015.

X-Y. Zhao, D. Sun, and K-C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4): 1737–1765, 2010.

G. Zhou and K-C. Toh. Superlinear convergence of a Newton-type algorithm for monotone equations. *Journal of Optimization Theory and Applications*, 125(1):205–221, 2005.

A Proof of Proposition 4.1

We first prove that $\hat{\gamma}$ is an optimal solution of Eq. (3.4) if $\hat{w} = (\hat{\gamma}, \hat{X})$ satisfies $R(\hat{w}) = 0$ for some $\hat{X} \succeq 0$. Indeed, by the definition of R from Eq. (4.2), we have

$$\tfrac{1}{2\lambda_2} Q\hat{\gamma} - \tfrac{1}{2\lambda_2} z - \Phi(\hat{X}) = 0, \tag{A.1}$$

and

$$\hat{X} - \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) = 0. \tag{A.2}$$

By the definition of $\mathrm{proj}_{\mathcal{S}_+^n}$, we have

$$\langle X - \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)), \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) - \hat{X} + (\Phi^\star(\hat{\gamma}) + \lambda_1 I)\rangle \geq 0 \text{ for all } X \succeq 0.$$

Plugging Eq. (A.2) into the above inequality yields that

$$\langle X - \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0 \text{ for all } X \succeq 0.$$

By setting $X = 0$ and $X = 2\hat{X}$, we have $\langle \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \leq 0$ and $\langle \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0$. Thus, we have

$$\langle \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle = 0, \quad \langle X, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0 \text{ for all } X \succeq 0. \tag{A.3}$$

Suppose that $\gamma \in \mathbb{R}^n$ satisfies that $\Phi^\star(\gamma) + \lambda_1 I \succeq 0$, we have

$$\begin{aligned} 0 &\overset{(A.1)}{=} (\gamma - \hat{\gamma})^\top \left(\tfrac{1}{2\lambda_2} Q\hat{\gamma} - \tfrac{1}{2\lambda_2} z - \Phi(\hat{X})\right) \\ &= \left(\tfrac{1}{4\lambda_2}\gamma^\top Q\gamma - \tfrac{1}{2\lambda_2}\gamma^\top z\right) - \left(\tfrac{1}{4\lambda_2}\hat{\gamma}^\top Q\hat{\gamma} - \tfrac{1}{2\lambda_2}\hat{\gamma}^\top z\right) - \tfrac{1}{4\lambda_2}(\gamma - \hat{\gamma})^\top Q(\gamma - \hat{\gamma}) - (\gamma - \hat{\gamma})^\top \Phi(\hat{X}) \\ &\leq \left(\tfrac{1}{4\lambda_2}\gamma^\top Q\gamma - \tfrac{1}{2\lambda_2}\gamma^\top z\right) - \left(\tfrac{1}{4\lambda_2}\hat{\gamma}^\top Q\hat{\gamma} - \tfrac{1}{2\lambda_2}\hat{\gamma}^\top z\right) - (\gamma - \hat{\gamma})^\top \Phi(\hat{X}) \end{aligned}$$

Since $\Phi^\star$ is the adjoint of Φ, we have $(\gamma - \hat{\gamma})^\top \Phi(\hat{X}) = \langle \hat{X}, \Phi^\star(\gamma) - \Phi^\star(\hat{\gamma})\rangle$. By combining this equality with $\Phi^\star(\gamma) + \lambda_1 I \succeq 0$ and the first equality in Eq. (A.3), we have

$$(\gamma - \hat{\gamma})^\top \Phi(\hat{X}) = \langle \hat{X}, \Phi^\star(\gamma) + \lambda_1 I\rangle - \langle \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0.$$

Thus, we have

$$0 \leq \left(\tfrac{1}{4\lambda_2}\gamma^\top Q\gamma - \tfrac{1}{2\lambda_2}\gamma^\top z + \tfrac{q^2}{4\lambda_2}\right) - \left(\tfrac{1}{4\lambda_2}\hat{\gamma}^\top Q\hat{\gamma} - \tfrac{1}{2\lambda_2}\hat{\gamma}^\top z + \tfrac{q^2}{4\lambda_2}\right).$$

Combining the above inequality with the second inequality in Eq. (A.3) yields the desired result.

It suffices to prove that satisfies $R(\hat{w}) = 0$ for some $\hat{X} \succeq 0$ if $\hat{\gamma}$ is an optimal solution of Eq. (3.4). Indeed, we write that $\sum_{i=1}^n \hat{\gamma}_i \Phi_i \Phi_i^\top + \lambda_1 I \succeq 0$ and

$$\tfrac{1}{4\lambda_2}\hat{\gamma}^\top Q\hat{\gamma} - \tfrac{1}{2\lambda_2}\hat{\gamma}^\top z + \tfrac{q^2}{4\lambda_2} \leq \tfrac{1}{4\lambda_2}\gamma^\top Q\gamma - \tfrac{1}{2\lambda_2}\gamma^\top z + \tfrac{q^2}{4\lambda_2},$$

for all $\gamma \in \mathbb{R}^n$ satisfying that $\sum_{i=1}^n \gamma_i \Phi_i \Phi_i^\top + \lambda_1 I \succeq 0$. Then, the KKT condition guarantees that there exists some $\hat{X} \succeq 0$ satisfying that

$$\begin{aligned} \textstyle\sum_{i=1}^n \hat{\gamma}_i \Phi_i \Phi_i^\top + \lambda_1 I &\succeq 0, \\ \tfrac{1}{2\lambda_2} Q\hat{\gamma} - \tfrac{1}{2\lambda_2} z - \Phi(\hat{X}) &= 0, \\ \langle \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle &= 0. \end{aligned} \tag{A.4}$$

The first and third inequalities guarantee that

$$\langle X - \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0 \text{ for all } X \succeq 0.$$

By letting $X = \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I))$, we have

$$\langle \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) - \hat{X}, \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \geq 0. \tag{A.5}$$

Recall that the definition of $\mathrm{proj}_{\mathcal{S}_+^n}$ implies that

$$\langle X - \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)), \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) - \hat{X} + (\Phi^\star(\hat{\gamma}) + \lambda_1 I)\rangle \geq 0 \text{ for all } X \succeq 0.$$

By letting $X = \hat{X}$, we have

$$\|\mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) - \hat{X}\|^2 \leq \langle \hat{X} - \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)), \Phi^\star(\hat{\gamma}) + \lambda_1 I\rangle \overset{(A.5)}{\leq} 0.$$

Combining the above inequality with the second equality in Eq. (A.4) yields that

$$\tfrac{1}{2\lambda_2} Q\hat{\gamma} - \tfrac{1}{2\lambda_2} z - \Phi(\hat{X}) = 0, \qquad \hat{X} - \mathrm{proj}_{\mathcal{S}_+^n}(\hat{X} - (\Phi^\star(\hat{\gamma}) + \lambda_1 I)) = 0.$$

Combining these inequalities with the definition of R implies $R(\hat{w}) = 0$ and hence the desired result.

B Proof of Proposition 4.2

The strong semismoothness of R follows from the derivation given in Sun and Sun (2002) to establish the semismoothness of projection operators. Indeed, the projection over a positive semidefinite cone is guaranteed to be strongly semismooth (Sun and Sun, 2002, Corollary 4.15). Thus, we have that $\mathrm{proj}_{\mathcal{S}_+^n}(\cdot)$ is strongly semismooth. Since the strong semismoothness is closed under scalar multiplication, summation and composition, the residual map R is strongly semismooth.

C Proof of Lemma 4.3

As stated in Lemma 4.3, we compute $Z_k = X_k - (\Phi^\star(\gamma_k) + \lambda_1 I)$ and the spectral decomposition of Z_k (cf. Eq. (4.4)) to obtain P_k, Σ_k and the sets of the indices of positive and nonpositive eigenvalues α_k and $\bar{\alpha}_k$. We then compute Ω_k using Σ_k, α_k and $\bar{\alpha}_k$ and finally obtain that $\tilde{P}_k = P_k \otimes P_k$ and $\Gamma_k = \mathrm{diag}(\mathrm{vec}(\Omega_k))$. Thus, we can write the matrix form of $\mathcal{J}_k + \mu_k I$ as

$$J_k + \mu_k I = \begin{pmatrix} \frac{1}{2\lambda_2} Q + \mu_k I & -A \\ \tilde{P}_k \Gamma_k \tilde{P}_k^\top A^\top & \tilde{P}_k((\mu_k + 1)I - \Gamma_k)\tilde{P}_k^\top \end{pmatrix}.$$

For simplicity, we let $W_k = \tilde{P}_k \Gamma_k \tilde{P}_k^\top$ and $D_k = \tilde{P}_k((\mu_k+1)I - \Gamma_k)\tilde{P}_k^\top$. Then, the Schur complement trick implies that

$$\begin{aligned}(J_k + \mu_k I)^{-1} &= \begin{pmatrix} \frac{1}{2\lambda_2} Q + \mu_k I & -A \\ W_k A^\top & D_k \end{pmatrix}^{-1} \\ &= \begin{pmatrix} I & 0 \\ -D_k^{-1} W_k A^\top & I \end{pmatrix} \begin{pmatrix} (\frac{1}{2\lambda_2} Q + \mu_k I + A D_k^{-1} W_k A^\top)^{-1} & 0 \\ 0 & D_k^{-1} \end{pmatrix} \begin{pmatrix} I & A D_k^{-1} \\ 0 & I \end{pmatrix}.\end{aligned}$$

Define $T_k = \tilde{P}_k L_k \tilde{P}_k^\top$ where L_k is a diagonal matrix with $(L_k)_{ii} = \frac{(\Gamma_k)_{ii}}{\mu_k + 1 - (\Gamma_k)_{ii}}$ and $(\Gamma_k)_{ii} \in (0, 1]$ is the i^{th} diagonal entry of Γ_k. By the definition of W_k and D_k, we have $D_k^{-1} = \frac{1}{\mu_k+1}(I + T_k)$ and $D_k^{-1} W = T_k$. Using these two identities, we can further obtain that

$$\begin{aligned}&(J_k + \mu_k I)^{-1} \\ &= \begin{pmatrix} I & 0 \\ -T_k A^\top & I \end{pmatrix} \begin{pmatrix} (\frac{1}{2\lambda_2} Q + \mu_k I + A T_k A^\top)^{-1} & 0 \\ 0 & \frac{1}{\mu_k+1}(I + T_k) \end{pmatrix} \begin{pmatrix} I & \frac{1}{\mu_k+1}(A + A T_k) \\ 0 & I \end{pmatrix}.\end{aligned}$$

This completes the proof.

D Proof of Theorem 4.7

We can see from the scheme of Algorithm 2 that

$$\|R(w_k)\| \leq \|R(v_k)\| \quad \text{for all } k \geq 0,$$

where the iterates $\{v_k\}_{k\geq 0}$ are generated by applying the extragradient (EG) method for solving the min-max optimization problem in Eq. (4.1). We also have that Cai et al. (2022, Theorem 3) guarantees that $\|R(v_k)\| = O(1/\sqrt{k})$. Putting these pieces together yields that

$$\|R(w_k)\| = O(1/\sqrt{k}).$$

This completes the proof.

E Proof of Theorem 4.8

We analyze the convergence property for one-step SSN step as follows,

$$w_{k+1} = w_k + \Delta w_k,$$

where $\mu_k = \theta_k \|R(w_k)\|$ and

$$\|(\mathcal{J}_k + \mu_k \mathcal{I})[\Delta w_k] + R(w_k)\| \leq \tau \min\{1, \kappa \|R(w_k)\| \|\Delta w_k\|\}. \tag{E.1}$$

Since R is strongly smooth (cf. Proposition 4.2), we have

$$\tfrac{\|R(w+\Delta w) - R(w) - \mathcal{J}[\Delta w]\|}{\|\Delta w\|^2} \leq C, \quad \text{as } \Delta w \to 0.$$

Since w_0 is sufficiently close to $w^\star$ with $R(w^\star) = 0$ and the global convergence guarantee holds (cf. Theorem 4.7), we have

$$\|R(w_k + \Delta w_k) - R(w_k) - \mathcal{J}_k[\Delta w_k]\| \leq 2C \|\Delta w_k\|^2.$$

which implies that

$$\|R(w_{k+1})\| = \|R(w_k + \Delta w_k)\| \leq \|R(w_k) + \mathcal{J}_k[\Delta w_k]\| + 2C\|\Delta w_k\|^2. \tag{E.2}$$

Plugging Eq. (E.1) into Eq. (E.2) yields that

$$\begin{aligned} \|R(w_{k+1})\| &\leq 2C\|\Delta w_k\|^2 + \mu_k \|\Delta w_k\| + \tau\kappa \|R(w_k)\| \|\Delta w_k\| \\ &\leq 2C\|\Delta w_k\|^2 + (\theta_k + \tau\kappa)\|R(w_k)\| \|\Delta w_k\|. \end{aligned} \tag{E.3}$$

Since w_0 is sufficiently close to $w^\star$ with $R(w^\star) = 0$ and every element of $\partial R(w^\star)$ is invertible, we have that there exists some $\delta > 0$ such that

$$\|(\mathcal{J}_k + \mu_k \mathcal{I})[\Delta w_k]\| \geq \delta \|\Delta w_k\|.$$

The above equation together with Eq. (E.1) yields that

$$\|\Delta w_k\| \leq \tfrac{1}{\delta} \|(\mathcal{J}_k + \mu_k \mathcal{I})[\Delta w_k]\| \leq \tfrac{1}{\delta} \left(1 + \tau\kappa \|\Delta w_k\|\right) \|R(w_k)\|. \tag{E.4}$$

Plugging Eq. (E.4) into Eq. (E.3) yields that

$$\|R(w_{k+1})\| \leq \|R(w_k)\|^2 \left(\tfrac{2C}{\delta^2} \left(1 + \tau\kappa \|\Delta w_k\|\right)^2 + \tfrac{\theta_k + \tau\kappa}{\delta} \left(1 + \tau\kappa \|\Delta w_k\|\right) \right)$$

Note that $\|\Delta w_k\| \to 0$ and θ_k is bounded. Thus, we have $\|R(w_{k+1})\| = O(\|R(w_k)\|^2)$.

From the above arguments, we see that the quadratic convergence rate can be achieved if Algorithm 2 performs the SSN step when the initial iterate x_0 is sufficiently close to $w^\star$ with $R(w^\star) = 0$. This implies that the safeguarding steps will never affect in local sense where Algorithm 2 generates $\{w_k\}_{k\geq 0}$ by performing the SSN steps only. So Algorithm 2 achieves the local quadratic convergence. This completes the proof.

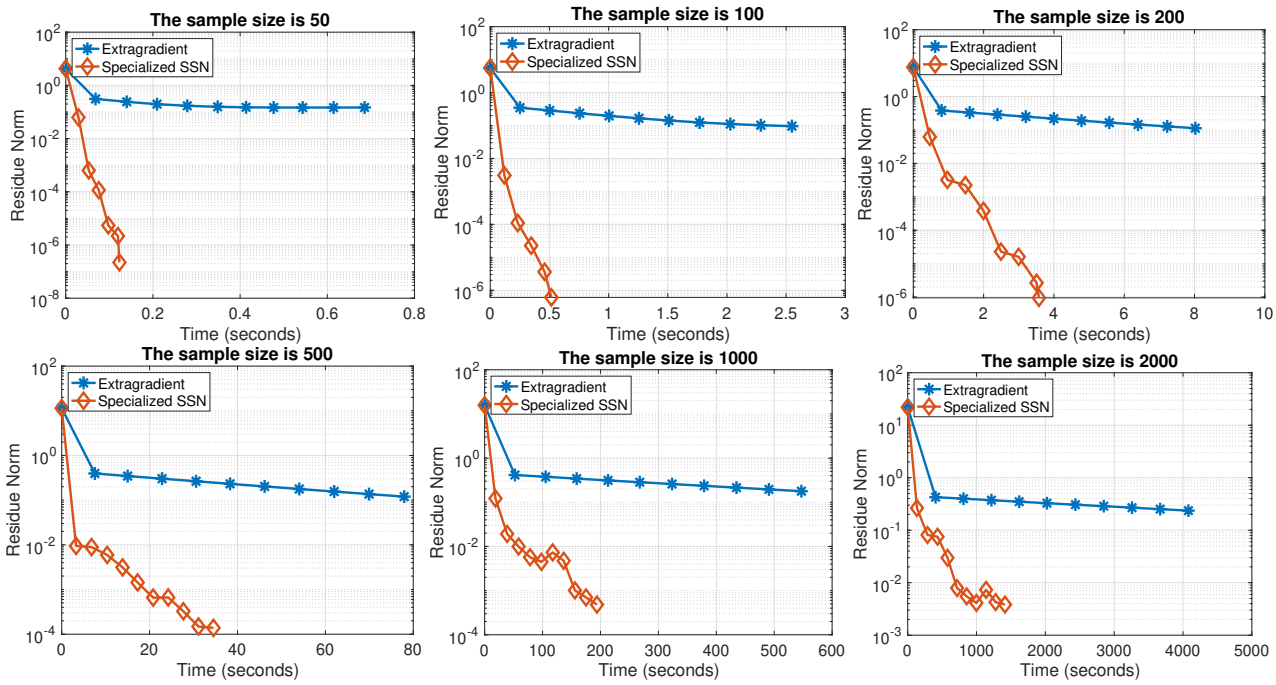

Figure 5: Performance of pure EG and our algorithm for solving kernel-based OT problems with the varying sample size $n \in \{50, 100, 200, 500, 1000, 2000\}$. The numerical results are presented as residue norm v.s. time (seconds).

F Additional Experiments

We compare our method with the pure extragradient (EG) method and summarize the numerical results in Figure 5. In particular, we find that our method consistently outperforms the pure EG method and can output a high-accurate solution in terms of the residue norm. The experimental setup is the same as that used in the main context. Indeed, we fix the dimension $d = 10$ and the bandwidth $\sigma^2 = 0.005$, and vary the sample size $n \in \{50, 100, 200, 500, 1000, 2000\}$. For the EG method, we tune the stepsize and set it as 0.01.

We also describe our setup for the experiment on the real-world 4i datasets from Bunne et al. (2021). Indeed, we draw the unperturbed/perturbed samples for training from 15 cell datasets as follows,

$$x_1, \ldots, x_{n_{\text{sample}}} \sim \mu_{\text{unperturb}}, \quad y_1, \ldots, y_{n_{\text{sample}}} \sim \nu^k_{\text{perturb}} \text{ for } 1 \le k \le 15.$$

where $x_i, y_i \in \mathbb{R}^{48}$ and $\mu_{\text{unperturb}}, \nu^k_{\text{perturb}}$ represent the unperturbed cells and k^{th} perturbed cells. For our algorithm, we generate 256 filling points and compare our method with the default implementation in `OTT` package (Cuturi et al., 2022). Here, the value of entropic parameter is automatically selected by `OTT` package.

Both our algorithm and `OTT` capture the OT map T from training samples. Then, we fix the number of test samples as $m = 200$ and use the OT distance to measure the differences between $\frac{1}{m}\sum_{j=1}^m \delta_{T(\hat{x}_j)}$ and $\frac{1}{m}\sum_{j=1}^m \delta_{\hat{y}_j}$, where $\hat{x}_1, \ldots, \hat{x}_m \sim \mu_{\text{unperturb}}$ and $\hat{y}_1, \ldots, \hat{y}_m \sim \nu^k_{\text{perturb}}$ are unperturbed/perturbed samples for testing. Figure 6 reports the results on 15 single-cell datasets.

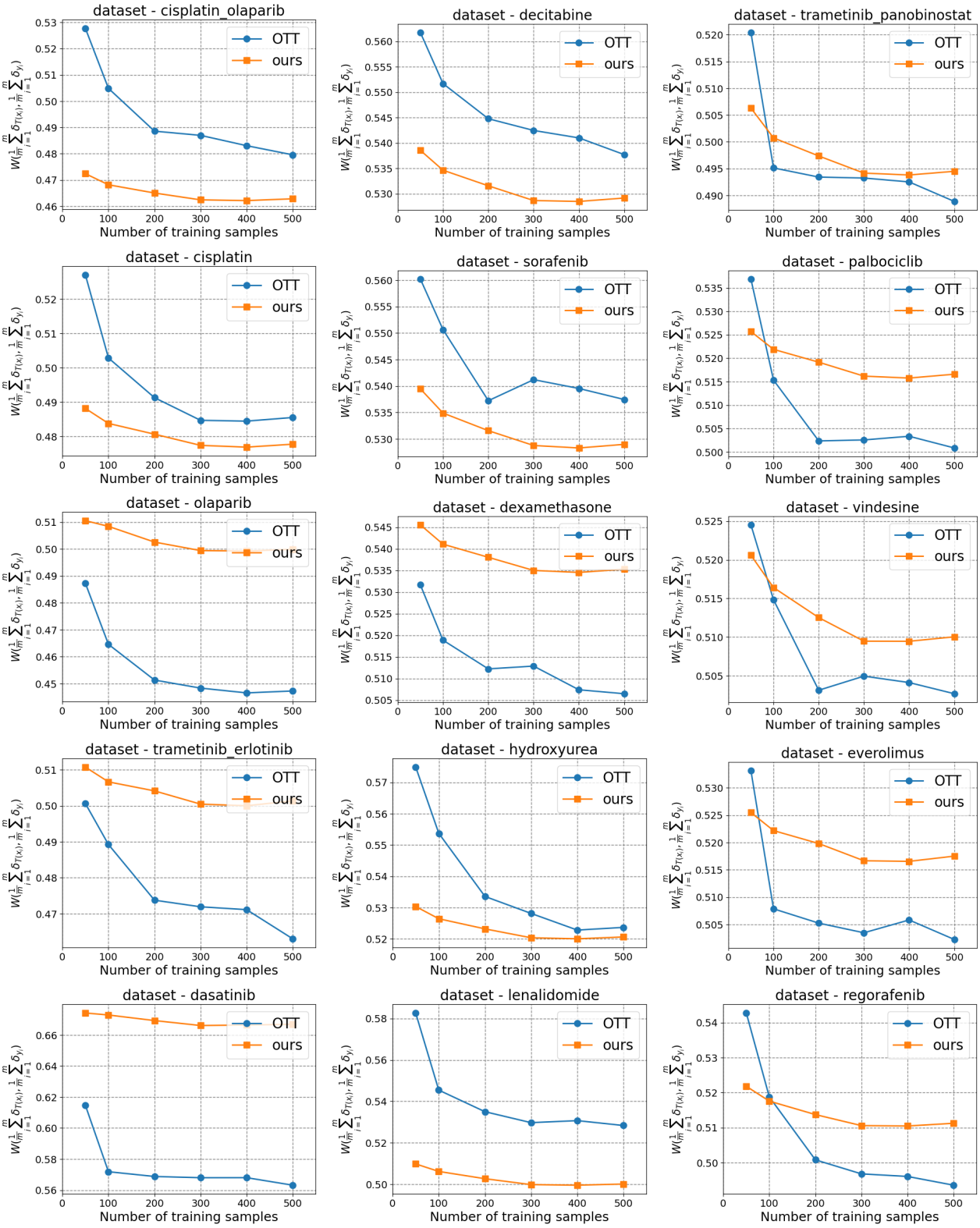

Figure 6: Performance of `OTT` and kernel-based OT estimators computed by our algorithm on all of 15 drug perturbation datasets. X-axis represent the number of training samples and Y-axis represents the error induced by OT map T on test samples in terms of OT distance.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**]
   (b) Complete proofs of all theoretical results. [**Yes**]
   (c) Clear explanations of any assumptions. [**Yes**]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Yes**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]