Multiple Wasserstein Gradient Descent Algorithm for Multi-Objective Distributional Optimization

Dai Hai Nguyen[1] **Hiroshi Mamitsuka**[2] **Atsuyoshi Nakamura**[1]

[1]Hokkaido University, Hokkaido, Japan
[2]Kyoto University, Kyoto, Japan

Abstract

We address the optimization problem of simultaneously minimizing multiple objective functionals over a family of probability distributions. This type of Multi-Objective Distributional Optimization commonly arises in machine learning and statistics, with applications in areas such as multiple target sampling, multi-task learning, and multi-objective generative modeling. To solve this problem, we propose an iterative particle-based algorithm, which we call Multiple Wasserstein Gradient Descent (MWGraD), which constructs a flow of intermediate empirical distributions, each being represented by a set of particles, which gradually minimize the multiple objective functionals simultaneously. Specifically, MWGraD consists of two key steps at each iteration. First, it estimates the Wasserstein gradient for each objective functional based on the current particles. Then, it aggregates these gradients into a single Wasserstein gradient using dynamically adjusted weights and updates the particles accordingly. In addition, we provide theoretical analysis and present experimental results on both synthetic and real-world datasets, demonstrating the effectiveness of MWGraD.

1 INTRODUCTION

Many problems in machine learning and computational statistics turn into distributional optimization, where the goal is to minimize a functional $F : \mathcal{P}_2(\mathcal{X}) \to \mathbb{R}$ over the set of probability distributions: $\min_{q\in\mathcal{P}_2(\mathcal{X})} F(q)$, where $\mathcal{P}_2(\mathcal{X})$ denotes the set of probability distributions defined on the domain $\mathcal{X}$ ($\subseteq \mathbb{R}^d$) with finite second-order moment. This formulation arises in a variety of well-known problems, including Bayesian inference (e.g., variational autoencoder [Kingma and Welling, 2014]) and synthetic sample generation (e.g. generative adversarial networks [Goodfellow et al., 2020]). These models aim to approximate a target distribution π by generating samples (or also called particles) in a way that minimizes the dissimilarity $D(q, \pi)$ between the empirical probability distribution q derived from the particles and the target distribution π. Common dissimilarity measures include Kullback-Leiber (KL) divergence, Jensen-Shanon (JS) divergence, and Wasserstein distance in the optimal transport [Villani, 2021]. By setting $F(q) = D(q, \pi)$, the task can be framed as a distributional optimization problem. This problem can be solved using iterative algorithms, such as Wasserstein gradient descent [Zhang and Sra, 2016], which proceeds in two main steps at each iteration: (1) estimating the Wasserstein gradient of F with respect to the current distribution, and (2) applying the exponential mapping to update the distribution on $\mathcal{P}_2(\mathcal{X})$. However, step (1) can be non-trivial when the target distribution π is represented by a set of samples, making it challenging to estimate the Wasserstein gradient of F. To tackle this problem, Variational Transport (VT) [Liu et al., 2021b] is proposed. The key idea of VT is to assume that F has a variational form and reformulates F as a variational maximization problem. Solving this problem allows to approximate the Wasserstein gradient of F through samples from π, with resulting solution specifying a direction to update each particle. This can be viewed as a forward discretization of the Wasserstein gradient flow [Santambrogio, 2015]. Inspired by VT, several other methods have been introduced to address variants of distributional optimization problems, including MirrorVT [Nguyen and Sakurai, 2023] and MYVT [Nguyen and Sakurai, 2024].

On the other side, multi-objective optimization (MOO) [Deb et al., 2016] optimizes multiple objective functions simultaneously, and can be formulated as

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x}) = \min_{\mathbf{x}\in\mathcal{X}} f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_K(\mathbf{x}), \tag{1}$$

where $K \geq 2$ represents the number of objectives, $f_k(\mathbf{x})$ is the k-th objective function, and $\mathcal{X} \subseteq \mathbb{R}^d$ is the feasible set of d-dimensional vectors. Different from single-objective

optimization, in MOO, it could have two vectors where one performs better for task i and the other performs better for task $j \neq i$. Therefore, Pareto optimality is defined to deal with such an incomparable case. In particular, for two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we say that $\mathbf{y}$ dominates $\mathbf{x}$ if $f_k(\mathbf{x}) \leq f_k(\mathbf{y})$ for all $k \in [K]$, where $[K]$ denotes $\{1, 2, 3..., K\}$, and $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{y})$. A solution $\mathbf{x}$ is Pareto optimality if no other solution in $\mathcal{X}$ dominates it. MOO has found applications in various domains, including online advertising [Ma et al., 2018] and reinforcement learning [Thomas et al., 2021]. However, one of the main challenges in MOO is **gradient conflict**: objectives with large gradients can dominate the update direction, potentially leading to poor performance for other objectives with smaller gradients. To mitigate this issue, several MOO-based methods have been proposed to balance the contribution of all objectives. A typical method is Multiple Gradient Descent Algorithm (MGDA) [Désidéri, 2012], which seeks a conflict-avoidant update direction that maximizes the minimal improvement across all objectives. MGDA converges to a Pareto stationary point, where no common descent direction exists for all objective functions. Building on MGDA, several other methods have been introduced to further enhance conflict avoidance and improve solution balancing, including CAGrad [Liu et al., 2021a], PCGrad [Yu et al., 2020], GradDrop [Chen et al., 2020].

The work most closely related to ours is MT-SGD [Phan et al., 2022], which is designed for multi-target sampling. In this setup, we are given a set of multiple unnormalized target distributions and aim to generate particles that simultaneously approximate these target distributions. MT-SGD is shown to reduce to multi-objective optimization, where each objective function corresponds to a KL divergence for one of the target distributions. Additionally, MT-SGD has been successfully applied to multi-task learning, achieving state-of-the-art performance across various baselines, thanks to its ability to sample particles from the joint likelihood of multiple target distributions. Inspired by MT-SGD, we introduce the formulation of Multi-Objective Distribution Optimization (MODO). Given a set of objective functionals $F_1(q), F_2(q), ..., F_K(q) : \mathcal{P}_2(\mathcal{X}) \to \mathbb{R}$, where each $F_k(q)$ is defined over the space of probability distribution $q \in \mathcal{P}_2(\mathcal{X})$, our goal is to find the optimal distribution that minimizes the following vector-valued objective functional

$$\min_{q \in \mathcal{P}_2(\mathcal{X})} \mathbf{F}(q) = \min_{q \in \mathcal{P}_2(\mathcal{X})} F_1(q), F_2(q), ..., F_K(q). \quad (2)$$

Note that, while each f_k in (1) is defined over the space of vectors $\mathbf{x} \in \mathcal{X}$, each F_k in (2) is defined over the space of probability distributions $q \in \mathcal{P}_2(\mathcal{X})$. Similar to MOO, for two distribution $p, q \in \mathcal{P}_2(\mathcal{X})$, we say that p dominates q if $F_k(q) \leq F_k(p)$ for all $k \in [K]$, and $\mathbf{F}(p) \neq \mathbf{F}(q)$. A distribution q is Pareto optimality if no other distribution in $\mathcal{P}_2(\mathcal{X})$ dominates it.

To solve the MODO problem, we introduce an iterative algorithm, which we call Multi-objective Wasserstein Gradient Descent (MWGraD) by constructing a flow of probability distributions, gradually minimizing all the objective functionals. Specifically, MWGraD consists of two key steps at each iteration. First, for each objective functional F_k (for $k \in [K]$), MWGraD estimates the Wasserstein gradient based on the current probability distribution. Second, MWGraD aggregates these gradients into a single Wasserstein gradient using dynamically updated weights and updates the current probability distribution accordingly. In practice, MWGraD operates on a flow of empirical distributions, where each distribution is represented by a set of particles that are updated iteratively. We emphasize that MWGraD can be viewed as a generalized version of MT-SGD [Phan et al., 2022]. That is, while MT-SGD specifically tackles the multi-target sampling problem as a form of MODO with the KL divergence as the objective functional, MWGraD can handle a broader class of functionals. Furthermore, we provide theoretical analysis on the convergence of MWGraD to the Pareto stationary point, and experimental results on both synthetic and real-world datasets, demonstrating the effectiveness of the proposed algorithm.

2 RELATED WORKS

Distributional Optimization. Two widely used Bayesian sampling methods are Gradient Markov chain Monte Carlo (MCMC) [Welling and Teh, 2011] and Stein variational gradient descent (SVGD) [Liu and Wang, 2016]. Gradient MCMC generates samples from a Markov chain to approximate a target distribution (e.g., a posterior), but the resulting samples can be highly correlated. In contrast, SVGD initializes a set of particles and updates them iteratively to approximate the target distribution, often achieving good approximations with relatively fewer samples. As noted by [Chen et al., 2018], SVGD can be viewed as simulating the steepest descending curves, or gradient flows, of the KL-divergence on a certain kernel-related distribution space. Specifically, the functional $F(q)$ is defined as $KL(q, \pi)$, where $\pi(\mathbf{x}) \propto \exp\{-g(\mathbf{x})\}$ and $g(\mathbf{x})$ is often referred to as the energy function or potential function. Inspired by this perspective, other particle-based variational inference (ParVIs) methods have been developed to simulate the gradient flow in the Wasserstein space. The particle optimization (PO) method [Chen and Zhang, 2017] and the w-SGLD method [Chen et al., 2018] adopt the minimizing movement scheme [Jordan et al., 1998] to approximate the gradient flow using a set of particles. The Blob method [Chen et al., 2018] uses the vector field formulation of the gradient flow and approximates the update direction using particles. However, when F is not the energy functional and π is represented by a set of samples, it becomes non-trivial to define the Wasserstein gradient flow of F. To address this challenge, the VT algorithm [Liu et al., 2021b] assumes that F admits a variational form. This assumption allows the Wasserstein gradient to

be estimated using samples from both the current empirical distribution and the target distribution, enabling particle updates in specified directions. Building on VT, MirrorVT [Nguyen and Sakurai, 2023] extends this framework to optimize $F(q)$ when q is defined over a constrained domain. MYVT [Nguyen and Sakurai, 2024] further generalizes the approach to regularized distributional optimization, where the objective is composed of two functionals: one with a variational form and the other expressed as the expectation of a possibly nonsmooth convex function.

Multi-Objective Optimization (MOO). Several gradient-based techniques have been proposed for MOO. Among the most popular is MGDA [Désidéri, 2012], which seeks to find a conflict-avoidant update direction that maximizes the minimal improvement across all objectives. PCGrad [Yu et al., 2020] mitigates the gradient conflict by projecting the gradient of each task on the norm plane of other tasks. GradDrop [Chen et al., 2020] randomly drops out conflicted gradients, while CAGrad [Liu et al., 2021a] adds a constraint to ensure the update direction is close to the average gradient. The methods most closely related to our work are MOO-SVGD [Liu et al., 2021c] and MT-SGD [Phan et al., 2022], which enable sampling from multiple target distributions, a task that can be seen as an instance of MODO. MT-SGD aims to update particles in a way that brings them closer to all target distributions, effectively generating diverse particles that lie within the joint high-likelihood region of all targets. In contrast, MOO-SVGD uses MGDA [Désidéri, 2012] to update the particles individually and independently. In our synthetic experiments, we observe that the behavior of our proposed algorithm MWGraD closely resembles MT-SGD.

3 PRELIMINARIES

3.1 BASIC CONCEPTS OF OPTIMAL TRANSPORT AND WASSERSTEIN SPACE

Optimal transport [Villani, 2021] has received much attention in the machine learning community and has been shown to be an effective tool for comparing probability distributions in many applications [Nguyen and Tsuda, 2023, Nguyen et al., 2021, Petric Maretic et al., 2019, Nguyen et al., 2023]. Formally, given a measurable map $T : \mathcal{X} \to \mathcal{X}$ and $p \in \mathcal{P}_2(\mathcal{X})$, we say that q is the *push-forward measure* of p under T, denoted by $q = T\sharp p$, if for every Borel set $E \subseteq \mathcal{X}$, $q(E) = p(T^{-1}(E))$. For any $p, q \in \mathcal{P}_2(\mathcal{X})$, the 2-Wasserstein distance $\mathcal{W}_2(p, q)$ is defined as

$$\mathcal{W}_2^2(p,q) = \inf_{\pi \in \Pi(p,q)} \int_{\mathcal{X}\times\mathcal{X}} \|\mathbf{x} - \mathbf{x}'\|_2^2 \mathrm{d}\pi(\mathbf{x}, \mathbf{x}'),$$

where $\Pi(p, q)$ is all probability measures on $\mathcal{X} \times \mathcal{X}$ whose two marginals are equal to p and q, and $\|\cdot\|_2$ denotes the Euclidean norm. The metric space $(\mathcal{P}_2(\mathcal{X}), \mathcal{W}_2)$, also known as Wasserstein space, is an infinite-dimensional geodesic space [Villani et al., 2009, Definition 6.4]. Furthermore, we can endow the manifold $\mathcal{P}_2(\mathcal{X})$ with a Riemannian metric [Villani, 2021, p. 250], as follows: for any s_1, s_2 are two tangent vectors at p, where $s_1, s_2 \in \mathcal{T}_p\mathcal{P}_2(\mathcal{X})$, and $\mathcal{T}_p\mathcal{P}_2(\mathcal{X})$ denotes the space of tangent vectors at p, let $u_1, u_2 : \mathcal{X} \to \mathbb{R}$ be the solutions to the following elliptic equations $s_1 = -\mathtt{div}(\rho\nabla u_1)$ and $s_2 = -\mathtt{div}(\rho\nabla u_2)$, respectively, where $\mathtt{div}$ denotes the divergence operator on $\mathcal{X}$. The inner product between s_1 and s_2 is defined as

$$\langle s_1, s_2\rangle_p = \int_{\mathcal{X}} \langle \nabla u_1(\mathbf{x}), \nabla u_2(\mathbf{x})\rangle p(\mathbf{x}) d\mathbf{x}.$$

Definition 1. (First variation of a functional) Given a functional $F : \mathcal{P}_2(\mathcal{X}) \to \mathbb{R}$, the first variation of F evaluated at p, denoted by $\delta F(p) : \mathcal{X} \to \mathbb{R}$, is given as follows

$$\lim_{\epsilon\to 0} \frac{1}{\epsilon}\left(F(p + \epsilon\chi) - F(p)\right) = \int_{\mathcal{X}} \delta F(p)(\mathbf{x})\chi(\mathbf{x})\mathrm{d}\mathbf{x},$$

for all $\chi = q - p$, where $q \in \mathcal{P}_2(\mathcal{X})$.

With mild regularity assumptions, the Wasserstein gradient of F, denoted by $\mathtt{grad}F$, relates to the gradient of the first variation of F via the following continuity equation

$$\begin{aligned} &\mathtt{grad}F(p)(\mathbf{x}) = -\mathtt{div}\left(p(\mathbf{x})\nabla\delta F(p)(\mathbf{x})\right), \\ &\text{for all } \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{3}$$

We refer the readers to [Santambrogio, 2015] for details.

3.2 BASIC CONCEPTS OF MOO AND MGDA

For a MOO problem, very often, no single solution can optimize all the objectives at the same time. For instance, it could have two vectors where one performs better for the objective k and the other performs better for the objective $l \neq k$. Thus, Pareto optimality is defined to address such an incomparable case.

Definition 2. (Pareto optimality) For two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ of (1), we say that $\mathbf{y}$ dominates $\mathbf{x}$ if $f_k(\mathbf{x}) \leq f_k(\mathbf{y})$ for all $k \in [K]$ and $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{y})$. A solution $\mathbf{x}$ is Pareto optimal if no other solution in $\mathcal{X}$ dominates it.

Note that there is a set of Pareto optimal solutions, called Pareto set. Let denote the probability simplex as $\mathcal{W} = \left\{\mathbf{w} = (w_1, ..., w_K)^\top | \mathbf{w} \geq 0, \sum_{k=1}^K w_k = 1\right\}$, we introduce the concept of Pareto Stationarity (also referred to as Pareto Criticality) [Custódio et al., 2011] as follows.

Definition 3. (Pareto Stationary Solution) A solution $\mathbf{x}^* \in \mathcal{X}$ is a Pareto stationary solution iff some convex combination of the gradients $\{\nabla f_k(\mathbf{x}^*)\}$ vanishes, i.e., there exists some $\mathbf{w} \in \mathcal{W}$ such that $\nabla\mathbf{f}(\mathbf{x}^*)\mathbf{w} = \sum_{k=1}^K w_k \nabla f_k(\mathbf{x}^*) = \mathbf{0}$, where $\nabla\mathbf{f}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_K(\mathbf{x})]$.

MGDA [Désidéri, 2012] has gained significant attention in machine learning recently largely because of its gradient-based nature, in contrast to traditional MOO methods. In each iteration, MGDA updates the parameters as follows: $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \mathbf{d}^{(t)}$, where α is the learning rate and $\mathbf{d}^{(t)}$ is the MGDA search direction at t-th iteration. The key idea of MGDA is to find a update direction $\mathbf{d}^{(t)}$ that maximizes the minimum decrease across all the objective by solving the following primal problem

$$\mathbf{d}^{(t)} = \arg\min_{\mathbf{d}\in\mathbb{R}^d} \left\{ \max_{k\in[K]} \langle \nabla f_k(\mathbf{x}^{(t)}), \mathbf{d}\rangle + \frac{1}{2}\|\mathbf{d}\|_2^2 \right\}.$$

To simplify the optimization, such primal problem has a dual objective as a min-norm oracle

$$\mathbf{w}^{(t)} = \arg\min_{\mathbf{w}\in\mathcal{W}} \|\sum_{k=1}^{K} w_k \nabla f_k(\mathbf{x}^{(t)})\|_2^2.$$

The update direction is then calculated by $\mathbf{d}^{(t)} = \nabla \mathbf{f}(\mathbf{x}^{(t)})\mathbf{w}^{(t)} = \sum_{k=1}^{K} w_k^{(t)} \nabla f_k(\mathbf{x}^{(t)})$.

4 MULTIPLE WASSERSTEIN GRADIENT DESCENT ALGORITHM

In this section, we introduce MWGraD, an iterative algorithm for solving the MODO problem (2). Specifically, we construct a flow of distributions $q^{(0)}$, $q^{(1)}$,..., $q^{(T)}$, starting with a simple distribution $q^{(0)}$, such as standard normal distribution, and progressively minimizing all the objective functionals $(F_1(q), F_2(q), ..., F_K(q))$ simultaneously. We begin by reformulating the MODO problem, then present the method for aggregating multiple Wasserstein gradients, followed by the detailed descriptions of algorithms.

4.1 REFORMULATION OF MODO AND MULTIPLE WASSERSTEIN GRADIENT AGGREGATION METHOD

To construct the sequence $\{q^{(t)}\}_{t\geq 0}$, we first reformulate the MODO problem (2) as follows. For any $q \in \mathcal{P}_2(\mathcal{X})$ and any tangent vector $s \in \mathcal{T}_q\mathcal{P}_2(\mathcal{X})$ at q, let $\gamma : [0,1] \to \mathcal{P}_2(\mathcal{X})$ be a curve satisfying $\gamma(0) = q$ and $\gamma'(0) = s$. By the definition of directional derivative, we have that

$$\lim_{h\to 0} \frac{1}{h}\left[F_k(\gamma(h)) - F_k(q)\right] = \int_{\mathcal{X}} \delta F_k(q)(\mathbf{x}) s(\mathbf{x}) d\mathbf{x}. \quad (4)$$

Let $\mathbf{u} : \mathcal{X} \to \mathcal{X}$ be a vector field, where $\mathbf{u} \in \mathcal{V}$, and $\mathcal{V}$ denotes the space of velocity fields $\mathbf{u}$. Assume that $\mathbf{u}$ satisfies the following elliptic equation

$$s(\mathbf{x}) + \mathrm{div}(q(\mathbf{x})\mathbf{u}(\mathbf{x})) = 0, \forall \mathbf{x} \in \mathcal{X}. \quad (5)$$

By the integration by parts, we obtain

$$\begin{aligned} &\lim_{h\to 0} \frac{1}{h}\left[F_k(\gamma(h)) - F_k(q)\right] = \\ &- \int_{\mathcal{X}} \delta F_k(q)(\mathbf{x}) \mathrm{div}(q(\mathbf{x})\mathbf{u}(\mathbf{x})) d\mathbf{x} \\ &= - \int_{\mathcal{X}} \mathtt{div}\left(\delta F_k(q)(\mathbf{x})\mathbf{u}(\mathbf{x}) q(\mathbf{x})\right) d\mathbf{x} \\ &+ \int_{\mathcal{X}} \langle \nabla \delta F_k(q)(\mathbf{x}), \mathbf{u}(\mathbf{x})\rangle q(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (6)$$

By the divergence theorem [Rudin, 2021], it holds that the first term on the right-hand side is equal to zero. Thus, we obtain that

$$\begin{aligned} &\lim_{h\to 0} \frac{1}{h}\left[F_k(\gamma(h)) - F_k(q)\right] \\ &= \int_{\mathcal{X}} \langle \nabla \delta F_k(q)(\mathbf{x}), \mathbf{u}(\mathbf{x})\rangle q(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (7)$$

We can similarly rewrite (7) for the opposite direction of the update as follows

$$\begin{aligned} &\lim_{h\to 0} \frac{1}{h}\left[F_k(q) - F_k(\gamma(h))\right] \\ &= \int_{\mathcal{X}} \langle \nabla \delta F_k(q)(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (8)$$

where $\mathbf{v} \in \mathcal{V}$ and $\mathbf{v}(\mathbf{x}) = -\mathbf{u}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$. In the MODO problem (2), there are multiple potentially conflicting objectives $\{F_k\}_{k=1}^K$. Inspired by [Désidéri, 2012], we aim to estimate a tangent vector $s^{(t)}$, for the t-th iteration, that minimizes each objective F_k. Based on (8), we reformulate the problem as follows: we aim to find $s^{(t)}$ that maximizes the minimum decrease across all the objectives

$$\begin{aligned} &\max_{s\in\mathcal{T}_{q^{(t)}}\mathcal{P}_2(\mathcal{X})} \min_{k\in[K]} \frac{1}{h}\left(F_k(q^{(t)}) - F_k(\gamma(h))\right) \\ &\approx \max_{\mathbf{v}\in\mathcal{V}} \min_{k\in[K]} \int_{\mathcal{X}} \langle \nabla \delta F_k(q^{(t)})(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (9)$$

where the approximation is based on (8) and $\gamma : [0,1] \to \mathcal{P}_2(\mathcal{X})$ is a curve satisfying that $\gamma(0) = q^{(t)}$ and $\gamma'(0) = s$. To regularize the update direction (i.e., vector field $\mathbf{v}$), we introduce a regularization term to (9) and solve for $s^{(t)}$ by optimizing

$$\begin{aligned} &\max_{\mathbf{v}\in\mathcal{V}} \min_{k\in[K]} \int_{\mathcal{X}} \langle \nabla \delta F_k(q^{(t)})(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x} - \\ &\frac{1}{2}\int_{\mathcal{X}} \|\mathbf{v}(\mathbf{x})\|_2^2 q^{(t)}(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (10)$$

The following theorem provides the solution to problem (10).

Theorem 1. *Problem (10) has a solution $\boldsymbol{v}^{(t)}$ as follows. For $\boldsymbol{x} \in \mathcal{X}$, we have that*

$$\boldsymbol{v}^{(t)}(\boldsymbol{x}) = \boldsymbol{V}^{(t)}(\boldsymbol{x})\boldsymbol{w}^* = \sum_{k=1}^{K} w_k^* \boldsymbol{v}_k^{(t)}(\boldsymbol{x}), \quad (11)$$

where $\boldsymbol{v}_k^{(t)}(\boldsymbol{x}) = \nabla\delta F_k(q^{(t)})(\boldsymbol{x})$ *for* $k \in [K]$, $\boldsymbol{V}^{(t)}(\boldsymbol{x}) = \left[\boldsymbol{v}_1^{(t)}(\boldsymbol{x}), \boldsymbol{v}_2^{(t)}(\boldsymbol{x}), ..., \boldsymbol{v}_K^{(t)}(\boldsymbol{x})\right]$, *and*

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \frac{1}{2}\int_{\mathcal{X}} \|\boldsymbol{V}^{(t)}(\boldsymbol{x})\boldsymbol{w}\|_2^2 q^{(t)}(\boldsymbol{x})d\boldsymbol{x}. \quad (12)$$

The proof follows from the method of Lagrange multipliers and is detailed in Appendix A. Note that the velocity used to update the current particles from $q^{(t)}$ is computed as a weighted sum of the velocities corresponding to each of K objective functionals. The weights are obtained by solving the min-norm oracle in Theorem 1.

4.2 ALGORITHM AND IMPLEMENTATION

To make the solution for $\mathbf{w}$ in the min-norm oracle (Theorem 1) computationally feasible, we approximate $q^{(t)}$ using m particles $\left\{\mathbf{x}_i^{(t)}\right\}_{i=1}^m$. Each particle is updated as $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \alpha\mathbf{v}^{(t)}(\mathbf{x}_i^{(t)})$, where α is the step size. The velocity fields $\mathbf{v}_k^{(t)} = \nabla\delta F_k(q^{(t)})$ for $k \in [K]$ need to be computed at each step, but exact computation is difficult, so we focus on approximation methods for $\mathbf{v}_k^{(t)}$ for two specific forms of $F_k(q)$.

Energy Functional. Consider $F_k(q)$ to be defined as

$$\begin{aligned} F_k(q) &= \int_{\mathcal{X}} g_k(\mathbf{x})q(\mathbf{x})d\mathbf{x} + \int_{\mathcal{X}} \log q(\mathbf{x})q(\mathbf{x})d\mathbf{x} \\ &= KL(q||\exp\{-g_k\}). \end{aligned} \quad (13)$$

This form of energy functional is commonly used in Bayesian learning, where the goal is to approximate the posterior distribution with q. It is straightforward to verify that the first variation of $F_k(q)$ is $\delta F_k(q)(\mathbf{x}) = g_k(\mathbf{x}) + \log q(\mathbf{x}) + 1$. Thus, the velocity $\mathbf{v}_k^{(t)}(\mathbf{x})$ can be computed as the gradient of $\delta F_k(q^{(t)})(\mathbf{x})$. However, directly applying the particle-based approximation is infeasible because the term $\log q^{(t)}(\mathbf{x})$ is undefined with discrete representations of $q^{(t)}(\mathbf{x})$. To address this issue, we present two commonly used techniques for approximating $\mathbf{v}_k^{(t)}(\mathbf{x})$, SVGD [Liu and Wang, 2016], and Blob methods [Carrillo et al., 2019], both of which are kernel-based.

We apply the idea of **SVGD** [Liu and Wang, 2016] to approximate $\mathbf{v}_k^{(t)}(\mathbf{x})$ with $\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x})$, given by

$$\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim q^{(t)}}\left[K(\mathbf{x},\mathbf{y})\left(\nabla g_k(\mathbf{y}) + \nabla\log q^{(t)}(\mathbf{y})\right)\right].$$

Applying integration by parts, we obtain

$$\begin{aligned} &\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}) = \\ &= \int_{\mathcal{X}} K(\mathbf{x},\mathbf{y})\nabla g_k(\mathbf{y})q^{(t)}(\mathbf{y})d\mathbf{y} + \int_{\mathcal{X}} K(\mathbf{x},\mathbf{y})\nabla q^{(t)}(\mathbf{y})d\mathbf{y} \\ &= \int_{\mathcal{X}} K(\mathbf{x},\mathbf{y})\nabla g_k(\mathbf{y})q^{(t)}(\mathbf{y})d\mathbf{y} - \int_{\mathcal{X}} \nabla_{\mathbf{y}}K(\mathbf{x},\mathbf{y})q^{(t)}(\mathbf{y})d\mathbf{y}. \end{aligned} \quad (14)$$

Thus, the particle approximation of $\mathbf{v}_k^{(t)}$ becomes

$$\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}) = \sum_{j=1}^m K(\mathbf{x},\mathbf{x}_j^{(t)})\nabla g_k(\mathbf{x}_j^{(t)}) - \sum_{j=1}^m \nabla_{\mathbf{x}_j^{(t)}} K(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)}) \quad (15)$$

We can also apply the idea of **Blob methods** [Carrillo et al., 2019] to smooth the second term of the energy functional by the kernel function K, and approximate the velocity field. Specifically, the particle approximation for $\mathbf{v}_k^{(t)}$ is given by

$$\begin{aligned} \tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}) = \nabla g_k(\mathbf{x}) &- \sum_{j=1}^m \nabla_{\mathbf{x}_j^{(t)}} K(\mathbf{x},\mathbf{x}_j^{(t)}) / \sum_{l=1}^m K(\mathbf{x}_j^{(t)},\mathbf{x}_l^{(t)}) \\ &- \sum_{j=1}^m \nabla_{\mathbf{x}_j^{(t)}} K(\mathbf{x},\mathbf{x}_j^{(t)}) / \sum_{l=1}^m K(\mathbf{x},\mathbf{x}_j^{(t)}). \end{aligned} \quad (16)$$

For the detailed derivation of the update, see Proposition 3.12 in [Carrillo et al., 2019].

Dissimilarity Functions. Let $F_k(q)$ be defined as a dissimilarity function D between q and the target distribution π_k, characterized by a set of samples. Common dissimilarity functions include KL divergence and JS divergence. In this case, estimating the first variation $\delta F_k(q)$ is not straightforward. To address this issue, following [Liu et al., 2021b], we assume that $F_k(q)$ has the variational form as follows:

$$F_k(q) = D(q,\pi_k) = \sup_{h_k\in\mathcal{H}} \{\mathbb{E}_{\mathbf{x}\sim q}[h_k(\mathbf{x})] - F_k^*(h_k)\}, \quad (17)$$

where $\mathcal{H}$ is a class of square-integrable functions on $\mathcal{X}$ with respect to the Lebesgue measure, and $F_k^* : \mathcal{H} \to \mathbb{R}$ is a convex conjugate functional of F_k, for $k \in [K]$. Note that the variational form of F_k involves a supremum over a function class $\mathcal{H}$. We restrict $\mathcal{H}$ to a subset of the square-integrable functions, such as a class of deep neural networks or a Reproducing kernel Hilbert space (RKHS), which allows for a numerically feasible maximization process. More importantly, it is shown in [Liu et al., 2021b] that the optimal solution h_k^* to the problem (17) is the first variation of F_k, i.e. $h_k^* = \delta F_k(q)$. As an example, when D_k is the KL divergence, its variational form is

$$KL(q,\pi_k) = \sup_{h_k\in\mathcal{H}} \left\{\mathbb{E}_{\mathbf{x}\sim q}[h_k(\mathbf{x})] - \log\mathbb{E}_{\mathbf{x}\sim\pi_k}\left[e^{h_k(\mathbf{x})}\right]\right\}. \quad (18)$$

We can parameterize h_k using a neural network h_{θ_k}, where θ_k denotes its parameters. The parameters can be estimated using stochastic gradient descent to maximize the following empirical objective function

$$\max_{\theta_k} \frac{1}{m}\sum_{i=1}^{m} h_{\theta_k}(\mathbf{x}_i^{(t)}) - \log \frac{1}{n}\sum_{j=1}^{n} \exp(h_{\theta_k}(\mathbf{y}_{k,j})), \quad (19)$$

where the target π_k is represented by the set of samples $\{\mathbf{y}_{k,j}\}_{j=1}^{n}$. The neural network $h_{\theta_k^*}$ after learning can be used to estimate the velocity corresponding to F_k by taking the gradient of the network with respect to its input:

$$\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}_i^{(t)}) = \nabla h_{\theta_k^*}(\mathbf{x}_i^{(t)}), \text{ for } i \in [m], k \in [K]. \quad (20)$$

Update of $\mathbf{w}^{(t)}$. Following the approach of [Zhang et al., 2024], instead of computing the exact solution of $\mathbf{w}$ to the optimization problem in Theorem 1, we approximate it by taking one step of gradient descent and then use the updated weights to aggregate the velocity fields. Specifically, we update each particle $\mathbf{x}_i^{(t)}$ along the following direction

$$\tilde{\mathbf{v}}^{(t)}(\mathbf{x}_i^{(t)}) = \sum_{k=1}^{K} w_k^{(t)} \tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}_i^{(t)}), \quad (21)$$

and update the weights $\mathbf{w}^{(t)}$ as follows

$$\mathbf{w}^{(t+1)} = \Pi_{\mathcal{W}}\left(\mathbf{w}^{(t)} - \beta\left[\sum_{i=1}^{m}\left(\tilde{\mathbf{V}}^{(t)}(\mathbf{x}_i^{(t)})\right)^{\top}\tilde{\mathbf{V}}^{(t)}(\mathbf{x}_i^{(t)})\right]\mathbf{w}^{(t)}\right), \quad (22)$$

where $\tilde{\mathbf{V}}^{(t)}(\mathbf{x}) = \left[\tilde{\mathbf{v}}_1^{(t)}(\mathbf{x}), \tilde{\mathbf{v}}_2^{(t)}(\mathbf{x}), ..., \tilde{\mathbf{v}}_K^{(t)}(\mathbf{x})\right]$ is the approximation of $\mathbf{V}^{(t)}(\mathbf{x})$, $\Pi_{\mathcal{W}}$ is the projection operator on the simplex $\mathcal{W}$, and β is the step size for updating $\mathbf{w}$. Taking all into account, we have Algorithm 1 (see Appendix B).

4.3 CONVERGENCE ANALYSIS

In this section, we analyze the convergence of MWGrad. We first define the Pareto stationary points in the space of probability distributions, and then characterize how MWGraD can converge to the Pareto stationary points.

Definition 4. $q \in \mathcal{P}_2(\mathcal{X})$ is a Pareto stationary point if

$$\min_{\mathbf{w}\in\mathcal{W}} \langle \texttt{grad}\mathbf{F}(q)\mathbf{w}, \texttt{grad}\mathbf{F}(q)\mathbf{w}\rangle_q = 0,$$

where $\texttt{grad}\mathbf{F}(q)(\mathbf{x}) = [\texttt{grad}F_1(q)(\mathbf{x}), \texttt{grad}F_2(q)(\mathbf{x}), ..., \texttt{grad}F_k(q)(\mathbf{x})]$, and $\texttt{grad}\mathbf{F}(q)(\mathbf{x})\mathbf{w} = \sum_{k=1}^{K} w_k \texttt{grad}F_k(q)(\mathbf{x})$. Further, we call q an ϵ-accurate Pareto stationary distribution if

$$\min_{\mathbf{w}\in\mathcal{W}} \langle \texttt{grad}\mathbf{F}(q)\mathbf{w}, \texttt{grad}\mathbf{F}(q)\mathbf{w}\rangle \leq \epsilon^2.$$

A detailed discussion on Pareto stationary distribution can be found in Appendix C.

As previously discussed, we need to approximate the true velocity field $\mathbf{v}_k^{(t)}$ with $\tilde{\mathbf{v}}_k^{(t)}$ for $k \in [K]$. The true Wasserstein gradient of F_k at $q^{(t)}$ is given by $\texttt{grad}F_k(q^{(t)}) = -\texttt{div}(q^{(t)}\mathbf{v}_k^{(t)})$, while its estimate is given by $-\texttt{div}(q^{(t)}\tilde{\mathbf{v}}_k^{(t)})$. The deviation between them is expressed as $\xi_k^{(t)} = -\texttt{div}(q^{(t)}(\tilde{\mathbf{v}}_k^{(t)} - \mathbf{v}_k^{(t)})) \in \mathcal{T}_{q^{(t)}}\mathcal{P}_2(\mathcal{X})$. The gradient error is then defined as

$$\epsilon_k^{(t)} = \langle \xi_k^{(t)}, \xi_k^{(t)}\rangle_{q^{(t)}} = \int_{\mathcal{X}} \|\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}) - \mathbf{v}_k^{(t)}(\mathbf{x})\|_2^2 q^{(t)}(\mathbf{x})\mathrm{d}\mathbf{x}, \quad (23)$$

In addition, we assume that the gradient error is upper bounded by a constant $\sigma > 0$ for $k \in [K]$ and $t \geq 0$.

Assumption 1 (**Wasserstein gradient error**). We assume that there is a constant $\sigma > 0$ such that

$$\langle \xi_k^{(t)}, \xi_k^{(t)}\rangle_{q^{(t)}} = \epsilon_k^{(t)} \leq \sigma^2.$$

We further make the following assumption on the functionals F_k to analyze the convergence of MWGraD.

Assumption 2 (**Geodesic smoothness**). We assume that F_k is geodesically ℓ_k-smooth with respect to the 2-Wasserstein distance, for $k \in [K]$, in the sense that for $\forall p, q \in \mathcal{P}_2(\mathcal{X})$

$$F_k(q) \leq F_k(p) + \langle \texttt{grad}F_k(p), \texttt{Exp}_p^{-1}(q)\rangle_p + \frac{\ell_k}{2}\cdot \mathcal{W}_2^2(p,q), \quad (24)$$

where $\texttt{Exp}_p$ denotes the exponential mapping, which specifies how to move p along a tangent vector on $\mathcal{P}_2(\mathcal{X})$ and $\texttt{Exp}_p^{-1}$ denotes its inversion mapping, which maps a point on $\mathcal{P}_2(\mathcal{X})$ to a tangent vector. We refer the readers to [Santambrogio, 2015] for more details. We then have the following theorem for the convergence of MWGraD.

Theorem 2. *Let Assumptions 1 and 2 hold, and $\epsilon > 0$ be a small constant. Set $\alpha \leq \mathcal{O}(\epsilon^2)$, $\beta \leq \mathcal{O}(\epsilon^2)$, $T \geq \max\left\{\Theta(\frac{1}{\alpha\epsilon^2}), \Theta(\frac{1}{\beta\epsilon^2})\right\}$. We then have that*

$$\begin{aligned}&\min_{0\leq t\leq T-1} \langle \texttt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}, \texttt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}\rangle_{q^{(t)}} \\ &\leq \frac{1}{T}\sum_{t=0}^{T-1} \langle \texttt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}, \texttt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}\rangle_{q^{(t)}} \\ &\leq \mathcal{O}(\epsilon^2) + 3\sigma^2.\end{aligned} \quad (25)$$

The formal version of Theorem 2 and its detailed proof can be found in Appendix D. Theorem 2 indicates that when we set $\alpha = \beta = \mathcal{O}(\epsilon^2)$ and $T \geq \Theta(\epsilon^{-4})$, we can find an $\sqrt{\mathcal{O}(\epsilon^2) + 3\sigma^2}$-accurate Pareto stationary point by running MWGraD. Furthermore, we can see that the squared norm of the convex combination of Wasserstein gradients in Theorem 2 is upper bounded by a sum of two terms. The first term $\mathcal{O}(\epsilon^2)$, which can be arbitrarily small, corresponds

to the convergence rate of MWGraD with the exact velocity fields. Meanwhile, the second term $3\sigma^2$ exhibits the effect of the gradient error caused by the approximation of velocity fields.

5 EXPERIMENTAL RESULTS

In this section, we present numerical experiments on both synthetic and real-world datasets to demonstrate the effectiveness of MWGraD. The code can be found at `https://github.com/haidnguyen0909/MWGraD`.

5.1 EXPERIMENTS ON SYNTHETIC DATASETS

Energy Functional. We consider $F_k(q)$ as an energy functional (13), related to sampling from multiple target distributions. Each target distribution is a mixture of two Gaussian distributions $\pi_k(\mathbf{x}) = \eta_{k1}\mathcal{N}(\mathbf{x}|\mu_{k1}, \Sigma_{k1}) + \eta_{k2}\mathcal{N}(\mathbf{x}|\mu_{k2}, \Sigma_{k2})$, $k = 1, 2, 3, 4$, where $\eta_{k1} = 0.7$, $\eta_{k2} = 0.3$ for $k = 1, 2, 3, 4$, the means $\mu_{11} = [4, -4]^\top$, $\mu_{12} = [0, 0.1]^\top$, $\mu_{21} = [-4, 4]^\top$, $\mu_{22} = [0, -0.1]^\top$, $\mu_{31} = [-4, -4]^\top$, $\mu_{32} = [0.1, 0]^\top$, $\mu_{41} = [4, 4]^\top$, $\mu_{42} = [-0.1, 0]^\top$, and the common covariance matrix Σ_{kj} is the identity matrix of size 2×2, for $k = 1, 2, 3, 4$ and $j = 1, 2$. The distribution q is represented by 50 particles, initially sampled from the standard distribution. We update the particles using MOO-SVGD [Liu et al., 2021c] and variants of our MWGraD, including MWGraD with SVGD (15) (denoted as MWGraD-SVGD) and Blob (16) (denoted as MWGraD-Blob) methods to approximate the velocities. Figure 1 shows a common high-density region around the origin.

We also consider the variational form (17) of KL divergence for sampling, where F_k is expressed as $KL(q, \pi_k)$. As noted earlier, the variational form of F_k can handle cases where π_k is characterized by samples. We can extend this form to the sampling task by expressing F_k as an energy functional by introducing the change of variables $h'_k(\mathbf{x}) = \exp\{h_k(\mathbf{x})\pi_k(\mathbf{x})/p(\mathbf{x})\}$, where $p(\mathbf{x})$ is a distribution that is easy to sample from, such as $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|0, \mathbf{I})$. Then, the variational form of $KL(q, \pi_k)$ can be rewritten as

$$KL(q, \pi_k) = \max_{h'_k \in \mathcal{H}^+} \left\{ \mathbb{E}_{\mathbf{x}\sim q}\left[\log\left(\frac{h'_k(\mathbf{x})p(\mathbf{x})}{\pi_k(\mathbf{x})}\right)\right] - \log \mathbb{E}_{\mathbf{x}\sim p}[h'_k(\mathbf{x})] \right\}, \tag{26}$$

where $\mathcal{H}^+$ is the space of positive functions. As we can use log of unnormalized density of π_k in the above optimization problem, h'_k can be estimated using samples drawn from q and p. We optimize (26) by parameterizing h'_k with a neural network with two layers, each of which has 50 neurons. We use the ReLU activation function in the last layer to guarantee the output of the neural network to be positive. Furthermore, h_k can be estimated from h'_k using: $h_k(\mathbf{x}) = \log h'_k(\mathbf{x}) + \log p(\mathbf{x}) - \log \pi(\mathbf{x})$. The parameters of the neural network are trained by running 20 steps of gradient ascent to optimize the objective (26). After training, the estimate h^*_k can be used to approximate the first variation of $F_k(q)$, i.e., $\tilde{\mathbf{v}}^{(t)}_k = \nabla h^*_k$. We denote this method by MWGraD-NN as we use the neural network to approximate the velocities. For all compared methods, we set the step size $\alpha = 0.0001$, while for variants of MWGraD, we set the step size $\beta = 0.001$. For MWGraD-SVGD and MWGraD-Blob, we use RBF as kernel K, i.e. $K(\mathbf{x}, \mathbf{y}) = \exp\left\{-\gamma\|\mathbf{x} - \mathbf{y}\|_2^2\right\}$, where we fix γ as 0.01. We run 2000 updates of particles.

Figure 1 shows the particles updated by MOO-SVGD and variants of MWGraD, including MWGraD-SVGD, MWGraD-Blob, and MWGraD-NN at selected iterations. We observe that particles by MOO-SVGD spread out and tend to cover all the modes of the targets, some of them even scatter along the trajectory due to the conflict in optimizing multiple objectives. In contrast, variants of MWGraD tend to cover the common high-density region with particles.

Dissimilarity Functions. Next we consider $F_k(q) = D(q, \pi_k)$, where D is the KL divergence or JS divergence, and π_k is characterized by a set of samples. We use the same target distribution used in the case of energy functional but we generate a set of samples to represent the target distribution instead of its energy functional. In our experiments, we generate 30 samples for each target distribution, so we have 120 samples for all objectives. For this case, MOO-SVGD, MWGraD-SVGD and MWGraD-Blob cannot be used. Thus we need to use the variational form of the KL and JS divergences. The detailed experimental settings are described in Appendix E. Figure 2 shows the particles updated by MWGraD at selected iterations, for two cases of KL divergence and JS divergence. We observe that MWGraD tend to cover the common high-density regions with particles, which is consistent with our observation in cases where $F_k(q)$ is represented by an energy functional.

5.2 EXPERIMENTS ON MULTI-TASK LEARNING

We follow the experimental settings outlined in [Phan et al., 2022] to verify the performance of MWGraD.

Multi-task Learning. We assume to have K prediction tasks and a training dataset $\mathbb{D}$. For each task $k \in [K]$, the model is represented by the parameter vector $\theta^k = [\mathbf{x}, \mathbf{z}^k]$, where $\mathbf{x}$ is the shared part of the model, and $\mathbf{z}^k$ is the task-specific non-shared part. The approach outlined in [Phan et al., 2022] is as follows. We maintain a set of m models $\theta_i = [\theta_i^k]_{k=1}^K$, where $i \in [m]$, and $\theta_i^k = [\mathbf{x}_i, \mathbf{z}_i^k]$. At each iteration, given the non-shared parts $\mathbf{z}_i^k$ for $i \in [m], k \in [K]$, we sample the shared part from the

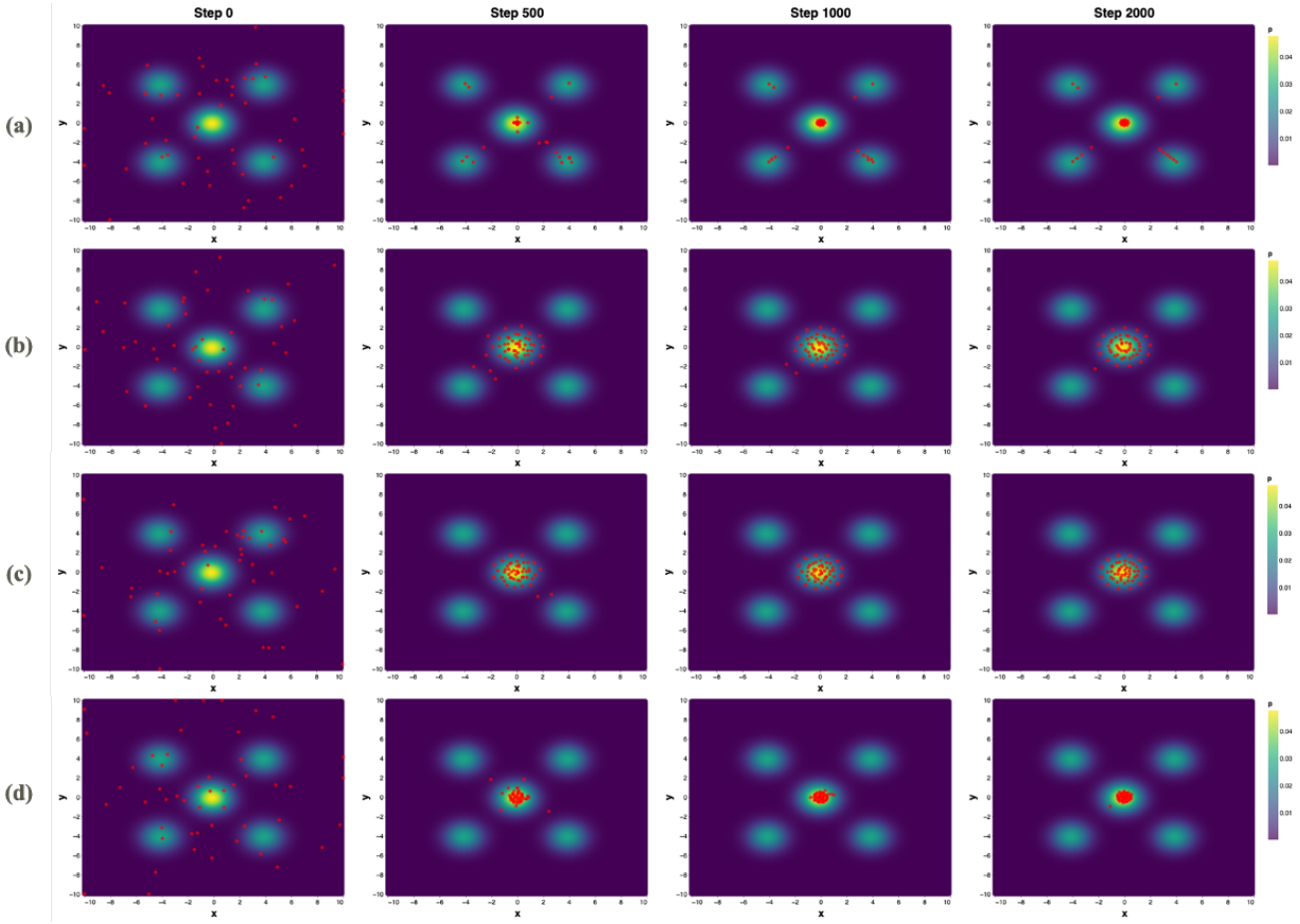

Figure 1: Sampling from multiple target distributions, where each target is a mixture of two Gaussians. These targets have a joint high-density region around the origin. Initially, 50 particles are sampled from the standard distribution, and then updated using (a) MOO-SVGD and variants of MWGraD, including (b) MWGraD-SVGD, (c) MWGraD-Blob and (d) MWGraD-NN. While MOO-SVGD tends to scatter particles across all the modes, MWGraD tends to move particles towards the joint high-density region.

multiple target distributions $p(\mathbf{x}|\mathbf{z}^k, \mathbb{D})$, $k \in [K]$. Here we apply MWGraD to sample the shared parts $[\mathbf{x}_i]_{i=1}^m$ from the multiple target distributions. Then, given the shared parts $[\mathbf{x}_i]_{i=1}^m$, for each task k, we update the corresponding non-shared parts $[\mathbf{z}_i^k]_{i=1}^m$ by sampling from the posterior distribution $p(\mathbf{z}^k|\mathbf{x}, \mathbb{D})$. This process corresponds to Bayesian sampling, for which we can use techniques, including SVGD, Blob methods or a neural network, as explained in the previous section. In our experiments, we use SVGD to sample the non-shared parts, while exploring different methods for sampling the shared parts, including MOO-SVGD, MT-SGD, MWGraD-SVGD, MWGraD-Blob and MWGraD-NN.

Datasets and Evaluation Metric. We validate the methods on three benchmark datasets: Multi-Fashion-MNIST [Sabour et al., 2017], Multi-MNIST [Phan et al., 2022], and Multi-Fashion [Phan et al., 2022]. Each of them consists of 120,000 training and 20,000 testing images from MNIST and FashionMNIST [Xiao et al., 2017] by overlaying an image on top of another. We compare our methods with the following baselinese: MOO-SVGD, MT-SGVD and MGDA, which achieved the best performance [Phan et al., 2022]. For the variants of MWGraD and MT-SGD, the reported results are from the ensemble prediction of five particle models. For MGDA, we train five particle models independently with different initializations and then ensemble these models. For evaluation metric, we compare the methods in terms of the accuracy for each task.

Results. Table 1 shows the ensemble accuracy of compared methods across three datasets. We observe that variants of MWGraD consistently outperform the other methods. For example, MWGraD-Blob achieves the best performance of 96.71%, 97.64% for Task 1 on the Multi-Fashion+MNIST and Multi-MNIST, resepectively, while MWGraD-NN achieves the best performance of 87.2% on the third dataset. For Task 2, MWGraD-Blob again achieves

Datasets	**Tasks**	MGDA	MOO-SVGD	MT-SGD	MWGraD -SVGD	**MWGraD** -Blob	MWGraD -NN
Multi-Fashion+MNIST	#1	94.4±0.6	94.8±0.4	96.2±0.3	95.7±0.4	**96.7**±0.5	95.9±0.4
	#2	85.5±0.5	85.6±0.2	87.8±0.6	88.9±0.6	**92.5**±0.4	88.2±0.3
Multi-MNIST	#1	93.4±0.4	93.1±0.3	94.4±0.5	94.5±0.4	**97.6**±0.2	**97.7**±0.5
	#2	91.8±0.6	91.2±0.2	92.9±0.5	93.2±0.6	**96.7**±0.5	95.5±0.4
Multi-Fashion	#1	84.1±0.8	83.8±0.8	84.9±0.6	85.1±0.7	86.8±0.3	**87.2**±0.4
	#2	83.3±0.4	83.1±0.3	84.6±0.5	84.3±0.4	**87.2**±0.5	85.3±0.6

Table 1: Experimental results on Multi-Fashion+MNIST, Multi-MNIST, and Multi-Fashion. We report the ensemble accuracy (higher is better) averaged over three independent runs with different initializations.

the best performance across all datasets, with average accuracies of 92.49%, 96.69% and 87.2%, respectively. Additionally, MWGraD-SVGD shows comparable performance to MT-SGD on all datasets. The reason is that both methods use the same velocity approximation, with the main difference being in the update of the weights $\mathbf{w}$: while MT-SGD solves for the optimal solution of $\mathbf{w}$ in Theorem 1, MWGraD-SVGD approximates the optimal solution by performing a gradient update (22). These experimental results clearly demonstrate the effectiveness of our proposed methods for the application of multi-task learning. See more details on the experiments in Appendix F.

6 CONCLUSION

In this paper, we have addressed the MODO problem, where the goal is to simultaneously minimize multiple functionals of probability distributions. We have introduced MWGraD, an iterative particle-based algorithm for solving MODO. At each iteration, it estimates the Wasserstein gradient for each objective using SGVD, Blob methods and neural networks. It then aggregates these gradients into one single Wasserstein gradient, which guides the updates of each particle. We have provided theoretical analyses and presented experiments on both synthetic and real-world datasets, demonstrating the effectiveness of MWGraD in identifying the joint high-density regions of objectives. In future work, we plan to explore MWGraD for real-world applications.

Acknowledgements

This work was supported in part by the International Collaborative Research Program of Institute for Chemical Research, Kyoto University (grant #2025-29), MEXT KAKENHI [grant number: 23K16939] (to D.H.N.), MEXT KAKENHI [grant numbers: 21H05027, 22H03645, 25H01144] (to H.M.), and JSPS KAKENHI [grant number: JP24H00685] (to A. N.).

References

José Antonio Carrillo, Katy Craig, and Francesco S Patacchini. A blob method for diffusion. *Calculus of Variations and Partial Differential Equations*, 58:1–53, 2019.

Changyou Chen and Ruiyi Zhang. Particle optimization in stochastic gradient mcmc. *arXiv preprint arXiv:1711.10927*, 2017.

Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020.

Ana Luísa Custódio, JF Aguilar Madeira, A Ismael F Vaz, and Luís Nunes Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21 (3):1109–1140, 2011.

Kalyanmoy Deb, Karthik Sindhya, and Jussi Hakanen. Multi-objective optimization. In *Decision sciences*, pages 161–200. CRC Press, 2016.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second international*

conference on learning representations, ICLR, volume 19, page 121, 2014.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021a.

Lewis Liu, Yufeng Zhang, Zhuoran Yang, Reza Babanezhad, and Zhaoran Wang. Infinite-dimensional optimization for zero-sum games via variational transport. In *International Conference on Machine Learning*, pages 7033–7044. PMLR, 2021b.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Xingchao Liu, Xin Tong, and Qiang Liu. Profiling pareto front with multi-objective stein variational gradient descent. *Advances in Neural Information Processing Systems*, 34:14721–14733, 2021c.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018.

Dai Hai Nguyen and Tetsuya Sakurai. Mirror variational transport: a particle-based algorithm for distributional optimization on constrained domains. *Machine Learning*, pages 1–25, 2023.

Dai Hai Nguyen and Tetsuya Sakurai. Moreau-yoshida variational transport: a general framework for solving regularized distributional optimization problems. *Machine Learning*, 113(9):6697–6724, 2024.

Dai Hai Nguyen and Koji Tsuda. On a linear fused gromov-wasserstein distance for graph structured data. *Pattern Recognition*, page 109351, 2023.

Dai Hai Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka. Learning subtree pattern importance for weisfeiler-lehman based graph kernels. *Machine Learning*, 110: 1585–1607, 2021.

Dai Hai Nguyen, Tetsuya Sakurai, and Hiroshi Mamitsuka. Wasserstein gradient flow over variational parameter space for variational inference. *arXiv preprint arXiv:2310.16705*, 2023.

Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. Got: an optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32, 2019.

Hoang Phan, Ngoc Tran, Trung Le, Toan Tran, Nhat Ho, and Dinh Phung. Stochastic multiple target sampling gradient descent. *Advances in neural information processing systems*, 35:22643–22655, 2022.

Walter Rudin. Principles of mathematical analysis. 2021.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

Philip S Thomas, Joelle Pineau, Romain Laroche, et al. Multi-objective spibb: Seldonian offline policy improvement with safety constraints in finite mdps. *Advances in Neural Information Processing Systems*, 34:2004–2017, 2021.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on learning theory*, pages 1617–1638. PMLR, 2016.

Qi Zhang, Peiyao Xiao, Kaiyi Ji, and Shaofeng Zou. On the convergence of multi-objective optimization under generalized smoothness. *arXiv preprint arXiv:2405.19440*, 2024.

A PROOF OF THEOREM 1

Proof. Problem (8) can be equivalently rewritten as the following optimization problem

$$\begin{aligned} &\min_{\mathbf{v},\mu} -\mu + \frac{1}{2}\int_{\mathcal{X}} \langle \mathbf{v}(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x} \\ &\text{such that } \mu \leq \int_{\mathcal{X}} \langle \mathbf{v}_k^{(t)}(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x}, \text{ for } k \in [K]. \end{aligned} \tag{27}$$

The Lagrange function is defined as follows

$$\mathcal{L}(\mathbf{v}, \mu, \mathbf{w}) = -\mu + \frac{1}{2}\int_{\mathcal{X}} \langle \mathbf{v}(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x} + \sum_{k=1}^{K} w_k \left(\mu - \int_{\mathcal{X}} \langle \mathbf{v}_k^{(t)}(\mathbf{x}), \mathbf{v}(\mathbf{x})\rangle q^{(t)}(\mathbf{x}) d\mathbf{x}\right),$$

where $\mathbf{w} = [w_1, w_2, ..., w_K]^\top \in \mathcal{W}$ are the dual variables corresponding to K constraints in (27). Taking the derivative of $\mathcal{L}$ with respect to $\mathbf{v}$ and setting it to zero, we obtain the optimal solution $\mathbf{v}^*$ as follows. For $\mathbf{x} \in \mathcal{X}$, we have that

$$\mathbf{v}^*(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathbf{v}_k^{(t)}(\mathbf{x}). \tag{28}$$

Substituting it to (27), we have the following optimization problem for $\mathbf{w}$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{2}\int_{\mathcal{X}} \|\sum_{k=1}^{K} w_k \mathbf{v}_k^{(t)}(\mathbf{x})\|_2^2 q^{(t)}(\mathbf{x}) d\mathbf{x},$$

which completes the proof. □

B ALGORITHM

Algorithm 1: Multi-objective Wasserstein Gradient Descent (MWGraD)

Input: Functionals $\{F_k\}$, number of particles m, number of iterations T, step sizes $\alpha > 0, \beta > 0$, weights $\mathbf{w}^{(0)}$.
Output: a set of m particles $\left\{\mathbf{x}_i^{(T)}\right\}_{i=1}^m$.
Sample m initial particles $\left\{\mathbf{x}_i^{(0)}\right\}_{i=1}^m$ from $\mathcal{N}(0, \mathbf{I}_d)$.
$t \leftarrow 0$
while $t < T$ **do**
  Estimate $\tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}_i^{(t)})$ by (15) or (16) or (20), for $i \in [m]$, $k \in [K]$
  Compute $\tilde{\mathbf{v}}^{(t)}(\mathbf{x}_i^{(t)}) \leftarrow \sum_{k=1}^K w_k^{(t)} \tilde{\mathbf{v}}_k^{(t)}(\mathbf{x}_i^{(t)})$, for $i \in [m]$
  Update $\mathbf{x}_i^{(t+1)} \leftarrow \mathbf{x}_i^{(t)} - \alpha \tilde{\mathbf{v}}^{(t)}(\mathbf{x}_i^{(t)})$, for $i \in [m]$
  Update $\mathbf{w}^{(t)}$ by (22)
  $t \leftarrow t + 1$
end

C A DETAILED DISCUSSION ON THE PARETO STATIONARY DISTRIBUTION

In this section, we examine the conditions under which a Pareto stationary distribution is also a Pareto optimal distribution. We begin with the following claim.

Claim 1 (Pareto optimality → Pareto stationarity). If a distribution q is not Pareto stationary, then there exists a descent direction that simultaneously improves all objective functions.

Proof. Suppose that at a point $q \in \mathcal{P}_2(\mathcal{X})$, there exists no convex combination of the Wasserstein gradients that sums to zero. That is, for all $\mathbf{w} \in \mathbb{R}^K$ with $\sum_{k=1}^K w_k = 1$, we have

$$\texttt{grad}\mathbf{F}(q)\,\mathbf{w} = \sum_{k=1}^{K} w_k \,\texttt{grad}F_k(q) \neq 0.$$

This implies that the gradients $\texttt{grad}F_k(q)$ for $k \in [K]$ all lie within a common open half-space. Therefore, there exists a tangent vector s lying in the opposite half-space such that

$$\langle s, \texttt{grad}F_k(q)\rangle_q < 0 \quad \text{for all } k \in [K].$$

This means that moving q along the direction s results in a simultaneous decrease of all objectives F_k. Hence, q cannot be a Pareto optimal distribution.

From the above argument, we conclude that Pareto optimality implies Pareto stationarity; in other words, Pareto optimality is a stronger condition.

Claim 2 (Pareto stationarity + geodesic strict convexity $\rightarrow$ Pareto optimality). If all objectives are (geodesically) strictly convex, and q is a Pareto stationary distribution, then q is a Pareto optimal distribution.

Proof. Assume q is a Pareto stationary distribution. Now suppose, for contradiction, that q is not Pareto optimal. Then there exists another distribution $q' \neq q$ such that

$$F_k(q') \leq F_k(q) \quad \text{for all } k \in [K], \quad \text{and} \quad F_j(q') < F_j(q) \text{ for some } j.$$

Let $\gamma : [0,1] \rightarrow \mathcal{P}_2(\mathcal{X})$ be a curve that connects q and q' and satisfies $\gamma(0) = q$ and $\gamma(1) = q'$. Let s be a tangent vector at q and s satisfies $s = \gamma'(0)$. As q is a Pareto stationary distribution, for any direction s in the tangent space at q, there is at least one objective functional F_k such that

$$\langle s, \texttt{grad}F_k(q)\rangle_q \geq 0.$$

Since F_k is strictly convex along the curve γ, we have

$$F_k(q') > F_k(q) + \langle \texttt{grad}F_k(q), s\rangle_q.$$

But $F_k(q') \leq F_k(q)$, which implies $\langle \texttt{grad}F_k(q), s\rangle_q < 0$. This contradicts the stationarity of q. Therefore, q must be Pareto optimal.

D FORMAL VERSION OF THEOREM 2 AND ITS PROOF

We first present the following lemma, which is useful to prove Theorem 2.

Lemma 3. *Assume that a functional G is geodesically L-smooth and $G(q) - G^* \leq C$ for $q \in \mathcal{P}_2(\mathcal{X})$, where $G^* = \min_{p \in \mathcal{P}_2(\mathcal{X})} G(p)$, and C is a constant. Then, there exists a constant M such that*

$$\langle \mathit{grad}G(q), \mathit{grad}G(q)\rangle_q \leq M^2 \tag{29}$$

Proof. Since G is geodesically L-smooth, we have for both $q, q' \in \mathcal{P}_2(\mathcal{X})$ that

$$G(q') \leq G(q) + \langle \texttt{grad}G(q), \texttt{Exp}_q^{-1}(q')\rangle_q + \frac{L}{2}\langle \texttt{Exp}_q^{-1}(q'), \texttt{Exp}_q^{-1}(q')\rangle_q \tag{30}$$

By choosing $q' = \texttt{Exp}_q(-\frac{1}{L}\texttt{grad}G(q))$, we have

$$G(q') \leq G(q) - \frac{1}{L}\langle \texttt{grad}G(q), \texttt{grad}G(q)\rangle_q + \frac{1}{2L}\langle \texttt{grad}G(q), \texttt{grad}G(q)\rangle_q \tag{31}$$

So, we have

$$\langle \texttt{grad}G(q), \texttt{grad}G(q)\rangle_q \leq 2L\left(G(q) - G(q')\right) \leq 2L\left(G(q) - G^*\right) \leq 2LC \tag{32}$$

By setting $M^2 = 2LC$, we conclude that

$$\langle \texttt{grad}G(q), \texttt{grad}G(q)\rangle_q \leq M^2 \tag{33}$$

□

Let $b_1 > 0$, $b_2 > 0$, $b_3 > 0$ and $C > 0$ be some constants such that

$$\Delta + b_1 + b_2 + b_3 \leq C.$$

Define $L = \max_{k\in[K]} \{\ell_k\}$ and $M = 2LC$. Then we have the following convergence theorem for Algorithm 1.

Theorem 4. *Let Assumptions 1 and 2 hold. Set α, β, T as follows:*

$$\begin{aligned}
\beta &\leq \frac{\epsilon^2}{24(KM^4 + 2KM^2\sigma^2 + K\sigma^4)} \\
\alpha &\leq \min\left\{\frac{4b_1}{3T\sigma^2}, \beta b_2, \frac{b_3}{2\beta T(KM^4 + 2KM\sigma^2 + K\sigma^4)}\right\} \\
T &\geq \max\left\{\frac{12\Delta}{\alpha\epsilon^2}, \frac{12}{\beta\epsilon^2}\right\}.
\end{aligned}$$

We then have that

$$\frac{1}{T}\sum_{t=0}^{T-1} \langle \mathtt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}, \mathtt{grad}\boldsymbol{F}(q^{(t)})\boldsymbol{w}^{(t)}\rangle_{q^{(t)}} \leq \epsilon^2 + 3\sigma^2. \tag{34}$$

Proof. As shown in Lemma 3, a bounded functional value implies a bounded Wasserstein gradient norm. In the following, we show that, with the properly selected parameters in Theorem 4, the functional value is bounded by induction.

For the base case, it is trivial to see that $F_k(q^{(0)}) - F_k^* \leq \Delta \leq C$ for $k \in [K]$. Now we assume that for any $k \in [K]$ and $t \leq \tau < T$, we have that $F_k(q^{(t)}) - F_k^* \leq C$ holds. We then prove that $F_k(q^{(\tau+1)}) - F_k^* \leq C$ holds for any $k \in [K]$.

For any $k \in [K]$, $t \leq \tau$, we have $F_k(q^{(t)}) - F_k^* \leq C$, which implies that $\langle \mathtt{grad}F_k(q^{(t)}), \mathtt{grad}F_k(q^{(t)})\rangle_{q^{(t)}} \leq M$. Since F_k is ℓ_k-smooth, it follows that

$$F_k(q^{(\tau+1)}) \leq F_k(q^{(\tau)}) + \langle \mathtt{grad}F_k(q^{(\tau)}), \mathtt{Exp}^{-1}_{q^{(\tau)}}(q^{(\tau+1)})\rangle + \frac{\ell_k}{2}\langle \mathtt{Exp}^{-1}_{q^{(\tau)}}(q^{(\tau+1)}), \mathtt{Exp}^{-1}_{q^{(\tau)}}(q^{(\tau+1)})\rangle_{q^{(\tau)}}.$$

Since $\mathtt{Exp}^{-1}_{q^{(t)}}(q^{(t+1)}) = (\mathtt{grad}\mathbf{F}(q^{(t)}) + \xi^{(t)})\mathbf{w}^{(t)}$, and $\ell_k \leq \max_{i\in[K]} \ell_i = L$, it follows that

$$\begin{aligned}
F_k(q^{(\tau+1)}) \leq\ & F_k(q^{(\tau)}) - \alpha\langle \mathtt{grad}F_k(q^{(\tau)}), (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\
&+ \frac{L\alpha^2}{2}\langle (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}, (\mathtt{grad}\mathbf{F}(q^{(t)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}}.
\end{aligned} \tag{35}$$

For any $\mathbf{w} \in \mathcal{W}$, we have that

$$\begin{aligned}
\mathbf{F}(q^{(\tau+1)})\mathbf{w} \leq\ & \mathbf{F}(q^{(\tau)})\mathbf{w} - \alpha\langle \mathtt{grad}F(q^{(\tau)})\mathbf{w}, (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(t)}\rangle_{q^{(\tau)}} \\
&+ \frac{L\alpha^2}{2}\langle (\mathtt{grad}F(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}, (\mathtt{grad}F(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\
\leq\ & \mathbf{F}(q^{(\tau)})\mathbf{w} - \alpha\langle \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} - \alpha\langle \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \xi^{(t)}\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\
&+ \alpha\langle \mathtt{grad}\mathbf{F}(p^{(\tau)})(\mathbf{w}^{(\tau)} - \mathbf{w}), (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\
&+ \frac{L}{2}\alpha^2\langle (\mathtt{grad}\mathbf{F}(p^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}, (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}}.
\end{aligned} \tag{36}$$

By the basic inequality $(a+b)^2 \leq 2(a^2 + b^2)$, we have that

$$\begin{aligned}
\langle (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}, (\mathtt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \leq\ & 2\langle \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\
&+ 2\langle \xi^{(t)}\mathbf{w}^{(\tau)}, \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}}, \\
\langle \mathtt{grad}\mathbf{F}(q^{\tau})\mathbf{w}^{(\tau)}, \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle \leq\ & \frac{1}{2}\langle \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \mathtt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} + \frac{1}{2}\langle \xi^{(t)}\mathbf{w}^{(\tau)}, \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}}.
\end{aligned} \tag{37}$$

Thus, combining (36), (37), we have that

$$\begin{aligned}\mathbf{F}(q^{(\tau+1)})\mathbf{w} \leq \mathbf{F}(q^{(\tau)})\mathbf{w} &- \alpha(\frac{1}{2} - L\alpha)\langle \texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\ &+ \alpha\left(L\alpha + \frac{1}{2}\right)\langle \xi^{(\tau)}\mathbf{w}^{(\tau)}, \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}} \\ &+ \alpha\langle \texttt{grad}\mathbf{F}(q^{(\tau)})(\mathbf{w}^{(\tau)} - \mathbf{w}), (\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(\tau)}}.\end{aligned} \tag{38}$$

Based on the update of $\mathbf{w}^{(\tau)}$, we have that

$$\mathbf{w}^{(\tau+1)} = \Pi_{\mathcal{W}}\left(\mathbf{w}^{(\tau)} - \beta(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\right), \tag{39}$$

where we have used the following identity for notational simplicity

$$(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)}) = \int_{\mathcal{X}} \tilde{\mathbf{V}}^{(\tau)}(\mathbf{x})^{\top}\tilde{\mathbf{V}}^{(\tau)}(\mathbf{x})q^{(\tau)}(\mathbf{x})d\mathbf{x},$$

and $\tilde{\mathbf{V}}^{(\tau)}(\mathbf{x}) = \left[\tilde{\mathbf{v}}_1^{(\tau)}(\mathbf{x}), \tilde{\mathbf{v}}_2^{(\tau)}(\mathbf{x}), ..., \tilde{\mathbf{v}}_K^{(\tau)}(\mathbf{x})\right]$ is the approximation of $\mathbf{V}^{(\tau)}(\mathbf{x})$.

Applying the non-expansiveness of the projection, it follows that

$$\begin{aligned}\|\mathbf{w}^{(\tau+1)} - \mathbf{w}\|_2^2 &= \|\Pi_{\mathcal{W}}\left(\mathbf{w}^{(\tau)} - \beta(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\right) - \mathbf{w}\|_2^2 \\ &\leq \|\mathbf{w}^{(\tau)} - \mathbf{w}\|_2^2 + \beta^2\|(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\|_2^2 \\ &- 2\beta\langle(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})(\mathbf{w}^{(\tau)} - \mathbf{w}), (\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} \\ &= \|\mathbf{w}^{(\tau)} - \mathbf{w}\|_2^2 + \beta^2\|(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\|_2^2 \\ &- 2\beta\langle\texttt{grad}\mathbf{F}(q^{(\tau)})(\mathbf{w}^{(\tau)} - \mathbf{w}), (\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} \\ &- 2\beta\langle\xi^{(\tau)}(\mathbf{w}^{(\tau)} - \mathbf{w}), \texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)} + \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(t)}}.\end{aligned} \tag{40}$$

Thus it follows that

$$\begin{aligned}\langle\texttt{grad}\mathbf{F}(q^{(\tau)})(\mathbf{w}^{(\tau)} - \mathbf{w}), (\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} &\leq \frac{1}{2\beta}\left(\|\mathbf{w}^{(\tau)} - \mathbf{w}\|_2^2 - \|\mathbf{w}^{(\tau+1)} - \mathbf{w}\|_2^2\right) \\ &- \langle\xi^{(t)}(\mathbf{w}^{(\tau)} - \mathbf{w}), \texttt{grad}\mathbf{F}(q^{(\tau)})w^{(\tau)} + \xi^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} \\ &+ \frac{\beta}{2}\|(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})^{\top}(\texttt{grad}\mathbf{F}(q^{(\tau)}) + \xi^{(\tau)})\mathbf{w}^{(\tau)}\|_{q^{(t)}}^2 \\ &\leq \frac{1}{2\beta}\left(\|\mathbf{w}^{(\tau)} - \mathbf{w}\|_2^2 - \|\mathbf{w}^{(\tau+1)} - \mathbf{w}\|_2^2\right) \\ &- \langle\xi^{(t)}(\mathbf{w}^{(\tau)} - \mathbf{w}), \texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)} + \delta^{(\tau)}\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} + 2\beta KM^4 + 4\beta KM^2\sigma^2 + 2\beta K\sigma^4,\end{aligned} \tag{41}$$

where the last inequality holds due to Assumption 2. Then plugging (41) into (38), we can show that

$$\begin{aligned}\mathbf{F}(q^{(\tau+1)})\mathbf{w} - \mathbf{F}(q^{(\tau)})\mathbf{w} \leq -\alpha\left(\frac{1}{2} - L\alpha\right)\langle\texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}, \texttt{grad}\mathbf{F}(q^{(\tau)})\mathbf{w}^{(\tau)}\rangle_{q^{(t)}} + \alpha(\frac{1}{2} + L\alpha)\sigma^2 \\ + \frac{\alpha}{2\beta}\left(\|\mathbf{w}^{(\tau)} - \mathbf{w}\|_2^2 - \|\mathbf{w}^{(\tau+1)} - \mathbf{w}\|_2^2\right) \\ + \alpha\sigma(M + \sigma) + 2\alpha\beta KM^4 + 4\alpha\beta KM^2\sigma^2 + 2\alpha\beta K\sigma^4.\end{aligned} \tag{42}$$

Taking sum of (42) from $t = 0$ to τ, for any $\mathbf{w} \in \mathcal{W}$, we have that

$$\begin{aligned}\mathbf{F}(q^{(\tau+1)})\mathbf{w} - \mathbf{F}(q^{(0)})\mathbf{w} &\leq -\frac{\alpha}{4}\sum_{t=0}^{\tau}\langle\texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}, \texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}\rangle_{q^{(t)}} + \frac{3}{4}\alpha T\sigma^2 \\ &+ \frac{\alpha}{2\beta}\left(\|\mathbf{w}^{(0)} - \mathbf{w}\|_2^2 - \|\mathbf{w}^{(\tau+1)} - \mathbf{w}\|_2^2\right) \\ &+ 2\alpha\beta KM^4T + 4\alpha\beta KM^2\sigma^2T + 2\alpha\beta K\sigma^4T \\ &\leq \frac{3}{4}\alpha T\sigma^2 + \frac{\alpha}{\beta} + 2\alpha\beta T(KM^4 + 2KM^2\sigma^2 + K\sigma^4)\end{aligned} \tag{43}$$

where the inequality is due to $\tau < T$, $\alpha L \leq 1/4$ and Assumption 2. Thus, we have that

$$\mathbf{F}(q^{(\tau+1)})\mathbf{w} - \mathbf{F}^*\mathbf{w} \leq \underbrace{\mathbf{F}(q^{(0)})\mathbf{w} - \mathbf{F}^*\mathbf{w}}_{\leq \Delta} + \underbrace{\frac{3}{4}\alpha T\sigma^2}_{\leq b_1} + \underbrace{\frac{\alpha}{\beta}}_{\leq b_2} + \underbrace{2\alpha\beta T(KM^4 + 2KM^2\sigma^2 + K\sigma^4)}_{\leq b_3} \leq C. \tag{44}$$

Now we finish the induction step and can show that $F_k(q^{(\tau+1)}) - F_k^* \leq C$ for all $k \in [K]$. Furthermore, according to (43), for $\tau = T-1$, we have that

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\langle \texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}, \texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}\rangle_{q^{(t)}} &\leq \frac{4(\mathbf{F}(q^{(0)})\mathbf{w} - \mathbf{F}^*\mathbf{w})}{\alpha T} + \frac{4}{\beta T} + 8\beta(KM^4 + 2KM^2\sigma^2 + K\sigma^4) + 3\sigma^2 \\ &\leq \epsilon^2 + 3\sigma^2,\end{aligned} \tag{45}$$

which completes the proof. □

Remarks. Although there is a circular dependency among the parameters β, α, and T, the required conditions in Theorem 4 can still be satisfied simultaneously. Specifically, if we choose

$$\alpha = \mathcal{O}(\epsilon^2), \quad \beta = \mathcal{O}(\epsilon^2), \quad \alpha T = \Theta(\epsilon^{-2}), \quad \beta T = \Theta(\epsilon^{-2}), \quad \text{and} \quad T = \Theta(\epsilon^{-4}),$$

then all the necessary assumptions hold. To make this construction concrete, consider the following example.

Let us set

$$\beta = a_1\epsilon^2, \quad \alpha = a_2\epsilon^2, \quad \text{and} \quad a_4\epsilon^{-4} \leq T \leq a_3\epsilon^{-4},$$

where the constants $a_1, a_2, a_3, a_4 > 0$ are chosen such that inequality (44) is satisfied. Specifically, we require

$$\begin{aligned}&0 < a_1 \leq \frac{1}{24(KM^4 + 2KM^2\sigma^2 + K\sigma^4)}, \\ &0 < a_2 \leq a_1 b_2, \\ &0 < a_3 \leq \min\left(\frac{4b_1\epsilon^4}{3\sigma^2}, \frac{b_3\epsilon^2}{2a_1(KM^4 + 2KM^2\sigma^2 + K\sigma^4)}\right).\end{aligned}$$

With this choice of constants, it is straightforward to verify that inequality (44) holds.

Next, we choose a_4 such that $0 < a_4 \leq a_3$, ensuring that T satisfies

$$a_4\epsilon^{-4} \leq T \leq a_3\epsilon^{-4}.$$

It follows that

$$\begin{aligned}T\beta &\geq a_1 a_4 \epsilon^{-2}, \\ T\alpha &\geq a_2 a_4 \epsilon^{-2}.\end{aligned}$$

We now analyze the upper bound in inequality (45):

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\left\langle \texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}, \texttt{grad}\mathbf{F}(q^{(t)})\mathbf{w}^{(t)}\right\rangle_{q^{(t)}} &\leq \frac{4\Delta}{a_2 a_4}\epsilon^2 + \frac{4}{a_1 a_4}\epsilon^2 \\ &\quad + 8(KM^4 + 2KM^2\sigma^2 + K\sigma^4)a_1\epsilon^2 + 3\sigma^2 \\ &= \mathcal{O}(\epsilon^2) + 3\sigma^2.\end{aligned} \tag{46}$$

Therefore, we have constructed a sequence of parameter choices (a_1, a_2, a_3, a_4) such that all theoretical conditions are met. Moreover, the norm of the convex combination of Wasserstein gradients is bounded by a term of order $\mathcal{O}(\epsilon^2)$—reflecting the convergence rate of MWGraD with exact velocity fields—plus a $3\sigma^2$ term, which accounts for the error introduced by approximate gradient computations.

As established in Theorem 4, which analyzes the convergence of MWGraD, the squared norm of the convex combination of Wasserstein gradients is bounded by two terms: one of order ϵ^2, and the other proportional to the approximation error σ^2. This result suggests that, in order to avoid convergence to a non–Pareto stationary point, it is important to minimize the approximation error σ^2 as much as possible. For instance, in the MWGraD-NN variant, where Wasserstein gradients are approximated using neural networks, we can reduce σ^2 by increasing the capacity of the networks—e.g., by adding more layers or neurons. However, this improvement in accuracy comes at the cost of increased computational complexity. It is also worth noting that Theorem 4 does not rely on any geodesic convexity assumptions. Therefore, the convergence guarantees extend naturally to non-convex settings.

E ADDITIONAL EXPERIMENTAL RESULTS ON SYNTHETIC DATASETS

We consider $F_k(q) = D(q, \pi_k)$, where D is the KL divergence or JS divergence, and π_k is characterized by a set of samples. We use the same target distribution used in the case of energy functional but we generate a set of samples to represent the target distribution instead of its energy functional. In our experiments, we generate 30 samples for each target distribution, so we have 120 samples for all objectives. For this case, MOO-SVGD, MWGraD-SVGD and MWGraD-Blob cannot be used. Thus we need to use the variational form of the KL and JS divergences. For KL divergence, we use its variational form (18). For JS divergence, as shown in [Nguyen and Sakurai, 2023], the variational form of the JS divergence is as follows:

$$JS(q, \pi_k) = \sup_{h_k \in \mathcal{H}^c} \{\mathbb{E}_{\mathbf{x}\sim q}\left[h_k(\mathbf{x})\right] - JS^*(h_k)\} \tag{47}$$

where $JS^*(h_k) = -\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim\pi}\left[\log\left(1 - 2e^{2h_k(\mathbf{x})}\right)\right]$, and $\mathcal{H}^c$ is the space of function h that satisfies: $h(\mathbf{x}) < 1/2\log(1/2)$ for all $\mathbf{x} \in \mathcal{X}$. We introduce the following change of variable: $h'_k(\mathbf{x}) = 1 - 2e^{2h_k(\mathbf{x})}$. It is easy to verify that $0 < h'_k(\mathbf{x}) < 1$ for all $\mathbf{x} \in \mathcal{X}$. Then, the variational form of JS divergence can be rewritten as:

$$\sup_{h'_k \in \mathcal{H}'} \{\mathbb{E}_{\mathbf{x}\sim q}\left[\log\left(1 - h'_k(\mathbf{x})\right)\right] + \mathbb{E}_{\mathbf{y}\sim\pi}\left[\log(h'_k(\mathbf{x}))\right]\}$$

where $\mathcal{H}'$ is the space of functions whose outputs are in between 0 and 1. As we have access to samples drawn from q and π, we can estimate h'_k and then estimate h_k using: $h_k(\mathbf{x}) = \frac{1}{2}\log\left(\frac{1-h'_k(\mathbf{x})}{2}\right)$ for all $\mathbf{x} \in \mathcal{X}$. For the function h'_k, we also use a neural network with two layers, each of which has 50 neurons, to parameterize it. To guarantee the outputs of h'_k to be in $(0, 1)$, we use the sigmoid activation function in the last layer. We set the same step sizes as in the previous experiments, i.e., $\alpha = 0.0001, \beta = 0.001$. We run 2000 updates of particles.

Figure 2 shows the particles updated by MWGraD at selected iterations, for two cases of KL divergence and JS divergence. We observe that MWGraD tend to cover the common high-density regions with particles, which is consistent with our observation in cases where $F_k(q)$ is represented by an energy functional.

F ADDITIONAL EXPERIMENTAL RESULTS ON REAL-WORLD DATASETS

In this section, we first examine the time complexity of MWGraD. We conduct experiments on three datasets—Multi-MNIST, Multi-Fashion, and Multi-Fashion+MNIST—to evaluate the runtime (in seconds) per training epoch for various MWGraD variants and baselines, including MOO-SVGD and MT-SGD. Table 2 presents the average runtimes of each method, computed over five runs.

We observe that MWGraD-SVGD and MWGraD-Blob have comparable runtimes to MT-SGD because they involve approximating Wasserstein gradients using kernel matrices, which is computationally efficient. In contrast, MWGraD-NN has a slower runtime due to the need to train a neural network to approximate the Wasserstein gradients of the objective functions at each iteration, which can be more time-consuming. Additionally, MOO-SVGD is slower than our methods, as it requires solving separate quadratic problems for each particle (each network). Specifically, MOO-SVGD is 9 seconds slower than MWGraD-SVGD and MWGraD-Blob, and 1 second slower than MWGraD-NN.

Second, we conduct experiments to verify the importance of key modules of MWGraD as follows. We compare the effectiveness of approximation methods, including MWGraD-SVGD, MWGraD-Blob and MWGraD-NN, for the Wasserstein gradients of objective functionals.

Table 3 (extracted from Table 1) shows the ensemble accuracy numbers of compared approximation methods over three independent runs with different initializations. We observe that MWGraD-Blob achieves the best performances of 96.71%,

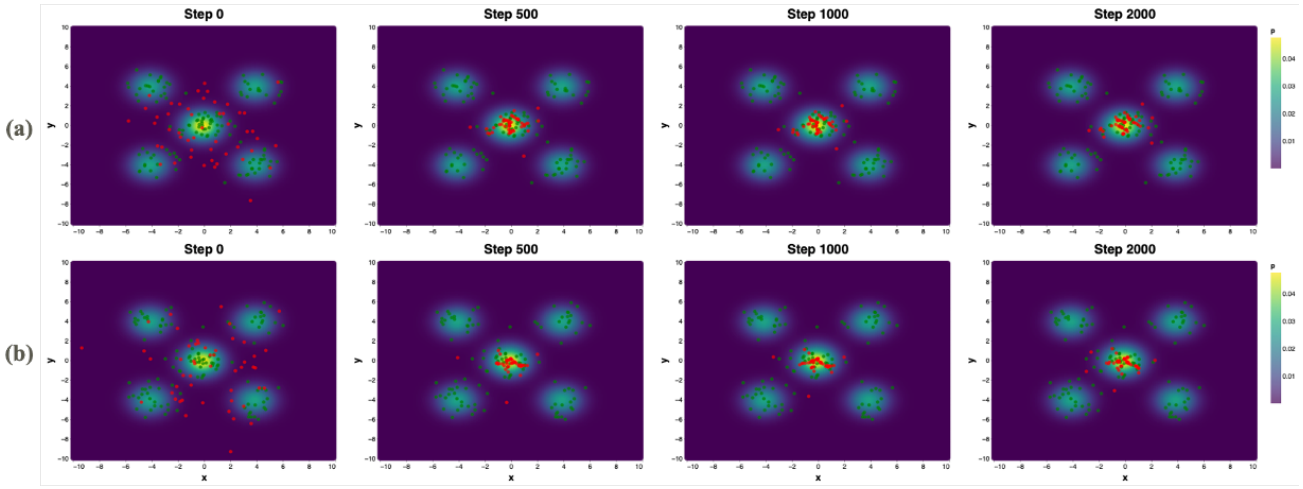

Figure 2: The MODO problem on synthetic dataset. There are four objectives, each of which is represented by 30 particles (green points) randomly drawn from a mixture of two Gaussian distributions. The dissimilarity function D is defined as the (a) KL divergence or (b) JS divergence. The objectives have a common high-density of particles. Initially 50 particles (red points) are sampled from the standard distribution to represent q, and then updated using MWGraD-NN. In both cases of divergences, MWGraD-NN drives the particles to the joint high density region around the origin. Note that, in this toy experiments, MWGraD-SVGD, MWGraD-Blob, MOO-SVGD cannot be used as the objective functions are not the form of energy functionals.

Table 2: Average runtime of compared methods on real-world datasets.

Runtime	Multi-MNIST	Multi-Fashion	Multi-Fashion+MNIST
MOO-SVGD	54.4 ± 0.3	55.1 ± 0.6	58.2 ± 0.3
MT-SGD	48.1 ± 0.5	46.3 ± 0.3	44.1 ± 0.7
MWGraD-SVGD	47.2 ± 0.9	45.2 ± 0.8	45.5 ± 0.6
MWGraD-Blob	45.5 ± 0.6	46.2 ± 0.9	45.1 ± 0.5
MWGraD-NN	53.3 ± 0.6	54.3 ± 0.5	56.8 ± 0.8

97.64% for task 1 on Multi-Fashion+MNIST and Multi-MNIST, respectively, while MWGraD-NN achieves the best performance of 97.7% and 87.2% on Multi-MNIST and Multi-Fashion. For task 2, MWGraD-Bob again achieves the best performance across all datasets, with average accuracies of 92.5%, 96.7% and 87.2%, respectively.

Furthermore, we also explore the importance of the weight update step, and verify how the performance would change if the uniform weight were used to combine multiple Wasserstein gradients. Tables 4, 5 and 6 shown below compare the accuracies of task 1, task 2 and the average for each variant of MWGraD with and without the weight update step (we use the suffix "uniform" to indicate that the uniform weights are used). We observe that when we remove the weight update step, the performances of variants decrease. For example, for experiments on Multi-Fashion, the performance of MWGraD-Blob decreases from 87% to 84.9% (roughly 2%), that of MWGraD-SVGD decreases from 84.7% to 83.7% (1%) and that of MWGraD-NN decreases from 86.3% to 85.5% (0.7%). These experiments demonstrate the importance of the weight update step in our proposed methods.

Table 3: Average accuracies of approximation methods, including MWGraD-SVGD, MWGraD-Blob and MWGraD-NN, for the Wasserstein gradients of objective functionals on real-world datasets. .

Datasets	**Tasks**	MWGraD-SVGD	MWGraD-Blob	MWGraD-NN
Multi-Fashion+MNIST	1	95.7±0.4	**96.7**±0.5	95.9±0.4
	2	88.9±0.6	**92.5**±0.4	88.2±0.3
Multi-MNIST	1	94.5±0.4	**97.6**±0.2	**97.7**±0.5
	2	93.2±0.6	**96.7**±0.5	95.5±0.4
Multi-Fashion	1	85.1±0.7	86.8±0.3	**87.2**±0.4
	2	84.3±0.4	**87.2**±0.5	85.3±0.6

Table 4: The accuracies of task 1, task 2 and the average for each variant of MWGraD with and without the weight update step (suffix 'uniform' indicates that the uniform weights are used). Dataset: **Multi-Fashion+MNIST**.

Task	MWGraD-SVGD	MWGraD-SVGD -uniform	MWGraD-Blob	MWGraD-Blob -uniform	MWGraD-NN	MWGraD-NN -uniform
1	95.7±0.4	94.1±0.9	96.7±0.5	94.2±0.6	95.9±0.4	95.5±0.3
2	88.9±0.6	87.2±0.8	92.5±0.4	92.8±0.8	88.2±0.3	87.9±0.7
Avg	92.3	90.7	94.6	93.5	92.1	91.7

Table 5: The accuracies of task 1, task 2 and the average for each variant of MWGraD with and without the weight update step. Dataset: **Multi-MNIST**.

Task	MWGraD-SVGD	MWGraD-SVGD -uniform	MWGraD-Blob	MWGraD-Blob -uniform	MWGraD-NN	MWGraD-NN -uniform
1	94.5±0.4	94.6±0.6	97.6±0.2	95.9±0.5	97.7±0.5	97.1±0.6
2	93.2±0.6	93.5±0.4	96.7±0.7	97.2±0.6	95.5±0.4	93.9±0.5
Avg	93.9	94.1	97.2	96.6	96.6	95.5

Table 6: The accuracies of task 1, task 2 and the average for each variant of MWGraD with and without the weight update step. Dataset: **Multi-Fashion**.

Task	MWGraD-SVGD	MWGraD-SVGD -uniform	MWGraD-Blob	MWGraD-Blob -uniform	MWGraD-NN	MWGraD-NN -uniform
1	85.1±0.7	85.2±0.4	86.8±0.4	84.1±0.7	87.2±0.4	85.9±0.3
2	84.3±0.4	82.1±0.6	87.2±0.5	85.8±0.6	85.3±0.6	85.1±0.8
Avg	84.7	83.7	87.0	84.9	86.3	85.5