Frank Wolfe in Probability Space

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a novel Frank-Wolfe (FW) procedure for the optimization of infinite-dimensional functionals of probability measures - a task which arises naturally in a wide range of areas including statistical learning (e.g. variational inference) and artificial intelligence (e.g. generative adversarial networks). Our FW procedure takes advantage of Wasserstein gradient flows and strong duality results recently developed in Distributionally Robust Optimization so that gradient steps (in the Wasserstein space) can be efficiently computed using finite-dimensional, convex optimization methods. We show how to choose the step sizes in order to guarantee exponentially fast iteration convergence, under reasonable assumptions on the functional to optimize. We apply our algorithm to a range of functionals arising from applications in nonparametric estimation.

1 Introduction

Problems in artificial intelligence, statistics, and optimization often find a common root as an infinite dimensional optimization problem in the form

$$\inf \left\{ J\left(\mu\right) : \mu \in \mathcal{P}\left(\mathbb{R}^d\right) \right\}, \tag{1}$$

for the space $\mathcal{P}\left(\mathbb{R}^d\right)$ of Borel probability measures over $\mathbb{R}^d$. In recent years, quantitative statistical and algorithmic treatments of these formulations have produced insights into modern computational methods– resulting in novel approaches to difficult, open problems. Recent works in robust optimization [5, 42, 52, 53], probabilistic fairness [55, 51], reinforcement learning [61, 62], and generative adversarial networks [40, 17, 18] highlight these gains and are linked by the following theme: problems in the form of (1) provide access to rich infinite dimensional structure that sidesteps brittle artifacts of finite dimensional formulations.

This paper provides the construction and analysis of a Frank-Wolfe algorithm for (1) that operates from this infinite dimensional perspective and yields concrete convergence and complexity guarantees for a sub-family of problems (1) which are well-behaved with respect to the Wasserstein distance of order 2 (see Algorithm 1 and Theorem 1). Under canonical conditions of smoothness and convexity we recover linear rates of convergence while, even for functionals which exhibit low degrees of smoothness and for conditions that go beyond convexity, we recover sublinear rates that are to be expected from finite dimensional analogues [34] (see Section 2.2).

Indeed, development of our Frank-Wolfe method is inspired by efforts in DRO [42, 5, 24, 52] which have considered variants of (1) in the form

$$\sup \left\{ \int f \, d\mu : D_c(\mu, \mu_0) \leq \delta \right\}, \tag{2}$$

where $D_c(\mu, \mu_0)$ is the optimal transport cost between μ and μ_0 (a reference measure) under some cost function c. The form of (2), itself, immediately suggests the basis of an infinite dimensional Frank-Wolfe procedure since it provides a "linear" objective subject to a local, "trust-region" constraint–

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

centered at μ_0. The relevance of (2) in distributionally robust optimization (DRO) and mathematical finance has resulted in a multitude of computational schemes [42, 32, 35, 52, 59] for solving (2). However, hitherto, such works have failed to consider (2) within the context of a general variational method for (1). What makes these efforts notable, within the scope of this work, is that they emphasize that the solution of (2) can be highly non-trivial. Indeed, without particular assumptions, (2) can disguise an computationally hard problem– despite being convex in a Banach sense on $\mathcal{P}(\mathbb{R}^d)$. Even in the case where the cost is the squared Euclidean norm $c(x, y) = \|x - y\|^2$ (the case of primary concern for this work), computational trouble can lie dormant– an artifact of inherently difficult problems in unconstrained optimization [13]. This should not be surprising, however, given specters of computational hardness dating back to early formulations of DRO [21]. To resolve these issues, we also provide a novel analysis of techniques from distributionally robust optimization (DRO) which illustrate how such formulations can be used, in a computationally tractable way, to construct first-order, variational methods.

1.1 Previous work

Distributionally robust optimization To navigate such pitfalls, one can consider particular instances of (2) where the objective and constraints are sufficiently structured to preclude computational intractability and permit solution via methods adapted to the provided structure. Early work with this line [26, 21, 58], has recently been supplemented by approaches [12, 25, 6, 44, 35, 65, 33] which focus directly on DRO formulations from particular contexts in machine learning and operations research. Unfortunately, the techniques offered by these efforts require assumptions which are too restrictive for this work. These assumptions typically relate to a specific form for the objective function or constraints in (2) (e.g. linear/convex functions/piecewise-convex objectives or constraints with support or density requirements, see [30, 64, 42, 32, 63, 4, 56] for additional examples). In this instance, such limitations preclude their applicability since, in general, a "gradient object" for a functional J (see Section 2) need not satisfy these conditions.

A second, more relevant, approach to compute DRO problems (2) is to restrict the level of robustness δ for which the problem is solved. This technique has been used in works such as [5, 52] and we apply this principle in a similar spirit to [52]. In that work, smoothness of the objective in (2) is used to, qualitatively, argue that a sufficiently small δ provides a computationally-tractable optimization problem. In contrast, we provide quantification of the level of robustness required to achieve this tractability and we demonstrate that this level robustness is sufficiently large to achieve canonical convergence rates for an infinite-dimensional Frank-Wolfe algorithm.

Variational methods Although formulation of a Frank-Wolfe method for (1) (with quantitative bounds on complexity and convergence) has not appeared in previous literature, certain, tangentially-related variational methods offer conceptual similarity. The first of these methods is [39], which draws similar inspiration from finite-dimensional Frank-Wolfe procedures. However, [39] narrowly considers the Sinkhorn barycenter problem and utilizes a different Frank-Wolfe constraint which is adapted to the Banach space structure of $\mathcal{P}(\mathbb{R}^d)$. The second, more closely related effort, is [37] where similar, infinite-dimensional conditions (Section 2.2) are used to study particle-based methods for computing Nash equilibria of zero-sum games. While the setting and procedures developed in this work are completely different, we note that, as a special case, our Frank-Wolfe method can also produce a particle-based optimization procedure. This hints at possible insightful connections with other particle methods [38, 23, 10, 9] which are left for future consideration.

2 Wasserstein geometry

This work considers the problem

$$\min_{\nu \in \mathcal{P}_2(\mathbb{R}^d)} J(\nu) \tag{3}$$

for functionals $J : \mathcal{P}_2(\mathbb{R}^d) \to \bar{\mathbb{R}}$ over the subset of Borel measures $\mathcal{P}(\mathbb{R}^d)$ defined by

$$\mathcal{P}_2(\mathbb{R}^d) := \left\{ \mu \in \mathcal{P}(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|x\|^2 \, d\mu(x) < \infty \right\} \tag{4}$$

In particular, we consider functionals J that are differentiable (Definition 1) with respect to Wasserstein distance of order 2

$$\mathcal{W}^2(\mu, \nu) := \inf_{\gamma \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\gamma(x, y) \tag{5}$$

where $\Pi(\mu, \nu)$ is the set of all joint couplings with marginals $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$. Common examples of functionals that can be cast within this framework are as follows.

Example 1 (Divergences)**.** For any convex, lower-semicontinuous function $f : \mathbb{R}_+ \to \mathbb{R}$ such that $f(1) = 0$, one can consider a "f-divergence" of the form

$$J(\mu) := D_f(\mu||\nu) = \int_{\mathbb{R}^d} f\left(\frac{d\mu}{d\nu}\right) d\nu. \tag{6}$$

For instance, if $f(t) = t \log t$, (6) reduces to Kullback-Leibler divergence D_{KL}.

Example 2 (Integral Probability Metrics)**.** For a set of real valued functions F on $\mathbb{R}^d$ one can define the discrepancy

$$J(\mu) := \text{IPM}\,(\mu, \nu) = \sup_{f \in F} \int_{\mathbb{R}^d} f\, d\mu - \int_{\mathbb{R}^d} f\, d\nu \tag{7}$$

for $\mu, \nu \in \mathcal{P}\left(\mathbb{R}^d\right)$, where ν is a fixed, reference measure. Such discrepancies are termed Integral Probability Metrics (IPMs), although they may not strictly satisfy the requirements of a metric– say, by failing to distinguish all pairs of measures. Instead, for a pair of measures $\mu, \nu \in \mathcal{P}\left(\mathbb{R}^d\right)$, IPMs can be interpreted as measuring the extent to which μ and ν differ on functions in F– or, rather, measuring the extent to which μ and ν can be distinguished by F. Concretely, consider $F = \{f : \|f\|_H \leq 1\}$ where H is a reproducing kernel hilbert space (RKHS). In this case, one obtains the dual formulation of Maximum Mean Discrepancy (MMD).

2.1 Properties and differentiability for Wasserstein space

Under the Wasserstein distance, $\mathcal{P}_2(\mathbb{R}^d)$ is a Polish space [57] and, more importantly, it is a *geodesic* space. That is, for every $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, there exists a constant-speed geodesic curve $\mu_t : [0, 1] \to \mathcal{P}_2(\mathbb{R}^d)$ where $\mu_0 = \mu$, $\mu_1 = \nu$ and

$$\mathcal{W}(\mu_t, \mu_s) = |t - s|\mathcal{W}(\mu_0, \mu_1) \tag{8}$$

Moreover, there is a bijection between constant-speed geodesics and optimal transport plans [1, Theorem 7.2.2]. Every geodesic corresponds to a unique, optimal transport plan $\gamma \in \Pi\,(\mu, \nu)$ such that

$$\mu_t = ((1-t)x + ty)_{\#}\,\gamma \qquad \text{where} \quad \mathcal{W}^2(\mu, \nu) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \; d\gamma(x, y), \tag{9}$$

and μ_t is the distribution of the random variable $(1-t)X + tY$ with the pair (X, Y) following distribution γ. Conversely, every optimal transport plan gives rise to a unique geodesic via (9).

Since our Frank-Wolfe method minimizes a sequence of linear approximations, one must define the notion of a gradient (of a functional J) to be compatible with respect to the geometry of these geodesics. In particular, as $\mathcal{P}_2(\mathbb{R}^d)$ is curved under $\mathcal{W}$ (see Appendix A), gradients must be defined in terms of a selections in an appropriate cotangent bundle. For Wasserstein space, this cotangent bundle (denoted $\text{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$) is essentially the set of vector fields on $\mathbb{R}^d$ that can be approximated by gradients of smooth functions (see Appendix A for details). This results in the following definition.

Definition 1 (Wasserstein differentiability)**.** Let S be a geodesically convex set; that is, $\mu_t \in S$ for any geodesic μ_t between $\mu, \nu \in S$. A functional $J : \mathcal{P}_2(\mathbb{R}^d) \to \bar{\mathbb{R}}$ is Wasserstein differentiable on S if there is a map $F : \mathcal{P}_2(\mathbb{R}^d) \to \text{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$ such that for all $\mu, \nu \in S$ and a geodesic $\mu_t : [0, 1] \to \mathcal{P}_2\left(\mathbb{R}^d\right)$ between μ and ν, one has

$$\lim_{\alpha \to 0^+} \frac{J\left(\mu_\alpha\right) - J(\mu)}{\alpha} = \int_{\mathbb{R}^d \times \mathbb{R}^d} F(\mu; x)^T (y - x)\, d\gamma(x, y), \tag{10}$$

where γ is the unique optimal transport plan (9) corresponding to μ_t. Note that $F(\mu; x) = \left(F(\mu)\right)(x)$ provides a aesthetic way of representing the evaluation at $x \in \mathbb{R}^d$ of the output of F at μ. The map F is called the Wasserstein derivative of J.

Remark 1. The description of differentiability provided by Definition 1 falls within the general framework of metric derivatives and Wasserstein gradient flows, largely codified in [1]. This framework is now a well-established component of the theory of Wasserstein spaces, while the relation (10), itself, presents only a narrow structuring of ideas from this framework. Definition 1, however, is often how works in statistical and algorithmic fields interact with this broader area [54, 15, 36, 37]. Moreover,

this literature demonstrates the most motivating feature of (10): a large number of functionals of interest for machine learning and statistical inference exhibit Wasserstein gradients in the sense of (10). The curious reader is referred to [1, 50, 8] for precise statements of conditions under which (10) is guaranteed. However, F is intimately relative to the Gateaux differential for J [50, 54]. Recall, the Gateaux differential for a functional J exists when there is an appropriate, dual space D^* on a closed subspace $D \subseteq \mathcal{P}(\mathbb{R}^d)$ such that

$$\langle dJ(\mu), \nu - \mu \rangle = \lim_{\alpha \to 0^+} \frac{J(\mu + \alpha(\nu - \mu)) - J(\mu)}{\alpha} \tag{11}$$

for some $dJ(\mu) \in D^*$ and all μ in some set S such that $S - S \subseteq D$. In instances where the Gauteaux differential $dJ(\mu)$ exists, the Wasserstein derivative F will usually exist [37] and be given by $\nabla dJ(\mu)$. Here, we use the finite dimensional gradient operator $\nabla(\cdot)$ formally, and omit a rigorous exposition on this operation in the context of $\mathrm{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$.

It should be noted that computation of the Wasserstein derivative might be difficult. Indeed, for a J in a variational form such as (7), computation of the Wasserstein derivative is equivalent to finding a *witness* function that achieves the supremum [50]. In the case of a pathological sets (in (7)), such a task might be intractable. To resolve this issue, and to simplify our treatment, this work utilizes the existence of an oracle for the computation of a Wasserstein gradient. This oracle permits a unified description of our Frank-Wolfe algorithm and abstracts away variation in functional-specific computational cost. Recall that a function in the Hölder space $C^1(\mathbb{R}^d)$ is called L-smooth if has L-Lipschitz gradients.

Definition 2 (Wasserstein Derivative Oracle)**.** Let $J : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ be a Wasserstein differentiable functional on a set S with Wasserstein derivative $F : \mathcal{P}_2(\mathbb{R}^d) \to \mathrm{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$. A L-smooth Wasserstein derivative oracle over S is an oracle which, given sample access to a distribution $\mu \in S$ and an error parameter ϵ, returns an L-smooth function $\widehat{\phi}_\mu \in C^1(\mathbb{R}^d)$ satisfying

$$\left\| \nabla \widehat{\phi}_\mu - F(\mu) \right\|_{L^2(\mu)} \leq \epsilon \tag{12}$$

where $\|\cdot\|_{L^2(\mu)}$ is the canonical norm on the space $L^2(\mu)$ of square integrable functions with respect to $\mu \in \mathcal{P}(\mathbb{R}^d)$. In this work, the output of this oracle is represented as $\Theta(\mu, \epsilon)$.

Remark 2. The qualification that the Wasserstein derivative oracle return an L-smooth function is necessary to exclude the, aforementioned, possibility of a pathological Wasserstein derivative– that would be intractable for use in a computational procedure. In some ways, this is representative of the fact that the cotangent space $\mathrm{CoTan}(\mu)$ at a point μ is too large. Such a condition is common in other variational methods [3, 18, 62, 19] and is relatively superficial– when coupled with the degree of approximation afforded by ϵ. Via smoothing techniques [49, 10, 36], functionals can often be assumed to have Wasserstein derivatives which are $C^1(\mathbb{R}^d)$ or are well-approximable by $C^1(\mathbb{R}^d)$ functions.

2.2 Smoothness and Łojasiewicz inequalities

In finite dimensions, iterative, gradient-based methods typically require the specification of two conditions in order to achieve convergence.

- The accuracy of local, linear approximations that are provided by the gradient.
- The extent to which local descent makes global progress on the objective.

Here, we state these conditions in the context of functionals over Wasserstein space.

Definition 3 (α-Hölder smoothness)**.** Let S be a geodesically convex set and let $J : \mathcal{P}_2\left(\mathbb{R}^d\right) \to \mathbb{R}$ be a functional which is continuously Wasserstein differentiable on the set S. J is said to be locally α-Holder smooth on S with parameters T and Δ if for all $\mu \in S$ and all $\nu \in S$ such that $W(\mu, \nu) \leq \Delta$, there exists an optimal transport plan $\gamma \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d)$ such that

$$J(\nu) \leq J(\mu) + \int_{\mathbb{R}^d \times \mathbb{R}^d} F(\mu; x)^T (y - x)\, d\gamma(x, y) + \frac{T}{1+\alpha} \mathcal{W}^{1+\alpha}(\nu, \mu) \tag{13}$$

Definition 4 (Łojasiewicz inequality)**.** A Wasserstein differentiable functional J on a set $S \subseteq \mathcal{P}_2(\mathbb{R}^d)$ is said to satisfy a *Łojasiewicz inequality* with parameter τ and exponent θ if for all $\mu \in S$ and $J_* := \inf_{\mu \in S} J(\mu)$

$$\tau \left(J(\mu) - J_*\right)^\theta \leq \|F(\mu)\|_{L^2(\mu)} \tag{14}$$

where F is the Wasserstein derivative (10) of J.

Remark 3. More restrictive versions of both (13) and (14) commonly appear in previous literature [3, 31, 37, 18, 14]. In most cases, the α-Hölder smoothness condition (13) is stated for $\alpha = 1$ and required to hold globally ($\Delta = \infty$). This smoothness criterion is considerably weaker since it requires that the Wasserstein gradient only provide a local approximation that is slightly more than first-order accurate. Moreover, such a condition can be necessary when the Wasserstein derivative (10) is not Lipschitz with respect to $\mathcal{W}$ in the cotangent space norm on $\mathrm{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$– see [41] for such an example. Additionally, statement of the Łojasiewicz inequality (14) is broader than canonical treatments due to the presence of the auxiliary power θ. Most often, the specific instances of either $\theta = 1/2$ or $\theta = 1$ are considered. The case $\theta = 1$ is implied for (geodesically) convex functionals J with a $\mathcal{W}$-bounded level set, while $\theta = 1/2$ is implied for strongly convex J [1, 31].

3 Frank-Wolfe algorithm

Algorithm 1 provides our Frank-Wolfe procedure along with its associated convergence guarantees and sample complexities in Theorem 1 and Proposition 1. To obtain these guarantees, we require the

Algorithm 1 Frank Wolfe for (3)

Input: Wasserstein derivative oracle Θ, initial distribution μ_0, smoothness parameter α, gradient error $\hat{\epsilon}$, estimation error $\bar{\epsilon}$, iterate error $\widetilde{\epsilon}$, stopping threshold r, step sizes $(\beta_1, \beta_2, \beta_3)$, number of iterations k

for $1 \leq i \leq k$ **do**

Let $\widehat{\phi}_{\mu_{i-1}} \leftarrow \Theta(\mu_{i-1}, \hat{\epsilon})$ (▷) Definition 2

Compute $\left\|\nabla \widehat{\phi}_{\mu_{i-1}}\right\|_{L^2(\mu_{i-1})} - \bar{\epsilon} \leq s \leq \left\|\nabla \widehat{\phi}_{\mu_{i-1}}\right\|_{L^2(\mu_{i-1})}$

if $s \leq r$, **then break**

else $\delta \leftarrow \min\left(\beta_1, \beta_2 s, \beta_3 s^{1/\alpha}\right),\ \zeta \leftarrow \delta\widetilde{\epsilon}$

Compute μ_i satisfying $W(\mu_i, \mu_{i-1}) \leq \delta$ and (▷) Proposition 1

$$\int \widehat{\phi}_{\mu_{i-1}}\, d\mu_i - \inf_{W(\nu, \mu_{i-1}) \leq \delta} \int \widehat{\phi}_{\mu_{i-1}}\, d\nu \leq \zeta \tag{15}$$

return μ_i

following assumptions–phrased in the language from Section 2.

Assumption 1 (Smoothness assumption)**.** The functional $J : \mathcal{P}_2(\mathbb{R}^d) \to \bar{\mathbb{R}}$ is Wasserstein differentiable (Definition 1) and locally *α-Holder smooth* (Definition 3) on a set $S \subseteq \mathcal{P}_2(\mathbb{R}^d)$ with parameters T and $\Delta_1 > 0$ (Definition 3). Further, an L-smooth Wasserstein derivative oracle (Definition 2) for J exists.

Assumption 2 (Local richness)**.** The set S is rich enough to contain the solution to (2) for $\mu \in S$, L-smooth $-f$, and $\delta \leq \Delta_2$.

Assumption 3 (Łojasiewicz assumption)**.** The functional J satisfies a *Łojasiewicz inequality* (14) on $S \subseteq \mathcal{P}_2(\mathbb{R}^d)$ with parameters $\tau > 0$ and θ.

Theorem 1. *Under Assumptions 1, 2, 3, and an appropriate choice of input parameters, Algorithm 1 computes a distribution μ^* satisfying*

$$r(\mu^*) := J(\mu^*) - \inf_{\mu \in S} J(\mu) \leq \epsilon \tag{16}$$

in at most

$$k = \widetilde{O}\left(\epsilon^{-p_-}\right) \tag{17}$$

iterations, where p_- denotes the negative part of $p = 1 - \alpha^*\theta$ for the dual exponent $\alpha^* = (1+\alpha)/\alpha$. The notation $\widetilde{O}(\cdot)$ omits logarithmic factors in it's arguments.

Remark 4. In the case of a (geodescially) strongly-convex and 1-Hölder smooth functional J, the Łojasiewicz inequality (14) holds with $\theta = 1/2$ and one obtains standard $\widetilde{O}(1)$ complexity (in terms of ϵ). This is to be expected from finite dimensional analogues [34]. Similarly, for J which is only convex (with a $\mathcal{W}$-bounded level set), (14) holds with $\theta = 1$ and (17) yields a $\widetilde{O}(\epsilon^{-1})$ complexity that mimics canonical results. The step size required to achieve these complexities is illustrated by the choice of δ in Algorithm 1.

For functionals which are α-Hölder smooth for $\alpha < 1$, the dependence on the dual exponent α^* in (17) can be rather punishing for small α. It is natural to ask if this exponent could be improved within the scope of Assumptions 1, 2, and 3. Moreover, in finite dimensions, it is well known that first-order methods for convex and α-Hölder smooth functions (also known as weakly smooth functions) can obtain ϵ-optimal solutions in $O(\epsilon^{-2/(1+3\alpha)})$ iterations [45]. Hence, it could even be considered whether, given geodesic-convexity assumptions on J, a better iteration complexity for Algorithm 1 would be obtainable.

We conjecture that such improvements are unlikely - particularly those that would draw on analogy from finite dimensional techniques. The motivation for this is as follows. A common approach to establishing improved iteration complexities for convex, α-Hölder smooth functions, in finite dimensions is to consider their gradient oracles as inexact oracles for convex, 1-Hölder smooth functions [22]. Using either averaging arguments or accelerated methods, more rapid progress on an underlying objective can then be made with these inexact oracles. Our Frank-Wolfe method already utilizes an inexact step (15) so, conceptually, such an approach could be applied to Algorithm 1. Unfortunately, however, such finite dimensional analogies fail due to the difficulty of averaging distributions in Wasserstein space. Indeed, in finite dimensions, averaging is crucial to prevent error accumulation from outpacing objective progress. Since Wasserstein space is positively curved (Appendix A) computing analogous convex combinations of the μ_i in Algorithm 1 is, itself, a variational problem that might be as expensive to compute as the original problem (3).

3.1 Computation of the Frank-Wolfe step

An algorithm for computation of the Frank Wolfe step (15) is given in Appendix C and the net result of this procedure is as follows.

Proposition 1. *Under the assumptions of Theorem 1, there exists a stochastic algorithm which (with high probability) provides sample access to a distribution satisfying* (15)*. Moreover, in the setting of* (17)*, this algorithm requires* $\widetilde{O}\left(\epsilon^{-2\alpha^*\theta}\right)$ *samples and gradient evaluations.*

Crucially, the property enabling Proposition 1 is *duality*. The Frank-Wolfe problem (15) exhibits a dual of the form

$$\sup_{\lambda \geq 0} \left[\int_{\mathbb{R}^d} \left(\inf_{y \in \mathbb{R}^d} f(y) + \frac{\lambda}{2} \|x - y\|^2 \right) d\mu(x) - \frac{\delta^2 \lambda}{2} \right] \tag{18}$$

and (18) is amenable to computation using techniques from finite dimensional optimization. This approach derives from distributionally robust optimization [5, 52, 24, 32] where such techniques have been used to produce methods in optimization and machine learning that are robust to adversarial perturbations.

In general, solution of (15) for any level of δ could be computationally hard [5, 52]. However, since a Frank-Wolfe procedure need only solve *local* problems, not *global* ones, we show: there is a δ which is, simultaneously, small enough to enable the efficient computation of (18)), yet large enough to produce (17). The techniques used to achieve these results most closely resemble ideas from [52]. However, we provide a precise quantification of the δ in (15) that is required to achieve computation tractability and develop a procedure which yields guarantees on the primal-dual gap of (15) and (18). In turn, the theoretical insights that we obtain suggest an empirical procedure in which λ in (18) can be updated relatively infrequently within our Frank-Wolfe algorithm, provided that it is chosen on a proper scale. Such an implementation is investigated in Section 4 and yields significant computational savings.

It is also worth noting that the implementation in Appendix C requires only sample access to μ_0 in order to provide sample access to a distribution satisfying (16). Thus, all operations in Algorithm 1 can be implemented using only sample access to μ_0. Practically, however, it is often more efficient to maintain approximations to the iterates μ_i via a non-parametric estimator. When this is done, it

results in an additional, additive error in the residual (16) at each step of Algorithm 1. If this error is on the order of the error produced by the Wasserstein derivative oracle Θ, the iteration complexity (17) remains unaffected. Moreover, analysis of the error induced by a particular non-parametric approximation of the μ_i is highly problem dependent– so we do not consider it in the context of Theorem 1.

4 Computational examples

In this section, we demonstrate the application our Frank-Wolfe algorithm to several non-parametric estimation problems in statistics and machine learning. All simulations are implemented using Python 3.8 on a high performance computing server running Ubuntu 18.04 with a Gen10 Quad Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz processor.

4.1 Gaussian deconvolution

A classical task in nonparametric statistics [11, 7] is to infer a latent, data-generating distribution $\mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$ from a set of observations that are corrupted by independent, additive Gaussian noise. For observations $Y_1, \ldots, Y_n$ such that $Y_i = X_i + Z_i$ where $X_i \sim \nu, Z_i \sim N\left(0, \sigma^2\right)$. One seeks to compute a non-parametric estimate of μ — the variance of the noise σ^2 is considered known. Since Z_i is independent of X_i, this task amounts to "deconvolving" μ from the distribution of Z_i. A natural candidate for μ is the maximum-likelihood estimator (MLE),

$$\widehat{\mu} := \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\arg\max} \ \sum_{i=1}^{n} \log \int_{\mathbb{R}^d} g_\sigma\left(Y_i - x\right) \, d\mu(x) \tag{19}$$

where g_σ is the density of Z_i. We refer the reader to [48, Section 3] for further details but note that it was shown in [48] that $\widehat{\mu}$ has an equivalent characterization

$$\hat{\mu} = \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\arg\min} \ D_{\sigma^2}(\mu, \hat{P}_Y) \tag{20}$$

where

$$D_{\sigma^2}(\mu_1, \mu_2) := \inf_{\pi \in \Pi(\mu_1, \mu_2)} \frac{1}{2} \int \|x - y\|^2 \, d\pi(x, y) + \sigma^2 D_{KL}(\pi \,||\, \mu_1 \otimes \mu_2) \tag{21}$$

is the entropic optimal transportation distance [20] and $\hat{P}_Y$ is the empirical distribution of the Y_i. The problem (20) readily lies within the framework of (3) for $J(\mu) := D_{\sigma^2}(\mu, \hat{P}_Y)$. Moreover, it is known [39] that the Wasserstein derivative (10) of $D_{\sigma^2}(\mu, \hat{P}_Y)$ with respect μ is given by

$$\nabla \hat{\phi}_\mu(x) = \sigma^2 \log \left(\frac{1}{n} \sum_{i=1}^{n} \exp\left(\left(v_i^* - \|x - y_i\|^2 / 2 \right) / \sigma^2 \right) \right) \tag{22}$$

where $v^* \in \mathbb{R}^n$ is dual variable (corresponding to $\hat{P}_Y$) which is optimal for $D_{\sigma^2}(\mu, \hat{P}_Y)$. This provides a Wasserstein derivative oracle for (20) as the vector v^* can be readily approximated using sinkhorn or stochastic gradient algorithms [47].

Toy example On 2D Gaussian mixture A simple, two dimensional instance of (19) is shown in Figure 1 on a dataset Y_i of 50 samples with mixture of 4 Gaussians — illustrated by the kernel density estimator of the Y_i, shown in red. The behavior of our Frank-Wolfe Algorithm is depicted over the course of several iterations, where the foreground contours provide the density of the iterate, μ_i, that is maintained by the algorithm. It can be easily observed that despite the small overlap between our initial distribution and target one, our method reaches the global optima very quickly.

Uniform strategy for λ and sensitivity analysis on high-dimensional examples Since the number of bisection ascent step for λ in (18) that arises during the search can be large, the original algorithm may be computationally rather demanding for high-dimensional cases. Thus, we are motivated to develop an approach that makes λ as a constant which only depends on the step size $\lambda = \varepsilon / \delta^{1/2}$ without other one-dimensional optimization methods as an inner solver, where ε is the scaling factor. Figure 2(a) provides a convergence behavior comparison between the vanilla Algorithm 1 and the modification leveraging this new uniform strategy. Not surprisingly, the modified

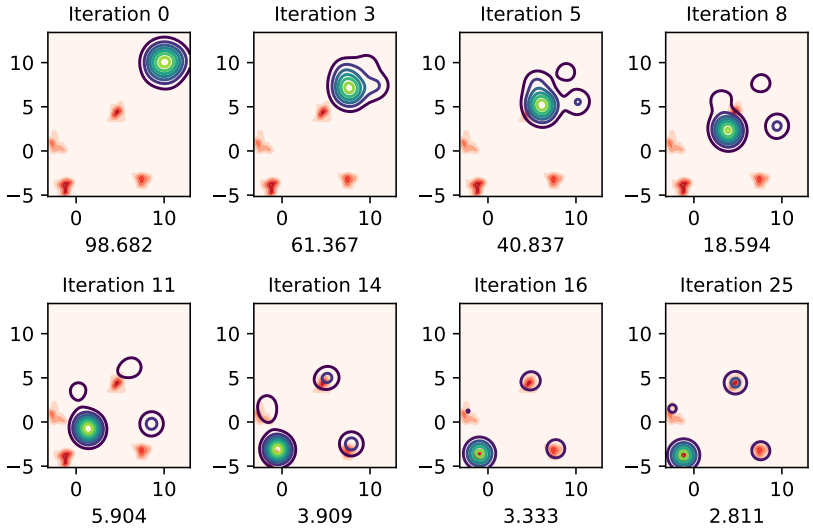
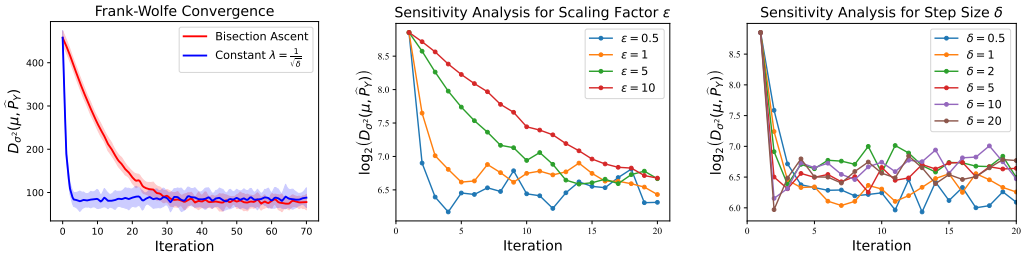


Figure 1: Toy Example on a 2D Gaussian Mixture. Note that the initial distribution is set as $N([10, 10], \sigma^2\mathbf{I})$ for $\sigma^2 = 0.4$ and the number of particles is 200. The bisection method of Appendix C is used with tolerance set to $1e^{-3}$. The objective value $D_{\sigma^2}(\mu_i, \hat{P}_y)$ is below to each sub-figure.



(a) Objective Value (b) Scaling Factor ε (c) Step size δ

Figure 2: High-dimensional Gaussian deconvolution for $d = 64$. For 50 data points, Y_i, sampled from a mixture of 7-Gaussians, 200 particles are used in a non-parametric estimate the μ_i. In Figure 2(a), the tolerance for the ascent method is $1e^{-3}$ and the shaded bands show the standard derivation over 10 independent runs with random initializations. In Figure 2(b), the step size δ is 0.5.

Frank-Wolfe algorithm can get to the local region around the global optima faster but with a larger variance, as the uniform strategy is indeed a more aggressive strategy at the early stage. To further support the uniform strategy and our Frank-Wolfe framework, we conduct extensive sensitivity experiments on the hyperparameter (e.g., step size δ and scaling factor ε). Both Figure 2(b) and 2(c) demonstrate that our algorithm is robust to these crucial hyperparamter. We can also observe that it is better not to choose a relatively large step size although we can converge faster at the beginning but suffer from the risk of divergence, as the maximum step size is controlled by the smooth parameter in theory. Hence, our experiment results here also corroborate the theoretical findings.

It is worth mentioning that this new uniform strategy can make our Frank-Wolfe framework be extended to an asynchronous decentralized parallel setting easily and thus can further meet the requirements of large-scale applications. Based on the superior performance, we left its rigorous convergence analysis as an open question.

4.2 Nonparametric learning with student-teacher networks

The rise of generative adversarial networks (GANs) [27] and efforts connecting neural networks and kernel regression [16], have generated interest in maximum mean discrepancy (MMD), particularly with respect to it's role in constructing high-dimensional, distributional embeddings [19, 43]. This development is predicated on the observation that any neural network $(x, \theta) \to \psi(x, \theta)$, which

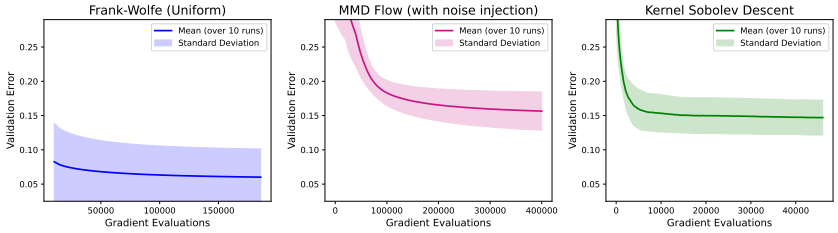


Figure 3: Student-Teacher Network; The detailed implementation set up is same as [3, Appendix G]. The left one is the result for our Frank-Wolfe method with the uniform strategy $\lambda = \frac{0.05}{\delta}$ and the step size δ is 0.5. The number of particle is 200.

produces an output $\psi(x, \theta) \in \mathbb{R}^d$ from input data $x \in X \subseteq \mathbb{R}^d$ and parameters $\theta \in \Theta \subseteq \mathbb{R}^m$, yields a kernel on the parameter set Θ:

$$k(\theta_1, \theta_2) := \mathbb{E}_x \left[\psi(x, \theta_1)^T \psi(x, \theta_2)\right] \tag{23}$$

where the expectation over x is taken with respect to a data generating distribution. Via MMD, $k(\cdot, \cdot)$ induces a natural discrepancy measure between distributions over network parameters θ. Thus, learning of a generative image model can be expressed as minimizing MMD with respect to latent, generative distribution for ν. We refer to [43, 3] for further descriptions of these applications.

Being an integral probability metric (7), squared MMD lies well within the framework of this paper

$$J(\mu) := \mathrm{MMD}^2(\mu, \nu). \tag{24}$$

and the Wasserstein derivative (Definition 1) of J admits a natural expression [3] as the difference between the mean embeddings of μ and ν

$$f_\mu^*(x) = \mathbb{E}_{z\sim\mu}\left[k(z, x)\right] - \mathbb{E}_{z\sim\nu}\left[k(z, x)\right] \tag{25}$$

Indeed, f_μ^* can be readily computed via sampling methods– even when μ or ν are continuous or are large, discrete distributions [28]. Perhaps the most advantageous consequence of (25), however, is that the Wasserstein gradient directly inherits regularity present in k. Indeed, should $\nabla_x k(x, y)$ be L-Lipschitz in x (uniformly for all y), J (24) is naturally L-smooth [3]. This has led to the development of several variational or particle-based methods for minimizing (24) [3, 43, 19].

Figure 3 demonstrates the competitive performance of our method with two of baselines showcased in [3] on Student-Teacher network problem. Our method is shown on the left, the center plot shows the "MMD gradient flow" algorithm from [3], and the right plot provides the "Sobolev Descent" algorithm of [43]. Performance is evaluated in terms of MMD error on a validation dataset and is shown as a function of the total gradient evaluations performed by each method. This provides a better proxy for relative performance and convergence since an iteration of Algorithm 1 performs multiple solves that are, each, similar in terms of gradient complexity to a single iteration of MMD gradient flow or Sobolev descent. Further, the total number of gradient evaluations should not be viewed as a proxy for wall-time as, for each gradient evaluation, the number of operations performed by each method can vary widely. Indeed, for each gradient evaluation in Sobolev descent an entire linear system solve is performed, which is computational demanding in practice. Also, note that, as both MMD gradient flow and Sobolev descent are particle-based, Algorithm 1 was, for the purposes of comparison, instantiated with a particle distribution of equal size.

5 Conclusion

This paper introduces and studies a Frank-Wolfe procedure for the minimization of functionals of probability measures. While these methods have been widely studied in the finite-dimensional setting; our current environment presents both significant benefits and opportunities. First, many problems of interest can be posed in terms of the types of formulations that we study [17, 18, 5, 52, 19, 37, 62, 53, 55, 10]. Second, our algorithm can naturally be asynchronously parallelized. This is a research avenue of significant promise that we plan to explore in future work, especially in connection with the wide range of applications mentioned earlier.

Broader impact

As mentioned in the body of the paper the methods and problems that we study (i.e. Frank-Wolfe optimization for functionals of probability measures in infinite dimensions) can be applied to a wide

range of machine learning and data science problems (e.g. comparing probability models, generative networks and decision making under uncertainty - among others, [51, 3, 17, 18, 43, 40, 42, 61]). While our contribution has a theoretical orientation, we believe that the structure of our algorithm holds significant promise in the types of problems mentioned earlier. Therefore, we expect this work to significantly impact virtually any area in which data-intensive analytical methods play a significant role.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] The contributions of this work are mainly theoretical. We do not foresee any negative societal impact as a consequence.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See the supplemental material
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results? [Yes] See the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources 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...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] L. Ambrosio, N. Gigli, and G. Savare. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics. ETH Zürich. Birkhäuser Basel, 2005.

[2] Luigi Ambrosio and Nicola Gigli. Construction of the parallel transport in the wasserstein space. *Methods Appl. Anal.*, 15(1):1–30, 03 2008.

[3] Michael Arbel, Anna Korba, Adil SALIM, and Arthur Gretton. Maximum mean discrepancy gradient flow. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Daniel Bartl, Samuel Drapeau, and Ludovic Tangpi. Computational aspects of robust optimized certainty equivalents and option pricing. *Mathematical Finance*, 30(1):287–309, 2020.

[5] Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *SSRN Electronic Journal*, 04 2016.

[6] Jose Blanchet, Karthyek Murthy, and Fan Zhang. Optimal transport based distributionally robust optimization: Structural properties and iterative schemes. 2018.

[7] Claire Caillerie, Frédéric Chazal, Jérôme Dedecker, and Bertrand Michel. Deconvolution for the Wasserstein metric and geometric inference. *Electronic Journal of Statistics*, 5(none):1394 – 1423, 2011.

[8] R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications II: Mean Field Games with Common Noise and Master Equations*. Probability Theory and Stochastic Modelling. Springer International Publishing, 2018.

[9] J. A. Carrillo, K. Craig, L. Wang, and Chaozhen Wei. Primal dual methods for wasserstein gradient flows. *arXiv: Numerical Analysis*, 2019.

[10] José Antonio Carrillo, Katy Craig, and Francesco S. Patacchini. A blob method for diffusion. *Calculus of Variations and Partial Differential Equations*, 58(2):53, Feb 2019.

[11] Raymond J. Carroll and Peter Hall. Optimal rates of convergence for deconvolving a density. *Journal of the American Statistical Association*, 83(404):1184–1186, 1988.

[12] Z. Chen, D. Kuhn, and W. Wiesemann. Data-driven chance constrained programs over wasserstein balls. *arXiv: Optimization and Control*, 2018.

[13] T.C.E. Cheng and Mikhail Y. Kovalyov. An unconstrained optimization problem is np-hard given an oracle representation of its objective function: a technical note. *Computers & Operations Research*, 29(14):2087–2091, 2002.

[14] Sinho Chewi, Tyler Maunu, Philippe Rigollet, and Austin J. Stromme. Gradient descent algorithms for Bures-Wasserstein barycenters. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1276–1304. PMLR, 09–12 Jul 2020.

[15] Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[16] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[17] Casey Chu, Jose Blanchet, and Peter Glynn. Probability functional descent: A unifying perspective on GANs, variational inference, and reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1213–1222. PMLR, 09–15 Jun 2019.

[18] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. In *International Conference on Learning Representations*, 2020.

[19] Samuel Cohen, Michael Arbel, and Marc P. Deisenroth. Estimating barycenters of measures in high dimensions. *arXiv:2007.07105*, 2020.

[20] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[21] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[22] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, Aug 2014.

[23] Futoshi Futami, Zhenghang Cui, Issei Sato, and Masashi Sugiyama. Bayesian posterior approximation via greedy particle optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3606–3613, Jul. 2019.

[24] Rui Gao and A. Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *arXiv: Optimization and Control*, 2016.

[25] Soumyadip Ghosh, Mark Squillante, and Ebisa Wollega. Efficient stochastic gradient descent for learning with distributionally robust optimization. 2018.

[26] Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-part-1):902–917, 2010.

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[28] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, March 2012.

[29] O. Güler. *Foundations of Optimization*. Graduate Texts in Mathematics. Springer New York, 2010.

[30] Grani A. Hanasusanto and Daniel Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. *Operations Research*, 66(3):849–869, 2018.

[31] Daniel Hauer and José Mazón. Kurdyka–łojasiewicz–simon inequality for gradient flows in metric spaces. *Transactions of the American Mathematical Society*, 372(7):4917–4976, 2019.

[32] Daniel Kuhn, Peyman Mohajerin Esfahani, V. Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. *ArXiv*, abs/1908.08729, 2019.

[33] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8847–8860. Curran Associates, Inc., 2020.

[34] Guoyin Li and Ting Kei Pong. Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of Computational Mathematics*, 18(5):1199–1232, Oct 2018.

[35] Jiajin Li, Sen Huang, and Anthony Man-Cho So. A first-order algorithmic framework for distributionally robust logistic regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4082–4092. PMLR, 09–15 Jun 2019.

[37] L. Liu, Y. Zhang, Z. Yang, R. Babanezhad, and Z. Wang. Infinite-dimensional game optimization via variational transport. *OPT 2020*, 2020.

[38] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[39] Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[40] S. Mahdian, J. Blanchet, and P. Glynn. Optimal transport relaxations with application to wasserstein gans. *ArXiv*, abs/1906.03317, 2019.

[41] Quentin Mérigot, Alex Delalande, and Frederic Chazal. Quantitative stability of optimal transport maps and linearization of the 2-wasserstein space. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3186–3196. PMLR, 26–28 Aug 2020.

[42] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Math. Program.*, 171(1–2):115–166, September 2018.

[43] Youssef Mroueh, Tom Sercu, and Anant Raj. Sobolev descent. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2976–2985. PMLR, 16–18 Apr 2019.

[44] Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2216–2224, Red Hook, NY, USA, 2016. Curran Associates Inc.

[45] A. S. Nemirovskiĭ and Yu. E. Nesterov. Optimal methods for smooth convex minimization. *Zh. Vychisl. Mat. i Mat. Fiz.*, 25(3):356–369, 477, 1985.

[46] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2003.

[47] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[48] Philippe Rigollet and Jonathan Weed. Entropic optimal transport is maximum-likelihood deconvolution. *Comptes Rendus Mathematique*, 356(11):1228–1235, 2018.

[49] Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D. Lee. On the convergence and robustness of training gans with regularized optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7091–7101, Red Hook, NY, USA, 2018. Curran Associates Inc.

[50] F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Springer International Publishing, 2015.

[51] M. Scetbon, Laurent Meunier, J. Atif, and Marco Cuturi. Equitable and optimal transport with multiple agents. *arXiv: Machine Learning*, 2020.

[52] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[53] Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[54] Amirhossein Taghvaei and Prashant Mehta. Accelerated flow for probability distributions. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6076–6085. PMLR, 09–15 Jun 2019.

[55] Bahar Taskesen, V. Nguyen, Daniel Kuhn, and J. Blanchet. A distributionally robust approach to fair classification. *ArXiv*, abs/2007.09530, 2020.

[56] Bart P. G. Van Parys, Peyman Mohajerin Esfahani, and Daniel Kuhn. From data to decisions: Distributionally robust optimization is optimal. *Management Science*, 0(0):null, 0.

[57] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[58] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[59] Weijun Xie. On distributionally robust chance constrained programs with wasserstein distance. *Mathematical Programming*, 186(1):115–155, Mar 2021.

[60] K. Yoshida. *Functional Analysis*. Classics in mathematics / Springer. World Publishing Company, 1980.

[61] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.

[62] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.

[63] Chaoyue Zhao and Yongpei Guan. Data-driven risk-averse stochastic optimization with wasserstein metric. *Operations Research Letters*, 46(2):262–267, 2018.

[64] X. Zheng and H. Chen. Data-driven distributionally robust unit commitment with wasserstein metric: Tractable formulation and efficient solution method. *IEEE Transactions on Power Systems*, 35(6):4940–4943, 2020.

[65] A. Zhou, M. Yang, M. Wang, and Y. Zhang. A linear programming approximation of distributionally robust chance-constrained dispatch with wasserstein distance. *IEEE Transactions on Power Systems*, 35(5):3366–3377, 2020.

A Properties of Wasserstein space

The following properties of Wasserstein space make Definition 1 precise and are using in the convergence proofs in Appendix B.

Proposition 2 (Properties of Wasserstein space)**.**

- *For a constant-speed geodesic $\mu_t : [0,1] \to \mathcal{P}_2(\mathbb{R}^d)$ with respect to $\mathcal{W}$, there exists a (μ_t-almost surely) unique Borel vector field $v_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ which satisfies*

$$\mathcal{W}^2(\mu_0, \mu_1) = \int_0^1 \int_{\mathbb{R}^d} \|v_t(x)\|^2 \, d\mu_t(x) \, dt = \min_{v_t \in V_\mu} \int_0^1 \int_{\mathbb{R}^d} \|v_t(x)\|^2 \, d\mu_t(x) \tag{26}$$

  for

$$V_\mu := \left\{ v_t : \frac{d\mu_t}{dt} + \nabla \cdot (v_t \mu_t) = 0 \right\} \tag{27}$$

  defined as the set of all Borel vector fields which solve the continuity equation for μ_t. The continuity equation is understood in duality with $C_c^\infty(\mathbb{R}^d)$.

- *For any constant-speed geodesic μ_t, the corresponding optimal transport plan $\gamma \in \Pi(\mu_0, \mu_1)$ and the corresponding vector field v_t (given by* (26)*) satisfy the relation*

$$v_t((1-t)x + ty) = y - x, \qquad \gamma\text{-almost surely} \tag{28}$$

  for Lebesgue-almost every t.

- *The space $\mathcal{P}_2(\mathbb{R}^d)$ is positively curved under $\mathcal{W}$ and at each point $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, the tangent space*

$$\mathrm{Tan}(\mu) := \overline{\{\nabla\psi \ : \ \psi \in C_c^\infty(\mathbb{R}^d)\}}^{L^2(\mu)} \tag{29}$$

  is the closure in $L^2(\mu)$ of the gradients of smooth functions with compact support. Via the Riesz isomorphism, $\mathrm{CoTan}(\mu) = \mathrm{Tan}(\mu)$ where $\mathrm{CoTan}(\mu)$ denotes the cotangent space. The tangent and cotangent bundles will be denoted $\mathrm{Tan}_{\mathcal{P}_2(\mathbb{R}^d)}$ and $\mathrm{CoTan}_{\mathcal{P}_2(\mathbb{R}^d)}$, respectively.

Proof of Proposition 2. To verify the first bullet, we first establish the existence of such a v_t. Let μ_t be the constant speed geodesic and define the set of functions

$$A_\mu := \left\{ z \in L^2([0,1]) : \mathcal{W}(\mu_t, \mu_s) \leq \int_s^t z(r) \, dr \ \ \forall \ 0 \leq s \leq t \leq 1 \right\}$$

It is clear that the function $m(r) := \mathcal{W}(\mu_0, \mu_1)$ is in A and satisfies

$$m = \arg\min_{z \in A} \int_0^1 z^p(r) \, dr \tag{30}$$

for any $p \geq 1$. Hence, the metric derivative $|\mu'|$ of μ_t fulfills

$$|\mu'|(t) = d(\mu_0, \mu_1) \qquad \text{Lebesgue almost everywhere for } t \in [0,1]$$

By Theorem 8.3.1 in [1], there exists Borel vector field $v_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ satisfying the continuity equation (27) such that

$$\|v_t\|_{L^2(\mu_t)} = |\mu'|(t) = \mathcal{W}(\mu_0, \mu_1) \qquad \text{Lebesgue almost everywhere for } t \in [0,1] \tag{31}$$

Combined with (30), this implies that v_t is a solution of (26). Uniqueness of v_t follows directly from the second bullet.

For a constant-speed geodesic μ_t from μ to ν. Theorem 2.4 in [2] gives that, for any $\sigma \in \mathcal{P}_2(\mathbb{R}^d)$,

$$\frac{d}{dt} \frac{1}{2} \mathcal{W}^2(\mu_t, \sigma) = \int \langle v_t(x), x - y \rangle \, d\bar{\gamma}(x,y) \qquad \forall \ \bar{\gamma} \in \Pi_o(\mu_t, \sigma) \tag{32}$$

where $\Pi_o(\mu_t, \sigma) \subseteq \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ is the set of optimal transport plans between μ_t and σ. Setting $\sigma = \nu$, the fact that μ_t is a geodesic implies that there is a unique optimal coupling $\gamma \in \mathcal{P}\left(\mathbb{R}^d \times \mathbb{R}^d\right)$ between μ and ν such that

$$((1-t)x + ty, y)_\# \, \gamma \ \in \ \Pi_o(\mu_t, \sigma)$$

Hence, (32) gives

$$-(1-t)\mathcal{W}^2(\mu,\nu) = \int \langle v_t(x), x-y\rangle \; d\bar{\gamma}(x,y) = -(1-t)\int \langle v_t((1-t)x+ty), y-x\rangle \; d\gamma(x,y) \tag{33}$$

$$\Rightarrow \quad \mathcal{W}^2(\mu,\nu) = \int \langle v_t((1-t)x+ty), y-x\rangle \; d\gamma(x,y) \tag{34}$$

For t satisfying (31), the fact that $\|v_t\|_{L^2(\mu_t)} = \mathcal{W}(\mu,\nu)$ and $\|y-x\|_{L^2(\gamma)} = \mathcal{W}(\mu,\nu)$ means that (34) gives equality for Cauchy-Schwarz. Thus, $v_t((1-t)x+ty) = y-x$, γ-almost surely and (28) follows for Lebesgue almost every $t \in [0,1]$.

The final bullet is a direct restatement of the results of Section 8.4 in [1]. □

B Proof of iteration complexity (17)

In this section, we provide a proof of Theorem 1. Consider the following lemma which quantifies the stability of the sub-problems that are used in Algorithm 1.

Lemma 2. *Let $\gamma \in \Pi(\mu,\nu)$ be an optimal transport plan between $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ and $\nu \in \mathcal{P}_2(\mathbb{R}^d)$. If $\phi_\mu \in C^1(\mathbb{R}^d)$ is L-smooth*

$$\|\nabla\phi_\mu(x) - \nabla\phi_\mu(y)\| \le L\,\|x-y\| \tag{35}$$

then

$$\left|\int_{\mathbb{R}^d} \langle \nabla\phi_\mu(x), y-x\rangle \; d\gamma(x,y) - \left(\int_{\mathbb{R}^d} \phi_\mu \, d\nu - \int_{\mathbb{R}^d} \phi_\mu \, d\mu\right)\right| \le \frac{L}{2}\mathcal{W}^2(\nu,\mu) \tag{36}$$

Proof. First, it will be shown that

$$\left|\int_{\mathbb{R}^d} \langle \nabla\phi_\mu(x), y-x\rangle \; d\gamma(x,y) - \int_0^1 \langle \nabla\phi_\mu, v_t\rangle_{\mu_t} \; dt\right| \le \frac{L}{2}\mathcal{W}^2(\nu,\mu) \tag{37}$$

for μ_t and v_t which correspond (26) to the unique-constant speed geodesic given by $\gamma \in \Pi(\mu,\nu)$ (9). Notice that, since $\nabla\phi_\mu$ has at most linear growth, therefore both terms in the left-hand side of (37) are finite. Moreover, by (28), one has

$$\int_0^1 \langle \nabla\phi_\mu, v_t\rangle_{\mu_t} \; dt = \int_0^1 \int_{\mathbb{R}^d\times\mathbb{R}^d} \langle \nabla\phi_\mu((1-t)x+ty), y-x\rangle \; d\gamma(x,y)\, dt \tag{38}$$

Thus, Cauchy-Schwarz and (35) give

$$\begin{aligned}
&\left|\int_0^1 \langle \phi_\mu, v_t\rangle_{\mu_t} \; dt - \int_{\mathbb{R}^d\times\mathbb{R}^d} \langle \nabla\phi_\mu(x), y-x\rangle \; d\gamma(x,y)\right| = \\
&\left|\int_0^1 \int_{\mathbb{R}^d\times\mathbb{R}^d} \langle \nabla\phi_\mu((1-t)x+ty) - \nabla\phi_\mu(x), y-x\rangle \; d\gamma(x,y)\, dt\right| \le \\
&\int_0^1 \int_{\mathbb{R}^d} tL\,\|x-y\|^2 \; d\gamma(x,y)\, dt = \frac{L}{2}\mathcal{W}^2(\nu,\mu)
\end{aligned}$$

To obtain (36), it only remains to show that

$$\int_0^1 \langle \nabla\phi_\mu, v_t\rangle_{\mu_t} \; dt = \int \phi_\mu \, d\nu - \int \phi_\mu \, d\mu \tag{39}$$

Moreover, since v_t satisfies (27), Lemma 8.1.2 in [1] gives

$$\int_0^1 \langle \nabla\psi, v_t\rangle_{\mu_t} \; dt = \int \psi \, d\nu - \int \psi \, d\mu \tag{40}$$

for every $\psi \in C_c^1(\mathbb{R}^d)$– where $C_c^1(\mathbb{R}^d)$ denotes the space of continuously differentiable functions on $\mathbb{R}^d$ with compact support. Hence, (39) will be obtained from (40) by the following approximation argument.

Define the functions:

$$\beta_-(x) := \left(\sqrt{\|x\|^2+1} - \sqrt{2}\right)^{-1} \quad \text{and} \quad \beta_+(x) := \left(\sqrt{5} - \sqrt{\|x\|^2+1}\right)^{-1}$$

and

$$\eta(x) := \begin{cases} 1 & \text{if } \|x\| \le 1 \\ \frac{e^{\beta_-(x)}}{e^{\beta_-(x)}+e^{\beta_+(x)}} & \text{if } 1 < \|x\| < 2 \\ 0 & \text{if } \|x\| \ge 2 \end{cases}$$

It is easy to verify that $\eta \in C_c^\infty\left(\mathbb{R}^d\right)$ and $\|\nabla\eta(x)\| \le B$ for all $x \in \mathbb{R}^d$ and some constant B. Moreover, η provides a sequence of mollified approximations of ϕ_μ

$$\psi_k(x) := \phi_\mu(x)\eta_k(x) \quad \text{for} \quad \eta_k(x) := \eta\left(\frac{x}{k}\right)$$

where $\psi_k \in C_c^1(\mathbb{R}^d)$. Clearly, (40) holds for all such ψ_k. Thus, if

$$\lim_{k\to\infty} \int \psi_k \, d\nu - \int \psi_k \, d\mu = \int \phi_\mu \, d\nu - \int \phi_\mu \, d\mu \tag{41}$$

and

$$\lim_{k\to\infty} \int_0^1 \langle \nabla\psi_k, v_t\rangle_{\mu_t} \, dt = \int_0^1 \langle \nabla\phi_\mu, v_t\rangle_{\mu_t} \, dt \tag{42}$$

then (39) will follow directly from (40).

The relations (41) and (42) are straight-forward consequences of dominated convergence. Indeed, as $\eta_k \to 1$ and $\nabla\eta_k \to 0$ (pointwise), clearly

$$\psi_k \to \phi_\mu \quad \text{and} \quad \nabla\psi_k \to \nabla\phi_\mu \tag{43}$$

Quadratic growth of ϕ_μ yields $\phi_\mu \in L^2(\mu) \cap L^2(\nu)$ and combined with

$$|\psi_k(x)| \le |\phi_\mu(x)| \quad \forall x \in \mathbb{R}^d$$

(41) clearly holds via dominated convergence. One also has

$$\|\nabla\psi_k(x)\| \le \|\nabla\phi_\mu(x)\| + \frac{B|\phi_\mu(x)|}{k}\mathbf{1}_{\{\|x\|<2k\}} \tag{44}$$

Using the quadratic growth of ϕ_μ, linear growth of $\|\nabla\phi_\mu\|$, and the bound $\|x\|\,\mathbf{1}_{\{\|x\|<2k\}}/k \le 2$, (44) yields

$$\|\nabla\psi_k(x)\| \le \|\nabla\phi_\mu(x)\| + C\,\|x\|\,\mathbf{1}_{\{\|x\|<2k\}} + D \le E\,\|x\| + F \tag{45}$$

for some constants $C, D, E \in \mathbb{R}_+$. Recalling (38), (45) provides

$$\begin{aligned} \int_{\mathbb{R}^d\times\mathbb{R}^d} |\langle \nabla\psi_k((1-t)x+ty), y-x\rangle| \, d\gamma(x,y) &\le \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\nabla\psi_k((1-t)x+ty)\| \, \|y-x\| \, d\gamma(x,y) \\ &\le \int_{\mathbb{R}^d\times\mathbb{R}^d} \left(E\,\|(1-t)x+ty\| + F\right) \|y-x\| \, d\gamma(x,y) \\ &\le H \end{aligned} \tag{46}$$

for some $H \in \mathbb{R}_+$; where the last inequality is a result of Cauchy-Schwarz. The combination of pointwise convergence (43) and (46) then immediately yield (42) by dominated convergence and (38). □

We also require the following elementary results regarding the convergence of certain polynomial sequences.

Lemma 3. *Let $r_i \in \mathbb{R}_+$ be a sequence of non-negative numbers satisfying*

$$r_{i+1} \le r_i - \kappa r_i^p \tag{47}$$

for some constants $\kappa > 0$ and $p \ge 0$. Then,

$$r_n \le \begin{cases} e^{-\kappa n/r_0^{1-p}} r_0 & \text{if } p \le 1 \\ \left(\kappa n + r_0^{1-p}\right)^{-1/(p-1)} & \text{if } p > 1 \end{cases} \tag{48}$$

Proof. If $p \leq 1$, then (47) combined with the fact that r_i is a non-increasing sequence implies

$$r_i \leq \left(1 - \frac{\kappa}{r_0^{1-p}}\right) r_{i-1}$$

Iterating this inequality from 1 to n yields the first part of (48). Next, let $p > 1$ and notice that, by taking the reciprocals of both sides of (47) and rearranging, one obtains

$$\frac{\kappa r_{i-1}^{p-2}}{1 - \kappa r_{i-1}^{p-1}} \leq r_i^{-1} - r_{i-1}^{-1}$$

Summing this inequality over i (from 1 to n),

$$\kappa n r_k^{p-2} \leq \sum_{i=1}^{n} \frac{\kappa r_{i-1}^{p-2}}{1 - \kappa r_{i-1}^{p-1}} \leq r_k^{-1} - r_0^{-1}$$

where the first inequality is a result of r_i being non-increasing. Algebraic manipulation then provides

$$r_n \leq \left(\kappa n + r_0^{1-p}\right)^{-1/(p-1)}$$

□

Proof of Theorem 1. Recall the parameters specified in Assumptions 1 and 3 and let ϵ be the desired tolerance with which (16) should hold. Let Algorithm 1 be run with the following parameters:

$$\beta_1 = \min\left(\Delta_1, \Delta_2\right), \quad \beta_2 = \alpha(4L)^{-1}, \quad \beta_3 = (1 - \alpha/2)^{1/\alpha} T^{-1/\alpha} \tag{49}$$

and

$$r = \tau\epsilon^{\theta}/2, \quad \hat{\epsilon} = \left(2\alpha^*\right)^{-1} r, \quad \bar{\epsilon} = \alpha r/2, \quad \widetilde{\epsilon} = \left(4\alpha^*\right)^{-1} r, \quad k = \lceil M \rceil \tag{50}$$

where $\alpha^* = (1+\alpha)/\alpha$ is the dual exponent of $1+\alpha$ and M is defined in (65). It will be shown that the last iterate, μ_l, computed by Algorithm 1 satisfies (16).

First, we bound the decrease in J at each step of Algorithm 1. Let δ_i be the ith value of δ that is computed by Algorithm 1 and let s_i denote the ith value of s. One has the relation

$$\delta_i = \min\left(\beta_1, \beta_2 s_i, \beta_3 s_i^{\alpha^*-1}\right) \tag{51}$$

and, since $\delta_i \leq \Delta_2$ for all i, $\mu_0 \in S$ implies $\mu_i \in S$ for all i. Via the smoothness of J on S and $\delta_i \leq \Delta_1$, it follows that

$$J(\mu_i) \leq J(\mu_{i-1}) + \int_{\mathbb{R}^d \times \mathbb{R}^d} \langle F(\mu_{i-1}; x), y - x \rangle \, d\gamma(x,y) + \frac{T}{1+\alpha}\delta_i^{1+\alpha}$$

for any optimal transport plan $\gamma \in \Pi(\mu_i, \mu_{i-1})$ between μ_i and μ_{i-1}. Recognizing (12),

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} \left\langle F(\mu_{i-1}; x) - \nabla\widehat{\phi}_{\mu_{i-1}}(x), y - x \right\rangle d\gamma(x,y) \leq \left\| F(\mu_{i-1}; x) - \nabla\widehat{\phi}_{\mu_{i-1}} \right\|_{L^2(\mu_{i-1})} W(\mu_i, \mu_{i-1}) \leq \delta_i \hat{\epsilon}$$

and therefore

$$J(\mu_i) \leq J(\mu_{i-1}) + \int_{\mathbb{R}^d \times \mathbb{R}^d} \nabla\widehat{\phi}_{\mu_{i-1}}(x)^T (y - x) \, d\gamma(x,y) + \frac{T}{1+\alpha}\delta_i^{1+\alpha} + \delta_i \hat{\epsilon}$$

Via Lemma 2,

$$J(\mu_i) \leq J(\mu_{i-1}) + \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_i - \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_{i-1} + \frac{T}{1+\alpha}\delta_i^{1+\alpha} + \frac{L}{2}\delta_i^2 + \delta_i \hat{\epsilon} \tag{52}$$

Now, since $\widehat{\phi}_{\mu_{i-1}}$ is L-smooth, it is a Kantorovich potential [1, Section 6.1] for μ_{i-1}– under the cost function $L\left\|x - y\right\|^2/2$. Thus, there exists a geodesic ν_t (Proposition 2) such that: $\nu_0 = \mu_{i-1}$ and the transport plan $\gamma_t \in \Pi(\mu_{i-1}, \nu_t)$ between μ_{i-1} and ν_t satisfies [1, Section 8.3]

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} \left\langle \nabla\phi_{\mu_{i-1}}(x), y - x \right\rangle d\gamma_t(x,y) = -\frac{t}{L}\left\|\nabla\widehat{\phi}_{\mu_{i-1}}\right\|^2_{L^2(\mu_{i-1})} \quad \text{and} \quad \mathcal{W}\left(\nu_t, \mu_{i-1}\right) = \frac{t}{L}\left\|\nabla\widehat{\phi}_{\mu_{i-1}}\right\|_{L^2(\mu_{i-1})}$$

for $0 \leq t \leq 1$. For the sake of notation, define $g_{i-1} := \left\| \nabla \widehat{\phi}_{\mu_{i-1}} \right\|_{L^2(\mu_{i-1})}$ and set $t = L\delta_i / g_{i-1}$. Clearly, $t \leq 1$ since $\delta_i \leq \beta_2 s_i \leq \beta_2 g_{i-1}$.

By construction, μ_i also satisfies

$$\int \widehat{\phi}_{\mu_{i-1}} \, d\mu_i - \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_{i-1} \leq \int \widehat{\phi}_{\mu_{i-1}} \, d\nu_t - \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_{i-1} + \zeta_i$$

for $\zeta_i = \delta_i \widetilde{\epsilon}$. Hence, with another application of Lemma 2, one obtains

$$\begin{aligned} \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_i - \int \widehat{\phi}_{\mu_{i-1}} \, d\mu_{i-1} &\leq \int_0^t \left\langle \nabla \widehat{\phi}_{\mu_{i-1}}, v_s \right\rangle_{\nu_s} \, ds + \zeta_i \\ &\leq \int_{\mathbb{R}^d \times \mathbb{R}^d} \left\langle \nabla \widehat{\phi}_{\mu_{i-1}}(x), y - x \right\rangle \, d\gamma(x,y) + \frac{L}{2} \mathcal{W}(\nu_t, \mu_{i-1})^2 + \zeta_i \\ &= -\frac{t}{L} \left(1 - \frac{t}{2} \right) g_{i-1}^2 + \zeta_i \end{aligned} \tag{53}$$

Combining (53) with (52) and recalling $\delta_i = t g_{i-1} / L$ gives

$$J(\mu_i) \leq J(\mu_{i-1}) - \frac{t}{L} \left(C - t - \frac{D}{1+\alpha} t^\alpha \right) g_{i-1} + \zeta_i \tag{54}$$

for the values

$$C := 1 - \hat{\epsilon} \quad \text{and} \quad D := \frac{T}{L^\alpha g_{i-1}^{1-\alpha}}$$

Rewriting (54) using the residual term

$$r(\nu) := J(\nu) - \inf_{\mu \in S} J(\mu) \tag{55}$$

one obtains

$$r(\mu_i) \leq r(\mu_{i-1}) - \frac{t}{L} \left(C - t - \frac{D}{1+\alpha} t^\alpha \right) g_{i-1} + \zeta_i \tag{56}$$

This relation will now be used to show that Algorithm 1 makes sufficient progress on J, prior to the termination of it's loop.

Let l be the index of the last iterate μ_i which is computed by Algorithm 1. First, observe that if $s_{l+1} \leq r$, then early termination of the loop in Algorithm 1 has occurred. Using (14) and the definitions (50), it follows that

$$\begin{aligned} \tau \left(r(\mu_l) \right)^\theta &\leq \| F(\mu_l) \|_{L^2(\mu_l)} \leq g_l + \hat{\epsilon} \\ &\leq r + \bar{\epsilon} + \hat{\epsilon} \leq \tau \epsilon^\theta \end{aligned} \tag{57}$$

and, hence, sufficient progress on J has been made– μ_l satisfies (16). Thus, we need only analyze the case where early termination in Algorithm 1 does not occur and $l = k$ (50).

If $l = k$, then $s_i > r$ for all $i \leq k$ and, by extension, $g_{i-1} > r$ for all $i \leq k$ since s_i is a lower bound for g_{i-1}. In this case, the definitions of $\hat{\epsilon}$ and r (50) imply $C \geq 1 - \alpha / \left(2(1+\alpha) \right)$ and the choices for β_2 and β_3 (49) provide

$$t \leq \min \left(\frac{\alpha}{2(1+\alpha)}, \frac{(1 - \alpha/2)^{1/\alpha}}{D^{1/\alpha}} \right)$$

This gives

$$C - t - \frac{D}{(1+\alpha)} t^\alpha \geq (2\alpha^*)^{-1}$$

from which substitution into (56) yields

$$\begin{aligned} r(\mu_i) &\leq r(\mu_{i-1}) - \frac{t}{2L\alpha^*} g_{i-1} + \zeta_i \\ &\leq r(\mu_{i-1}) - \frac{\delta_i}{2\alpha^*} g_{i-1} + \zeta_i \\ &\leq r(\mu_{i-1}) - \frac{\delta_i}{4\alpha^*} g_{i-1} \end{aligned} \tag{58}$$

where the last inequality is a result of the definition of $\widetilde{\epsilon}$ (50), ζ_i, and $g_{i-1} > r$. As δ_i is the minimum of three different terms (51), (58) will be used to analyze the amount of progress, that is made on the objective J, corresponding to each of these three terms. Note, the following identities that will be used in the analysis of each term:

$$\left(1 - \frac{\alpha}{2}\right) g_{i-1} \leq g_{i-1} - \frac{\alpha r}{2} \leq g_{i-1} - \bar{\epsilon} \leq s_i \tag{59}$$

and

$$\begin{aligned} -g_{i-1}^p &\leq -\left(\|F(\mu_{i-1})\|_{L^2(\mu_{i-1})} - \hat{\epsilon}\right)^p \\ &\leq -\left(1 - \frac{\alpha}{2+\alpha}\right)^p \|F(\mu_{i-1})\|_{L^2(\mu_{i-1})}^p \leq -\frac{1}{2e} \|F(\mu_{i-1})\|_{L^2(\mu_{i-1})}^p \end{aligned} \tag{60}$$

for all $1 \leq p \leq \alpha^*$. The relation (59) simply observes that s_i is a multiplicative approximation to g_{i-1} in Algorithm 1, while (60) is a consequence of $r - \hat{\epsilon} \leq \|F(\mu_{i-1})\|_{L^2(\mu_{i-1})}$.

First, consider the case where $\delta_i = \beta_1$. Substitution into (58), coupled with (60), provides

$$r(\mu_i) \leq r(\mu_{i-1}) - \frac{\beta_1}{8e\alpha^*} \|F(\mu_{i-1})\|_{L^2(\mu_{i-1})} \tag{61}$$

Applying (14) to (61) and defining $r_i := r(\mu_i)$ (for the sake of notation) yields

$$r_i \leq r_{i-1} - \kappa^{(1)} r_{i-1}^{\theta} \quad \text{for} \quad \kappa^{(1)} := \omega\beta_1 \tag{62}$$

for the constant $\omega = (8e\alpha^*)^{-1}\tau$. In the cases (51) corresponding to β_2 and β_3, similar applications of the previous identities (along with (59)) give

$$r_i \leq r_{i-1} - \kappa^{(2)} r_{i-1}^{2\theta} \quad \text{for} \quad \kappa^{(2)} := \omega\tau(1-\alpha/2)\beta_2 \tag{63}$$

$$r_i \leq r_{i-1} - \kappa^{(3)} r_{i-1}^{\alpha^*\theta} \quad \text{for} \quad \kappa^{(3)} := \omega\left(\tau(1-\alpha/2)\right)^{1/\alpha}\beta_3 \tag{64}$$

Now, for the sake of notation, define the function

$$z(u,v) := u^{-1}\epsilon^{-(1-v)_-}\left(r_0 \log^{1/(1-v)}(r_0/\epsilon)\right)^{(1-v)_+}$$

where $(\cdot)_+$ and $(\cdot)_-$ denote the positive and negative parts. Using Lemma 3, it follows that, if (62) occurs for more than $\omega^{-1}z(\beta_1,\theta)$ iterations of Algorithm 1, then $r_k \leq \epsilon$, where k is the index of the last loop iteration in Algorithm 1. Similar deductions for (63) and (64) lead to the conclusion that, if

$$k \geq \omega^{-1}\left(z(\beta_1,\theta) + z(\tau(1-\alpha/2)\beta_2, 2\theta) + z((\tau(1-\alpha/2))^{1/\alpha}\beta_3, \alpha^*\theta)\right) := M \tag{65}$$

then either (62), (63), or (64) has occurred sufficiently many times during the execution of Algorithm 1 to guarantee $r_k \leq \epsilon$. As k has been chosen exactly so that $k = \lceil M \rceil$ (50), one obtains that μ_k satisfies (16). The desired complexity bound (17) on M now follows by plugging in for β_1, β_2, and β_3 in (65) and then, taking asymptotic estimates as $\epsilon \to 0$; the term $z((\tau(1-\alpha/2))^{1/\alpha}\beta_3, \alpha^*\theta)$ clearly dominates. □

C Computational solution of the Frank-Wolfe problem (15)

This section provides a concrete, computational procedure and complexity guarantee for the subroutine (15) in Algorithm 1 which, for $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, requires solution of the problem

$$\inf_{\nu \in \mathcal{P}_2(\mathbb{R}^d), \mathcal{W}(\nu,\mu) \leq \delta} \int f \, d\nu \tag{66}$$

The starting point to solve this problem is duality. The dual of (66) is

$$D(f) := \sup_{\lambda \in \mathbb{R}_+} \left[\int_{\mathbb{R}^d} f_\lambda(x)\, d\mu(x) - (\delta^2\lambda)/2\right], \qquad \text{where } f_\lambda(x) := \inf_{y \in \mathbb{R}^d} f(y) + \frac{\lambda}{2}\|x-y\|^2 \tag{67}$$

where f_λ is the Moreau-Yosida envelope [60] for f. The problem (67) permits practical computation since it requires only finite dimensional optimization procedures to calculate f_λ and perform ascent in λ. Moreover, strong duality between (66) and (67) holds under quite general circumstances [5] and, particularly for any of the circumstances in this work where f is assumed to be smooth.

Previous work [5, 52] has noted that, in general, solution of (67) might still be computationally infeasible for smooth f; for arbitrary λ, f_λ could obscure a computationally difficult problem with many local minima. However, for large enough λ, f_λ is quite computable since it's defining minimization problem becomes convex. So long as all relevant λ in (67) are large enough, this means that (67) will be efficiently computable. This is equivalent to ensuring that the trust-region size δ in (66) is not too large.

With the following results, we establish a bound on δ which is simultaneously small enough to achieve computational tractability for (67), but large enough to permit the iterative complexities of Theorem 1. Algorithms 2 and 3 are also provided to leverage these results and yield a computational procedure with concrete complexity for solving (66). For simplicity, rewrite (67) as

$$D(f) = \sup_{\lambda \in \mathbb{R}} \; g(\lambda) - \left(\delta^2 \lambda\right)/2, \qquad \text{where } g(\lambda) := \int_{\mathbb{R}^d} f_\lambda \, d\mu \tag{68}$$

Recall that a function $\phi : \mathbb{R}^d \to \bar{\mathbb{R}}$ is called *semiconvex* if

$$x \;\longrightarrow\; \phi(x) + \frac{\lambda}{2} \left\| x - x_0 \right\|^2 \tag{69}$$

is convex for some $\lambda \geq 0$ and some $x_0 \in \mathbb{R}^d$. Further, a continuously differentiable function ϕ is called L-smooth if it has L-Lipschitz gradients:

$$\left\| \nabla \phi(y) - \nabla \phi(x) \right\| \leq L \left\| y - x \right\| \tag{70}$$

Lemma 4. *If f is differentiable and ρ_*-semiconvex* (69)*, the function g* (68) *is differentiable on (ρ_*, ∞) and*

$$g'(\lambda) = \frac{1}{2} \int_{\mathbb{R}^d} \left\| y^*_{\lambda,x} - x \right\|^2 d\mu(x), \quad y^*_{\lambda,x} := \arg\min_{y \in \mathbb{R}^d} f(y) + \frac{\lambda}{2} \left\| y - x \right\|^2 \tag{71}$$

where the unique minimizer $y^*_{\lambda,x}$ satisfies

$$\frac{1}{2} \left\| y^*_{\lambda,x} - x \right\|^2 \leq \frac{2}{\left(\lambda - \rho_*\right)^2} \left\| \nabla f(x) \right\|^2 \tag{72}$$

Additionally, for any $\rho_* < \lambda_1 \leq \lambda_2$ one has

$$\left(1 - 2\sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_2 - \rho_*}} \right) g'(\lambda_1) \leq g'(\lambda_2) \tag{73}$$

That is, for any $t^* > \rho_*$, g' is $1/2$-Holder continuous on $[t^*, \infty)$ with a constant depending only on t^* and ρ_*.

Proof. Define the functions

$$a_\lambda(y; x) := f(y) + \frac{\lambda}{2} \left\| y - x \right\|^2 \qquad \text{and} \qquad z_x(\lambda) := \inf_{y \in \mathbb{R}} a_\lambda(y; x)$$

Since f is ρ_*-semiconvex (69), $a_\lambda(y; x)$ is $\lambda - \rho_*$ strongly convex in y for $\lambda > \rho_*$. Therefore, the minimizer $y_{\lambda,x}$ is unique. Further, semiconvexity and differentiability of f provide the lower bound

$$a_\lambda(y; x) \geq f(x) + l_\lambda(y; x) \qquad \text{where} \qquad l_\lambda(y; x) := \nabla f(x)^T (y - x) + \frac{\lambda - \rho_*}{2} \left\| y - x \right\|^2$$

Noticing $l_\lambda(y; x) > 0$ for any $y \in \mathbb{R}^d$ such that $\left\| y - x \right\| > \left(2 \left\| \nabla f(x) \right\|\right) / \left(\lambda - \rho_*\right)$, one obtains (72).

For open subsets $O \subset (\rho_*, \infty)$ whose closure does not contain ρ_*, (72) implies that the radius of the ball containing $y^*_{\lambda,x}$ is uniformly bounded for all $\lambda \in O$. Danskin's theorem [29] can, therefore, be

applied to the function $z_x(\lambda) := f_\lambda(x)$ (67) to conclude that $z_x(\lambda)$ is differentiable on (ρ_*, ∞) with derivative

$$z_x'(\lambda) = \frac{1}{2}\left\|y^*_{\lambda,x} - x\right\|^2$$

Observing that $g(\lambda) = \mathbb{E}_{x\sim\mu}\left[z_x(\lambda)\right]$, the conclusion (71) then follows from (72) and dominated convergence.

Finally, let $\rho_* < \lambda_1 \leq \lambda_2$. Since $z_x(\lambda)$ is concave in λ

$$|z_x'(\lambda_1) - z_x'(\lambda_2)| = z_x'(\lambda_1) - z_x'(\lambda_2)$$

and it is enough to show a one-sided bound on the quantity $z_x'(\lambda_1) - z_x'(\lambda_2)$. To this end, observe

$$z_x'(\lambda_1) - z_x'(\lambda_2) \leq \left\|y^*_{\lambda_1,x} - x\right\| \left\|y^*_{\lambda_2,x} - y^*_{\lambda_1,x}\right\| \tag{74}$$

Hence, (73) can be provided by producing a bound on $\left\|y^*_{\lambda_2,x} - y^*_{\lambda_1,x}\right\|$. Strong convexity of $a_\lambda(y;x)$ in y yields the identity

$$a_{\lambda_2}(y^*_{\lambda_2,x};x) + \frac{\lambda_2 - \rho_*}{2}\left\|y^*_{\lambda_2,x} - y^*_{\lambda_1,x}\right\|^2 \leq a_{\lambda_2}(y^*_{\lambda_1,x};x) = a_{\lambda_1}(y^*_{\lambda_1,x};x) + \frac{\lambda_2 - \lambda_1}{2}\left\|y^*_{\lambda_1,x} - x\right\|^2$$

which, when combined with the fact that $a_{\lambda_1}(y^*_{\lambda_1,x};x) \leq a_{\lambda_2}(y^*_{\lambda_1,x};x)$ ($z_x(\lambda)$ is non-decreasing in λ), gives

$$\left\|y^*_{\lambda_2,x} - y^*_{\lambda_1,x}\right\| \leq \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_2 - \rho_*}}\left\|y^*_{\lambda_1,x} - x\right\| \tag{75}$$

Applying (75) to (74) and rearranging produces

$$\left(1 - 2\sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_2 - \rho_*}}\right) z_x'(\lambda_1) \leq z_x'(\lambda_2) \tag{76}$$

Taking the expectation with respect to x on both sides of (76) yields (73). □

Lemma 5. *If f is L-smooth* (70) *and $L < \lambda$ then any optimizer*

$$f(y^*) + \frac{\lambda}{2}\left\|y^* - x\right\|^2 = \inf_{y\in\mathbb{R}^d} f(y) + \frac{\lambda}{2}\left\|y - x\right\|^2$$

satisfies $\|y^* - x\| \geq \frac{\|\nabla f(x)\|}{2\lambda}$.

Proof. From L-smoothness and the fact $\lambda > L$, the function

$$v(y) := f(y) + \frac{\lambda}{2}\left\|y - x\right\|^2$$

is $(\lambda - L)$-strongly convex. To show $\|\nabla f(x)\| / (2\lambda) \leq \|y^* - x\|$ notice

$$\nabla f(y^*) + \lambda(y^* - x) = 0 \tag{77}$$

by first-order optimality conditions for y^*. Combining (77) with the L-smoothness of f, one obtains

$$\begin{aligned} &\|\nabla f(x) - \nabla f(y^*)\|^2 \leq L^2\left\|x - y^*\right\|^2 \\ \Rightarrow\ &\|\nabla f(x)\|^2 + \left(\lambda^2 - L^2\right)\left\|x - y^*\right\|^2 \leq 2\lambda\nabla f(x)^T(x - y^*) \leq 2\lambda\left\|\nabla f(x)\right\|\left\|x - y^*\right\| \end{aligned} \tag{78}$$

Using the fact that $\lambda > L$, the desired result then follows directly from (78). □

The properties provided by Lemma 4 and Lemma 5 now enable establishment of a relationship between trust region size δ (66) and the decision variables in (67).

Proposition 3. *If f is differentiable and ρ_*-semiconvex then, for any $\epsilon > 0$, there exists a $\lambda_\epsilon \leq \rho_* + \|\nabla f(x)\|^2_{L^2(\mu)} / (2\epsilon)$ such that*

$$\left(\sup_{\lambda\in\mathbb{R}} g(\lambda) - \left(\delta^2\lambda\right)/2\right) - \left(g(\lambda_\epsilon) - \left(\delta^2\lambda_\epsilon\right)/2\right) \leq \epsilon \tag{79}$$

Further, if f is L-smooth (70) *and $\delta = \|\nabla f(x)\|_{L^2(\mu)} / C$ for $C \geq 2L$, then λ_ϵ can be chosen in the interval $[l, u] \subseteq \mathbb{R}$ for*

$$l = \rho_* \quad \textit{and} \quad u = \min\left(\beta, \rho_* + C\right) \tag{80}$$

where $\beta = \rho_* + \|\nabla f(x)\|^2_{L^2(\mu)} / (2\epsilon)$

Proof. For any $\hat{\lambda} \geq \rho_*$, ρ_*-semiconvexity of f and the definition of g (68) provide the lower and upper bounds

$$g(\hat{\lambda}) = \int_{\mathbb{R}^d} f_\lambda \, d\mu \geq \int_{\mathbb{R}^d} f \, d\mu - \frac{\|\nabla f(x)\|^2_{L^2(\mu)}}{2(\hat{\lambda} - \rho_*)} \qquad \text{and} \qquad g(\lambda) \leq \int f \, d\mu, \;\; \forall \lambda \in \mathbb{R}$$

These allow one to obtain the identity

$$g(\hat{\lambda}) - \left(\delta^2 \hat{\lambda}\right)/2 \geq \left(g(\lambda) - \left(\delta^2 \lambda\right)/2\right) - \frac{\|\nabla f(x)\|^2_{L^2(\mu)}}{2(\hat{\lambda} - \rho_*)} + \frac{\delta^2\left(\lambda - \hat{\lambda}\right)}{2} \tag{81}$$

for any $\hat{\lambda} \geq \rho_* \geq 0$. Via (81), Proposition 3 can be easily established; indeed let us first show (79).

Define $\lambda_n \in \mathbb{R}$ to be an optimizing sequence for (68)

$$\lim_{n\to\infty} g(\lambda_n) - \left(\delta^2 \lambda_n\right)/2 = D(f)$$

and set $\beta := \rho_* + \|\nabla f(x)\|^2_{L^2(\mu)}/(2\epsilon)$. Since g is upper-semicontinuous, it is sufficient to show that there exists a $\lambda_\epsilon \leq \beta$ satisfying (79) if $\beta < \liminf\limits_{n\to\infty} \lambda_n$. Since $\beta < \liminf_{n\to\infty} \lambda_n$, one can assume without loss of generality that $\beta < \lambda_n$ for all $n \in \mathbb{N}$. Substituting $\hat{\lambda} = \beta$ and $\lambda = \lambda_n$ in (81) simplifying provides

$$g(\beta) - \left(\delta^2 \beta\right)/2 \geq g(\lambda_n) - \left(\delta^2 \lambda_n\right)/2 - \epsilon \tag{82}$$

Taking the limit in (82) and setting $\lambda_\epsilon = \beta$ gives the desired result (79).

To show the second half of Proposition 3, observe that the previous result implies one can assume $\liminf_{n\to\infty} \lambda_n \leq \beta$ for an optimizing sequence λ_n– otherwise, β is ϵ-optimal. The immediate consequence of this assumption is that an optimizer λ^* of (68) exists. Indeed, L-smoothness of f provides $g(\lambda) = -\infty$ for any $\lambda < -L$ and, combined with $\liminf_{n\to\infty} \lambda_n \leq \beta$, the optimizing sequence λ_n can be assumed to be bounded. Via Bolzano-Weierstrauss, the sequence is therefore convergent to some $\lambda^* \leq \beta$ and upper-semicontinuity of g along with lower-semicontinuity of ψ^* then imply that λ^* is an optimizer of (68).

The main consequence of the existence of λ^* is that, in combination with (81), one has the upper bound

$$\begin{aligned} \frac{\delta^2(\lambda^* - \lambda)}{2} - \frac{1}{2(\lambda - \rho_*)_+} \|\nabla f(x)\|^2_{L_2(\mu)} + g(\lambda^*) - \left(\delta^2 \lambda^*\right)/2 \leq g(\lambda) - \left(\delta^2 \lambda\right)/2 \\ \Rightarrow \quad \frac{\delta^2\left(\lambda^* - \lambda\right)}{2} \leq \frac{\|\nabla f(x)\|^2_{L_2(\mu)}}{2(\lambda - \rho_*)_+} \\ \Rightarrow \quad \delta^2(\lambda - \rho_*)_+ \left(\lambda^* - \lambda\right) \leq \|\nabla f(x)\|^2_{L_2(\mu)} \end{aligned} \tag{83}$$

if $\lambda \leq \lambda^*$. Taking $\lambda = (\lambda^* + \rho_*)/2$ above will lead to the desired conclusion of Proposition 3– so long as $\rho_* \leq \lambda^*$. To show that $C \geq 4L$ implies $\rho_* \leq \lambda^*$, observe that Lemma 5, in combination with Lemma 4, implies

$$\frac{1}{8\lambda^2} \|\nabla f(x)\|^2_{L^2(\mu)} \leq \partial_+ g(\lambda), \quad \lambda \geq L \tag{84}$$

where $\partial_+(\cdot)$ denotes the derivative of g from the right. Under $C \geq 4L$, (84) produces the relation

$$\frac{\delta^2}{2} \leq \frac{1}{8L^2} \|\nabla f(x)\|^2_{L^2(\mu)} \leq \partial_+ g(L) \leq \partial_+ g(\rho_*) \tag{85}$$

since $\rho_* \leq L$. As g is concave, (85) immediately gives $g(\rho_*) - \left(\delta^2 \rho_*\right)/2 \geq g(\lambda) - \left(\delta^2 \lambda\right)/2$ for all $\lambda < \rho_*$. Hence, λ^* can be chosen so that $\rho_* \leq \lambda^*$. Finally, using the fact that $\rho_* \leq \lambda^*$ and substituting $\lambda = (\lambda^* + \rho_*)/2$ into (83), one obtains

$$\lambda^* \leq \rho_* + \left(\frac{4\|\nabla f(x)\|^2_{L^2(\mu)}}{\delta^2}\right)^{1/2} \leq \rho_* + C \tag{86}$$

After combining (86) with the bounds $\rho_* \leq \lambda^*$ and $\lambda^* \leq \beta$, the final conclusion of Proposition 3 follows. □

With the bounds of Proposition 3 in hand, a suitable gradient oracle for g (68) can be provided.

Definition 5 (Gradient oracle with high probability)**.** A function $\theta_g : \mathbb{R} \to \mathbb{R}$ is called a (ϵ, γ)-gradient oracle *with high probability* for g if, when queried with a λ, it returns an independent random sample $\theta_g(\lambda)$ satisfying

$$\mathbb{P}\left(\left[|\theta_g(\lambda) - g'(\lambda)| \geq \frac{\epsilon}{\max(\lambda - l, 1)}\right]\right) \leq \gamma \tag{87}$$

Algorithm 2 Gradient oracle for g (68)

Input: Distribution μ, point λ, semi-convexity parameter ρ_*, smoothness parameter L, error tolerance ϵ
Sample $x \sim \mu$
$y_0 \leftarrow x$, $\kappa \leftarrow \sqrt{(\lambda + L)/(\lambda - \rho_*)}$
$k \leftarrow \max\left(\lceil 4\kappa \log(12\kappa \|\nabla f(x)\| / \epsilon) \rceil, 0\right)$
for $1 \leq i \leq k$ **do**
  $z_i = y_{i-1} - \frac{1}{\kappa}\left(\nabla f(y_{i-1}) + \lambda(y_{i-1} - x)\right)$
  $y_i = z_i + \frac{\kappa - 1}{\kappa + 1}(z_i - z_{i-1})$
return $\theta = \frac{1}{2}\|y_k - x\|^2$

Proposition 4. *For a ρ_*-semiconvex function $f : \mathbb{R}^d \to \mathbb{R}$, which is also $L \geq \rho_*$ smooth* (70)*, the mean of*

$$K \geq \frac{64 \|\nabla f(x)\|^4_{L^4(\mu)}}{(\lambda - \rho_*)^2 \min\left((\lambda - \rho_*)^2, 1\right) \gamma \tilde{\epsilon}^2} \tag{88}$$

independent calls to Algorithm 2 with inputs $\lambda > \rho_*$ and $\epsilon = \tilde{\epsilon}/(2\max(\lambda - \rho_*, 1))$, provides a $(\tilde{\epsilon}, \gamma)$-gradient oracle with high probability (5) *on the interval (ρ_*, ∞).*

Proof. Consider the sample x which is computed by Algorithm 2. In light of Lemma 4, it is clear that

$$\theta^* := \frac{1}{2}\left\|y^*_{\lambda,x} - x\right\|^2$$

is an unbiased estimate of $g'(\lambda)$. Hence, to prove establish Proposition 4, it will first be shown that the output of Algorithm 2, θ, satisfies

$$|\theta - \theta^*| \leq \epsilon \quad \text{and} \quad \theta \leq \left(\frac{4\|\nabla f(x)\|}{\lambda - \rho_*}\right)^2 \tag{89}$$

when $\lambda \in (\rho_*, \infty)$.

To this end, notice that Algorithm 2 performs Nesterov's accelerated gradient descent [46] on the $\lambda - \rho_*$-strongly convex and $\lambda + L$-smooth function $a_\lambda(y; x)$. Strong convexity yields the identity

$$\frac{\lambda - \rho_*}{2}\left\|y^*_{\lambda,x} - y\right\|^2 \leq a_\lambda(y; x) - a_\lambda(y^*_{\lambda,x}; x) \tag{90}$$

while the convergence guarantees of accelerated gradient descent [46, Theorem 2.2.3] give

$$a_\lambda(y_k; x) - a_\lambda(y^*_{\lambda,x}; x) \leq \left(1 - \kappa^{-1}\right)^k (\lambda + L)\left\|y^*_{\lambda,x} - x\right\|^2 \tag{91}$$

for $\kappa = \sqrt{(\lambda + L)/(\lambda - \rho_*)}$. Combining these relations and setting $C = 2\|\nabla f(x)\|/(\lambda - \rho_*)$

$$\left\|y^*_{\lambda,x} - y_k\right\|^2 \leq \frac{2\left(a_\lambda(y_k; x) - a_\lambda(y^*_{\lambda,x}; x)\right)}{\lambda - \rho_*} \leq 2\left(1 - \kappa^{-1}\right)^k \kappa^2 \left\|y^*_{\lambda,x} - x\right\|^2 \leq 2\left(\frac{\epsilon}{6C}\right)^2 \tag{92}$$

since $k \geq 4\kappa \log(6\kappa C/\epsilon)$ and $\left\|y^*_{\lambda,x} - x\right\| \leq C$ via (72). Completing the analysis,

$$|\theta - \theta^*| = \frac{1}{2}\left|\|y_k - x\|^2 - \left\|y^*_{\lambda,x} - x\right\|^2\right| \leq \frac{1}{2}\left\|y_k - y^*_{\lambda,x}\right\|\left(\|y_k - x\| + \left\|y^*_{\lambda,x} - x\right\|\right) \tag{93}$$

$$\leq \frac{3}{2}\left\|y_k - y^*_{\lambda,x}\right\|\left\|y^*_{\lambda,x} - x\right\| \leq \epsilon \tag{94}$$

where triangle inequality provides both (93) and

$$\|y_k - x\| \leq 2 \left\| y^*_{\lambda,x} - x \right\| \leq 2C \tag{95}$$

Moreover, (94) is the desired left-hand inequality of (89) while (95) contains the desired right-hand inequality.

Establishing Proposition 4 is now a straightforward consequence of Chebyshev's inequality using (89). Indeed, one has

$$|\mathbb{E}[\theta] - g'(\lambda)| \leq \frac{\tilde{\epsilon}}{2\max(\lambda - \rho_*, 1)} \quad \text{and} \quad \theta \leq \frac{16}{(\lambda - \rho_*)^2} \|\nabla f(x)\|^2 \tag{96}$$

when $\epsilon = \tilde{\epsilon}/(2\max(\lambda - \rho_*, 1))$. Letting $\bar{\theta}$ be the average of K independent calls to Algorithm 2, Chebyshev's inequality gives

$$\mathbb{P}\left(\left| \bar{\theta} - \mathbb{E}[\theta] \right| \geq \frac{\tilde{\epsilon}}{2\max(\lambda - \rho_*, 1)} \right) \leq \frac{64 \ \|\nabla f(x)\|^4_{L^4(\mu)}}{(\lambda - \rho_*)^2 \min\left((\lambda - \rho_*)^2, 1 \right) \tilde{\epsilon}^2 K} \leq \gamma \tag{97}$$

□

The supergradient oracle of Proposition 4 provides a mechanism to perform ascent steps in λ to solve (68). Indeed, one can now perform bisection ascent for this problem. For the sake of our Frank-Wolfe procedure, it is of importance that this ascent procedure implicitly maintains a primal-feasible iterate for

$$\inf_{\pi \in \Pi(\mu)} \int f \, d\pi + \psi\left(\int \frac{1}{2} \|y - x\|^2 \, d\pi \right) \tag{98}$$

and makes progress on the primal-dual gap between (98) and (67). For this reason, we title the algorithm a "primal-dual" algorithm.

Algorithm 3 Primal-dual, bisection ascent for (68)

Input: Supergradient oracle θ_g, error tolerance ϵ, termination width B
  $\eta \leftarrow \infty$, $b \leftarrow l$
  while $u - l > \epsilon/B$ **do**
    $\lambda \leftarrow (l + u)/2$
    $\eta \leftarrow \theta_g(\lambda)$, $\eta \leftarrow \left(\eta - (\psi^*)'(\lambda) \right)$
    if $\eta < -\epsilon/\max(\lambda - b, 1)$ **then** $u \leftarrow \lambda$
    else $l \leftarrow \lambda$
  return u

Remark 5. The primal iterate that this algorithm maintains can be clarified by recalling that the conditions of Proposition 3 guarantee that $\lambda^* > \rho_*$ for any ρ_*-semiconvex f and optimal λ^* in (67). Since the function $y \mapsto f(y) + \lambda/2 \|x - y\|^2$ is strictly convex for $\lambda > \rho_*$, the distribution given by

$$(X, m(X)) \sim \pi_{\lambda,\mu}, \qquad X \sim \mu, \quad m_\lambda(x) = \arg\min_{y \in \mathbb{R}^d} f(y) + \frac{\lambda}{2} \|y - x\|^2 \tag{99}$$

provide the unique coupling such that

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} f(y) + \frac{\lambda^*}{2} \|y - x\|^2 \, d\pi_{\lambda,\mu}(x, y) = \int_{\mathbb{R}^d} \min_{y \in \mathbb{R}^d} \left(f(y) + \frac{\lambda^*}{2} \|y - x\|^2 \right) d\mu(x) \tag{100}$$

Through λ, Algorithm 3 implicitly maintains $\pi_{\lambda,\mu}$ and the criterion used for bisection of an interval in Algorithm 3 is designed to make progress on the primal-dual gap between the current dual iterate λ_i and $\pi_{\lambda_i,\mu}$:

$$G(\lambda_i) := \int f \, d\pi_{\lambda_i,\mu} + \infty \mathbf{1}_{(\delta,\infty)} \left(\int \|y - x\|^2 \, d\pi_{\lambda_i,\mu} \right) - \left(g(\lambda_i) - \left(\delta^2 \lambda_i \right)/2 \right) \tag{101}$$

This stands contrary to other approaches [52] for solving (67) where the computed dual feasible iterate λ_i need not provide a primal feasible iterate.

Theorem 6. *If $\mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$, f is L-smooth* (70)*, and $\delta \leq \|\nabla f\|_{L^2(\mu)} / (2L)$, there exists a stochastic algorithm which (for any probability $\gamma < 1$) computes a λ^* such that $\mathcal{W}\left(\nu_{\lambda^*}, \mu\right) \leq \delta$ and*

$$\int f \, d\nu_{\lambda^*} - \inf_{\mathcal{W}(\nu,\mu)\leq\delta} \int f \, d\nu \leq \epsilon \tag{102}$$

where ν_{λ^*} is second marginal of $\pi_{\lambda^*,\mu}$ in (99)*. This algorithm requires at most $\widetilde{O}(L^2 \|\nabla f\|^4_{L^4(\mu)} / ((1-\gamma)\epsilon^2))$ independent samples from μ and executes $\widetilde{O}(L^{5/2} \|\nabla f\|^4_{L^4(\mu)} / ((1-\gamma)\epsilon^2))$ gradient evaluations of f in expectation.*

Lemma 7. *Under the assumptions of Theorem 6, let Algorithm 3 be run with a (ϵ, τ)-gradient oracle (Definition 5) and $B = 4(g'(l))^2$ on the interval $[l, u]$. Recalling* (99)*, if $1 \leq l - L$ and $g'(u) - \delta^2/2 \leq 0 \leq g'(l) - \delta^2/2$, the output λ^* of Algorithm 3 satisfies (recall* (101)*)*

$$G(\lambda^*) \leq (4 + l)\epsilon \tag{103}$$

with probability at least $1 - \tau\left(\log_2\left(B(u-l)/\epsilon\right) + 1\right)$.

Proof. Let λ_i, u_i, l_i and η_i denote the ith values of λ, u, l and η which are computed by Algorithm 3– the indexes l_0, u_0 denote the initial values of these variables. Let k denote the total number of iterations performed by the loop of Algorithm 3. Since $u_i - l_i = (u_{i-1} - l_{i-1})/2$, it is clear that $k \leq \log_2\left(B(u_0 - l_0)/\epsilon\right) + 1$. Thus, using (87) and the fact that λ_i depends only on $\theta_g(\lambda_j)$ for $j < i$, one obtains the union bound

$$\mathbb{P}\left(\bigcup_{i\leq k}\left[|\theta_g(\lambda_i) - g'(\lambda_i)| \geq \frac{\epsilon}{\max(\lambda_i - l_0, 1)}\right]\right) \leq \tau\left(\log_2\left(B(u_0 - l_0)/\epsilon\right) + 1\right) \tag{104}$$

Hence, it need only be shown that (103) holds when

$$|\theta_g(\lambda_i) - g'(\lambda_i)| \leq \frac{\epsilon}{\max\left(\lambda_i - l_0, 1\right)} \qquad \forall\, i \leq k \tag{105}$$

For brevity, set $\epsilon_{\lambda_i} = \epsilon / \max(\lambda_i - l_0, 1)$ and recall $\eta_i = \theta_g(\lambda_i) - \delta^2/2$. Define $\eta_i^* := g'(\lambda_i) - \delta^2/2$ to be the true supergradient of (68) which η_i approximates. From (105)

$$\eta_i \eta_i^* \leq 0 \qquad \Rightarrow \qquad \max\left(|\eta_i|, |\eta_i^*|\right) \leq \epsilon_{\lambda_i} \tag{106}$$

Hence, at all iterations prior to the last iteration (iteration k), η_i and η_i^* have the same sign. Since $\lambda \mapsto \left(g(\lambda) - \left(\delta^2\lambda\right)/2\right)$ is concave, this gives

$$\sup_{\lambda\in[l_i,u_i]} g(\lambda) - \left(\delta^2\lambda\right)/2 = \sup_{\lambda\in[l_{i-1},u_{i-1}]} g(\lambda) - \left(\delta^2\lambda\right)/2 \tag{107}$$

for all $1 < i < k$. Additionally, if $\eta_k \eta_k^* > 0$ then (107) also holds for $i = k$.

From Algorithm 3, it is clear that either $u_k = \lambda_i$ for some $i > 0$ such that $\eta_i < -\epsilon_{\lambda_i}$ or $u_k = u_0$. Similarly, $l_k = \lambda_j$ for some $j > 0$ such that $\eta_j \geq -\epsilon_{\lambda_j}$ or $l_k = l_0$. One can assume, without loss of generality, that $l_k = \lambda_j$ for some $j > 0$ and, in combination with (106) and $g'(u_0) - \delta^2/2 \leq 0 \leq g'(l_0) - \delta^2/2$, this gives

$$-\epsilon_{\lambda_j} \leq g'(l_k) - \delta^2/2 \quad \text{and} \quad g'(u_k) - \delta^2/2 \leq 0 \tag{108}$$

Thus,

$$G(u_k) = \int f \, d\pi_{u_k,\mu} - \left(g(u_k) - \left(\delta^2 u_k\right)/2\right) = u_k\left(\delta^2/2 - g'(u_k)\right)$$

and using (73), one obtains

$$\begin{aligned} G(u_k) &\leq u_k\left(\delta^2/2 - \left(1 - \frac{2}{\sqrt{u_k - \rho_*}}(u_k - l_k)^{1/2}\right) g'(l_k)\right) \\ &\leq u_k \epsilon_{\lambda_j} + \frac{2u_k(u_k - l_k)^{1/2}}{\sqrt{u_k - \rho_*}} g'(l_k) \end{aligned} \tag{109}$$

where the last inequality is a result of (108). To bound the first term on the left side of (109), notice that $l_k \neq l_0$ implies there exists a minimal $t > 0$ such that $l_t \neq l_0$. Clearly,

$$u_k \epsilon_{\lambda_j} = \frac{u_k \epsilon}{\max(\lambda_j - l_0, 1)} \leq \frac{u_k \epsilon}{\max(\lambda_t - l_0, 1)} \leq \epsilon \frac{u_t - l_0}{\max(\lambda_t - l_0, 1)} + l_0 \epsilon \leq (2 + l_0)\epsilon$$

Combining this with the termination condition

$$u_k - l_k \leq \frac{\epsilon}{B} \leq \frac{\epsilon}{4g(l_0)^2} \leq \frac{\epsilon}{4g(l_k)^2}$$

to bound the second term of (109), one obtains

$$T \leq (4 + l_0)\epsilon$$

□

Proof of Theorem 6. Let $l = L + 1$ and $u = L + 1 + 4L$ and apply Algorithm 3 to the interval $[l, u]$ with the supergradient oracle given by Proposition 4. Set the error tolerance used by Algorithm 3 to $\epsilon / (4 + L + 1)$ and the termination width to $B := 16 \|\nabla f(x)\|^4_{L^2(\mu)}$. Likewise, the error tolerance used in Proposition 4 should be be $\epsilon / (4 + L + 1)$ and the error probability should be $(1 - \gamma)/(\log_2 (B(u - l)(4 + L + 1)/\epsilon) + 1)$.

Under this setting of parameters, Lemma 7 establishes that the output of λ^* of Algorithm 3 satisfies

$$G(\lambda^*) \leq \epsilon \tag{110}$$

with probability γ so long as

$$g'(u) - \delta^2/2 \leq 0 \leq g'(l) - \delta^2/2 \tag{111}$$

To see that (111) is fulfilled for the chosen l and u, notice that the conditions of Theorem 6 guarantee that Lemma 4 holds. Hence, (85) gives

$$g'(l) - \delta^2/2 \geq 0$$

Similarly, (72) provides

$$g'(u) - \delta^2/2 \leq 2 \|\nabla f(x)\|^2_{L^2(\mu)} / (u - L)^2 - \delta^2/2 \leq \|\nabla f(x)\|^2_{L^2(\mu)} / \left(16L^2\right) - \delta^2/2 \leq 0$$

Hence, (111) holds and the output λ^* of Algorithm 3 obeys (110) with probability γ.

It remains to compute a bound on the number of samples from μ which are required by this procedure. Clearly, by the definition of Algorithm 3, at most $\lceil \log_2 (B(u - l)(4 + L + 1)/\epsilon) \rceil$ calls are made to the supergradient oracle given by Proposition 4. Via (88), this yields that at most

$$\frac{(8(4 + L + 1)(\log_2 (B(u - l)(4 + L + 1)/\epsilon) + 1))^2 \|\nabla f(x)\|^4_{L^4(\mu)}}{(1 - \gamma)\epsilon^2} \tag{112}$$

invocations of Algorithm 2 are performed with an error parameter which is at least $\epsilon / (2(4 + L + 1)(\max(u - L, 1)))$. Since each invocation of Algorithm 2 requires a single sample of μ, it follows from (112) that

$$\widetilde{O}\left(\frac{L^2 \|\nabla f(x)\|^4_{L^4(\mu)}}{(1 - \gamma)\epsilon^2}\right)$$

samples are used by Algorithm 3– where $\widetilde{O}$ suppresses logarithmic factors in $L, C, M, \|\nabla f(x)\|^2_{L^2(\mu)}$, and ϵ.

Finally, to compute a bound on the expected number of gradient evaluations of f that are performed notice that, for an error parameter of ϵ, each of the k calls to Algorithm 2 (with error tolerance $\epsilon/(4(u - l))$) executes at most

$$t = \max\left(\left\lceil 4\kappa \log\left(\frac{48\kappa \|\nabla f(x)\| (u - l)}{\epsilon}\right)\right\rceil, 0\right)$$

gradient evaluations of f; x and κ are the random sample and condition number, respectively, which are used in Algorithm 2. Both x and κ are random variables, but $\kappa = ((\lambda + L)/(\lambda - \rho_*))^{1/2} \leq (1 + 2L)^{1/2}$ and (due to Jensen)

$$\mathbb{E}\left[\max\left(\log\left(z\right), 0\right)\right] \leq \log \mathbb{E}\left[\max\left(z, 1\right)\right]$$

for any non-negative random variable z. Hence, the expected number of gradient evaluations performed by Algorithm 2 obeys the bound

$$\mathbb{E}_\mu\left[t\right] \leq 4\left(1 + 2L\right)^{1/2} \log \mathbb{E}_\mu\left[\left(\max\left(\frac{48(1 + 2L)^{1/2}\left\|\nabla f(x)\right\|\left(u - l\right)}{\epsilon}, 1\right)\right]\right) \tag{113}$$

Summing over the k calls to Algorithm 2 and using the identity $u \leq l + 4L$, one obtains

$$\mathbb{E}_\mu\left[t\right] = \widetilde{O}\left(\frac{L^{5/2}\left\|\nabla f(x)\right\|^4_{L^4(\mu)}}{(1 - \gamma)\epsilon^2}\right)$$

□