Profiling Pareto Front With Multi-Objective Stein Variational Gradient Descent

Anonymous Author(s)
Affiliation
Address
email

Abstract

Finding diverse and representative Pareto solutions from the Pareto front is a key challenge in multi-objective optimization (MOO). In this work, we propose a novel gradient-based algorithm for profiling Pareto front by using Stein variational gradient descent (SVGD). We also provide a counterpart of our method based on Langevin dynamics. Our methods iteratively update a set of points in a parallel fashion to push them towards the Pareto front using multiple gradient descent, while encouraging the diversity between the particles by using the repulsive force mechanism in SVGD, or diffusion noise in Langevin dynamics. Compared with existing gradient-based methods that require predefined preference functions, our method can work efficiently in high dimensional problems, and can obtain more diverse solutions evenly distributed in the Pareto front. Moreover, our methods are theoretically guaranteed to converge to the Pareto front. We demonstrate the effectiveness of our method, especially the SVGD algorithm, through extensive experiments, showing its superiority over existing gradient-based algorithms.

1 Introduction

Many scientific and engineering problems involve optimizing multiple conflicting objectives [5, 21, 4], including, for example, designing wireless sensors [9], building electric power system [22], and training neural networks with multiple tasks [25]. With multiple conflicting objectives, it is impossible to find a single solution that optimizes all the objectives simultaneously. Instead, it is essential to find a set of diverse solutions in the Pareto front that represent different preferences on the different objective functions, so that the users can have a global view of how the optimal trade-off of the different objectives look like and select the solution according to their own preference.

Unfortunately, profiling the Pareto fronts casts a key computational challenge, especially for high dimensional problems [5, 9, 21]. Traditionally, a large literature has been devoted to developing black-box, derivative-free algorithms that are suitable for black-box optimization, such as these based on evolutionary algorithms [6] and Bayesian optimization [18, 1, 28]. However, the black-box algorithms tend to be expensive and can only be applied to small scale problems due to the lack of gradient information. Moreover, black-box algorithms are usually designed based on heuristic arguments, so there is not much theoretical understanding on the optimality conditions or their convergence speed.

Gradient-based MOO algorithms has been catching attention only recently, which include mainly multiple gradient descent (MGD) based methods [7, 16, 20]. However, the results of these methods are still dissatisfying in finding diverse and evenly distributed Pareto solutions in complex problems.

We introduce Stein variational gradient descent (SVGD) and Langevin dynamics as efficient approaches for profiling Pareto fronts. These algorithms are well-understood sampling tools. They iteratively evolve a group of interacting particles to represent the target distribution. When coupled

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

with MGD, these particles will be attracted to the Pareto front quickly. The main difference of these two algorithms is how they distribute points. The Langevin Dynamics uses stochastic noises to perturb the particle trajectories, so they can visit different areas of the Pareto front. On contrast, SVGD is a deterministic sampling algorithm that pushes the particles to high probability regions using MGD, while enforcing diversity between the particles using a repulsive force.

Theoretical analyses are provided for both algorithms to understand their limiting distributions and their convergence speed. One challenge with MGD based sampling is that the limiting distributions do not admit explicit formulations. This is mainly because the forcing from MGD is in general not the gradient of any function. However, we can show its non-gradient component is orthogonal to a large class of functions. Assuming each objective function is strongly convex and regular, we can also show the limiting distributions concentrate on the Pareto front. Moreover, we can show the two algorithms converge to good solutions of MOO with $O(1/t)$ and linear rate.

We test our methods on a variety of tasks, ranging from low-dimensional optimization to multi-task neural network optimization. On all the tasks tested, our method can obtain diverse and high quality Pareto solutions that distribute evenly on the Pareto front, without predefined preference vectors. Quantitatively, we substantially outperform PF-SMG and EPO with respect to the hypervolume metric.

2 Background

In Multi-objective optimization (MOO), we are interested in minimizing a vector-valued loss function $F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \in \mathbb{R}^m$, where $x \in \mathbb{R}^d$ and $f_i : \mathbb{R}^d \to \mathbb{R}, \quad i \in [m]$ the i-th scalar-valued objective function. Here we use notation $[m] = \{1, 2, \ldots, m\}$.

Obviously, we can not fully optimize all the objective functions simultaneously because they may be conflicting with each other. Instead, we are interested in finding the points which can not be improved simultaneously for the objective functions, yielding the notion of **Pareto optimality**.

Definition 1 (Pareto Optimality)**.** *For $x_1, x_2 \in \mathbb{R}^d$, We say that x_1 is* dominated *by x_2 iff $f_i(x_2) \leq f_i(x_1), \forall i \in [m]$, and at least one of them is strict inequality, that is, we have $f_i(x_2) < f_i(x_1)$ for at least one $i \in [m]$. A point x^* is called* globally Pareto optimal *on $\mathbb{R}^d$ iff it is not dominated by any other $x' \in \mathbb{R}^d$. A point x^* is called* locally Pareto optimal *iff there exists an open neighborhood $\mathcal{N}(x^*)$ of x^*, such that x^* is not dominated by $\forall x \in \mathcal{N}(x^*)$. The collection of globally (resp. locally) Pareto optimal points are called the global (resp. local) Pareto front (or set).*

If f_i are non-convex functions, it is only practically feasible to find locally optimal points. Therefore, we mainly focus on finding locally optimal solutions in this work and we refer the local Pareto front as Pareto front.

Our goal is to find diverse and representative solutions $\{x_i\}_{i=1}^n$ from the Pareto front, so that their function values $\{F(x_i)\}_{i=1}^n$ covers different preferences on different objectives. This would allow the end-users to have a global view on the optimal trade-off between the different objective functions and decide which solution based on their own preference.

Multiple Gradient Descent (MGD) is a simple gradient-based algorithm for finding a *single* Pareto point [7]. The key idea of this work is to extend MGD to find multiple diverse points by integrating it with sampling methods such as SVGD and Langevin dynamics. We introduce the background on MGD in this section.

MGD can be a natural extension of gradient descent to multiple objectives. The idea is to iteratively update the variable along the direction that ensures *all the objectives are decreased simultaneously* (which is called Pareto improvement). Let $g_i(x) = \nabla f_i(x)$ be the graident of the i-th objective. Assume we update the variable by $x' \leftarrow x - \epsilon g^*(x)$, where $g^*(x)$ is a vector field to be determined and ϵ is a small step size. By Taylor approximation, we have $\langle g_i,\ g^* \rangle \approx -(f_i(x') - f_i(x))/\epsilon$, which represents the decreasing rate of f_i when we update x along direction $g^*(x)$. In multiple gradient descent, g^* is chosen to maximize the slowest decreasing rate among all the objectives, that is,

$$g^*(x) = \arg\max_{g \in \mathbb{R}^d} \left\{ \min_{i \in [m]} \langle g, g_i(x) \rangle, \quad s.t. \quad \|g\| \leq 1 \right\}. \tag{1}$$

Therefore, $g^*(x)$ is taken to have positive inner products with all $g_i(x)$. If this is impossible to achieve, $\{g_i(x)\}_{i=1}^m$ will contain conflicting directions, and we would have $g^*(x) = 0$ which terminates the algorithm. Using Lagrangian duality, we can show that the optimal solution to (1) is $g^*(x) \propto \sum_{i=1}^m \lambda_i^*(x) g_i(x)$, where $\{\lambda_i^*(x)\}_{i=1}^m$ is the solution of

$$\min_{\{\lambda_i\}} \left\| \sum_{i=1}^m \lambda_i g_i(x) \right\| \ s.t. \ \sum_{i=1}^m \lambda_i = 1, \ \lambda_i \geq 0, \ \forall i. \tag{2}$$

This optimization has a simple closed-form solution when $m = 2$, and a fast algorithm is offered by [25] for $m > 2$. By construction, when the step size ϵ is small, MGD monotonically decreases *all* the objective simultaneously and will terminate when it arrives a local Pareto point; in this case we have $g^*(x) = \sum_{i=1}^m \lambda_i^*(x) g_i(x) = 0$, suggesting that the zero vector 0 is inside the convex hull of $\{g_i(x)\}_{i=1}^m$. We refer the readers to [7, 25, 16] for more discussion.

The degeneration problem is an important drawback of vanilla MGD. In particular, vanilla MGD only provides a single Pareto solution. The convergence point of MGD is determined by the initialization and the updates (e.g., step size) in an implicit and complicate way, and it is difficult to control it explicitly. It can be harder to converge to some Pareto points than the others. In fact, assume the Pareto front is an open set in $\mathbb{R}^d$, then the vanilla MGD, when initialized from outside of Pareto front, will terminate when it arrives boundary of the Pareto front, and hence *can not reach the interior points*. A simple Example 1 is provided in the appendix to demonstrate this.

3 Profiling Pareto Front with Particle Dynamics

We address the degeneration problem of MGD by integrating it with sampling algorithms for approximating distributions to obtain diverse solutions that fulfill the Pareto front more evenly. In this section, we first introduce our main algorithm, which integrates MGD with Stein variational gradient descent (SVGD), yielding an interacting particle system that quickly evolve diverse Pareto solutions with deterministic repulsive forces (Section 3.1). We also introduce a related idea of combining MGD with Langevin dynamics to obtain diverse solutions by injecting diffusion noises, which is found less favourable in practice than SVGD-based method but is of independent interest (Section 3.2).

3.1 Multi-Objective Stein Variational Gradient Descent

Stein Variational Gradient Descent (SVGD) [15] is a deterministic sampling method for finding diverse and representative points (a.k.a. particles) for approximating a un-normalized distribution, by iteratively evolving the particles with both gradient information and a special repulsive force. In this section, we integrate SVGD with MGD and adapt it to profile the Pareto front.

We derive our algorithm following how SVGD was derived in [15]. We initialize a set of points $\{x_i\}_{i=1}^n$ (a.k.a. particles) and iteratively updates them to 1) mimic the MGD trajectory and 2) maintain diversity between the points. For the derivation, assume the particles $\{x_i\}$ follow a distribution ρ and we update them with $x_i \leftarrow x_i - \epsilon\phi^*(x), \ \forall i \in [n]$, where ϕ is an update direction, selected from a predefined function space $\mathcal{H}$, to strike the balance of the following two factors:

1) We want to choose ϕ to make the dynamics mimic the MGD trajectory as close as possible; this can be framed as maximizing the average inner product $\mathbb{E}_{x\sim\rho}[\langle g^*(x), \phi(x)\rangle]$, to make the direction of ϕ and g^* as close as possible.

2) We also want to choose ϕ to encourage the diversity between the particles. Assume ρ is a continuous distribution, the diversity can be measured by the entropy $H(\rho)$, and we want to choose ϕ such that the distribution of the updated particles, denoted by ρ', is maximized. [15] showed that $H(\rho') = H(\rho) - \epsilon\mathbb{E}_{x\sim\rho}[\nabla \cdot \phi(x)] + O(\epsilon^2)$, where $\nabla\cdot$ denotes the diversity operator in vector calculus. Therefore, we should choose ϕ to maximize $\mathbb{E}_{x\sim\rho}[\nabla \cdot \phi(x)]$ to encourage the diversity between the particles.

Overall, we optimize ϕ to maximize a linear combination of the two terms above: $\phi_\rho^* = \arg\max_{\phi\in\mathcal{F}} \left\{\mathbb{E}_{x\sim\rho}[\langle g^*(x), \ \phi(x)\rangle \ - \ \alpha\nabla \cdot \phi(x)]\right\}$, where α is a positive coefficient that controls the importance of the divergence term. Following [15], we take $\mathcal{F}$ to be the unit ball of a reproducing kernel Hilbert space (RKHS) with a positive definite kernel $k(x, x')$. With the same derivation as [15], we get a closed form solution: $\phi_\rho^*(x) \propto \mathbb{E}_{x'\sim\rho}\left[g^*(x')k(x', x) - \alpha\nabla_{x'}k(x', x)\right]$.

By approximating ρ with the empirical distribution of the particles $\rho = \sum_i \delta_{x_i}/n$ and iteratively updating the particles with ϕ^*_ρ above, we obtain the following MOO-SVGD:

$$x_i \leftarrow x_i - \epsilon\hat{\phi}(x_i), \quad \text{where} \quad \hat{\phi}(x_i) = \frac{1}{n}\sum_{j=1}^{n} g^*(x_j)k(x_i, x_j) - \alpha\nabla_{x_j}k(x_i, x_j). \tag{3}$$

Formally, MOO-SVGD simply replaces the gradient term in SVGD with the multiple gradient g^*.

The intuition of the update is clear: the first term in $\hat{\phi}$ makes the dynamic follow the MGD direction so that the particles are pushed towards the Pareto front when $\{g^*(x_i)\}$ have large magnitudes; the second term pushes the different particles away from each other, so that more areas of the Pareto front is approximated. In particular, when all the particles are in the Pareto front with $g^*(x_i) \approx 0$, then only the pairwise repulsive force will push the particles to be away from each other and form a uniform distribution inside the Pareto front.

If g^* is the gradient vector of some scalar function, i.e., $g^*(x) = \nabla f^*(x)$, then our method reduces to the standard SVGD for sampling from distribution $\rho^*(x) \propto \exp(-f(x)/\alpha)$. In general, g^* is not a gradient vector field of any scalar-valued objective function, and hence our method yields a general "non-gradient" variant SVGD. Similar to vanilla SVGD, we can characterize the evolution of the density of the particles with a nonlinear different equation. In the large particle ($n \to \infty$) and continuous time limit, let ρ_t be the limit density of the particles at time t. Following [14], the evolution of ρ_t is governed by

$$\frac{d\rho_t(x)}{dt} = \nabla_x \cdot (\phi^*_\rho(x)\rho_t(x)). \tag{4}$$

Unlike the case of the vanilla SVGD, the behavior of (4) is less clear due to non-gradient nature of $g^*(x)$. In the following, we show that the stationary distribution of (4) is connected to a decomposition of $g^*(x)$ into a gradient component and non-gradient component.

Theorem 3.1. *Assume ρ^* is a fixed point of* (4) *for which $\log \rho^*(x)$ is continously differentiable and satisfies the Stein's identity $\mathbb{E}_{x\sim\rho^*}[\nabla \log \rho^*(x)k(x, x') + \nabla k(x, x')] = 0$ for every $x' \in \mathbb{R}^d$. Then we have the following decomposition of the vector field g^*:*

$$g^*(x) = -\alpha\nabla \log \rho^*(x) + \jmath(x), \tag{5}$$

where $\nabla \log \rho^*$ and $\jmath$ can be viewed as the gradient and non-gradient components of g^*, and $\jmath(x)$ satisfies the following property: its kernel embedding, defined as $\jmath_{[\rho^*]}(x) = \mathbb{E}_{y\sim\rho^*}[\jmath(y)k(x, y)]$, is orthogonal to the kernel embedding of any gradient field, that is, let h be any C^1 function with a compact support and $\nabla h_{[\rho^*]}(x) = \mathbb{E}_{y\sim\rho^*}[\nabla_y h(y)k(x, y)]$, then $\langle \jmath_{[\rho^*]},\ h'_{[\rho^*]}\rangle_{\mathcal{H}} = 0, \quad \forall h$.

Next, we show ρ^* concentrates on the Pareto front. In order to so, recall that $\|g^*(x)\| = 0$ indicates that x is on the Pareto front. Therefore we will show the norm of the MGD field g^* is small over ρ^* and SVGD updates can find a solution with small $\|g^*\|$.

To simplify our discussion, we limit our discussion on the case where a truncated (σ, m_d)-Gaussian kernel is used: $k(x, y) = (\det(2\pi\sigma^2 I))^{-\frac{1}{2}} \exp(-\frac{1}{2\sigma^2}\|x - y\|^2)1_{\|x-y\|\le m_d\sigma}$. Note that its gradient may be singular at the truncation threshold. Formally we can ignore such singularity and simply define formally: $\nabla_y k(x, y) := \frac{x-y}{\sigma^2}k(x, y)1_{\|x-y\|\le m_d\sigma}$. We also assume the objectives functions are regular and strongly convex:

Assumption 1. *Each objective function f_i is C^3 and the following hold for certain constants $0 < c_1 \le c_2 \le \infty$, $0 \le c_3 < \infty$, $c_1 I \preceq \nabla^2 f_i \preceq c_2 I, \quad |\partial^3_{j,j,k} f_i| \le c_3$. Here $\nabla^2 f_i$ is the Hessian matrix of f_i, and $A \preceq B$ indicates that $B - A$ is a positive semi-definite matrix.*

It is well known that, with strongly convex single objective function, gradient based algorithm can converge to the global minimum. We show this also true for MOO-SVGD method.

Theorem 3.2. *Suppose $\|g^*\|^2$ is C^1, g^* is L-Lipschitz and Assumption 1 holds. If we apply MOO-SVGD using the (σ, m_d)-truncated Gaussian kernel k, the equilibrium distribution ρ^* of MOO-SVGD satisfies*

$$\|g^*_{[\rho^*]}\|^2_{\mathcal{H}} \le \frac{1}{2c_1^2}(L + \alpha/\sigma^2)^2 m_d^2\sigma^2\|1_{[\rho^*]}\|^2_{\mathcal{H}}$$

If the MOO-SVGD density ρ_t following (4) *is considered, for any t there is an $s \leq t$ such that*

$$\|g^*_{[\rho_s]}(x)\|^2_{\mathcal{H}} \leq \frac{1}{2c_1^2}(L + \alpha/\sigma^2)^2 m_d^2 \sigma^2 \|1_{[\rho_s]}\|^2_{\mathcal{H}} + \frac{1}{t}\mathbb{E}_{x\sim\rho_0}\|g^*(x)\|^2.$$

Claim 1 of Theorem 3.2 says that, at the equilibrium ρ^*, the kernel embedding of g^* is bounded by the kernel embedding of the constant 1 multiplied by a constant. Moreover this constant can be arbitrarily small if σ and α are chosen to be small numbers. Claim 2 of Theorem 3.2 indicates the SVGD density flow ρ_t can find a solution of similar property in finite time.

3.2 Multi-Objective Langevin Dynamics

Besides SVGD, MGD can also be integrated with Langevin dynamics (LD), which obtain diverse solutions by injecting random noise into the updates. Similar to SVGD, we can achieve this by simply replacing the gradient field in LD with g^*, yielding MOO-LD:

$$x \leftarrow x - \epsilon g^*(x) + \sqrt{2\alpha\epsilon}\xi, \tag{6}$$

where ξ is a standard Gaussian noise and α is a positive coefficient that determines the magnitude of the noise. The intuition is again very clear: if x is far away from the Pareto front with a large multiple gradient $\|g^*(x)\|$, the dynamics will mainly drive x towards the Pareto front; when x is close to Pareto front with a small $\|g^*(x)\|$, the noise term will dominate and allow x random diffuse on Pareto front.

If g^* is the gradient vector of some sclar-valued function f^*, that is, $g^*(x) = \nabla f^*(x)$, then its stationary distribution equals $\rho^*(x) \propto \exp(-f^*(x)/\alpha)$. If g^* is not a gradient field, the Langevin dynamics is known to be irreversible [e.g., 23, 8].

As shown in our empirical results, the SVGD based method is more practically attractive than the Langevin method. We find that SVGD tends to provide more diverse and uniformly distributed points than Langevin dynamics. And for some of the challenge MOO benchmarks, Langevin dynamics can be stuck at small local region and is very slow to mix. In comparison, thanks to the deterministic updates, SVGD can obtain diverse Pareto solutions in much faster speed.

In the following, we consider the asymptotic property of the dynamics in (6). Consider the continuous time limit of the dynamics in (6) ($\epsilon \to +\infty$), which yields a stochastic differential equation $dx_t = -g^*(x_t)dt + \sqrt{2\alpha}dw_t$, where w_t denotes Brownian motion and t denotes time. Assume we initialize x_0 from a distribution with continuous differential density function ρ_0, then the evolution the density of $x(t)$, denoted by ρ_t, is governed is governed by a Fokker-Planck equation:

$$\frac{d}{dt}\rho_t(x) = \nabla \cdot (g^*(x)\rho_t(x) + \alpha\nabla\rho_t(x)), \tag{7}$$

where $\nabla\cdot$ denotes the divergence operator in vector calculus.

The stationary distribution of (7), denoted as ρ^*, yields a decomposition of $g^*(x)$ analogous to that of (5). Moreover, we can show the KL divergence between ρ_t and ρ^* decays:

Theorem 3.3. *Assume ρ^* is a fixed point of* (7) *for which* $\log \rho^*(x)$ *is C^1. Then we have the following decomposition of the vector field g^*: $g^*(x) = -\alpha\nabla \log \rho^*(x) + \jmath(x)$. For any C^1 function h be any C^1 function with a compact support, $\jmath$ is orthogonal with ∇ under ρ^*: $\int \jmath(x)^T \nabla h(x)\rho^*(x) = 0$. Further, if $\rho_t(x)\jmath(x) \to 0$ when $\|x\| \to \infty$, the KL divergence between ρ_t and ρ^* decreases monotonically with*

$$\frac{d}{dt}\mathrm{KL}(\rho_t \,||\, \rho^*) = -\alpha \mathrm{F}(\rho_t \,||\, \rho^*), \tag{8}$$

where $\mathrm{F}(\rho_t \,||\, \rho^*) = \mathbb{E}_{x\sim\rho_t}[\|\nabla \log(\rho_t(x)/\rho^*(x))\|^2]$ *is the Fisher divergence.*

Theorem 3.3 is similar to Theorem 3.1. The differences are: 1) there is no kernel; 2) Interestingly, SVGD does not seem to have a similar monotonic decreasing property similar to (8). In particular, if ρ^* is strongly log-concave, it is well known that (8) indicates that ρ_t converges to ρ^* linearly in KL divergence.

Analogues to Theorem 3.2, we can also show the MOO-LD yields a distribution that concentrates on the Pareto front. In fact, we can further show $\|g^*(x)\|$ is sub-Gaussian under ρ^*.

Theorem 3.4. *Under Assumption 1, let $M = c_3^2\alpha/c_1 + c_2^2$. For any constant $0 < b \leq \frac{c_1}{4\alpha^2 c_2^2}$, the fixed point ρ^* to* (7) *satisfies*

$$\mathbb{E}_{x\sim\rho^*}\|g^*(x)\|^2 \leq 4\alpha^2 Md/c_1, \quad \mathbb{E}_{x\sim\rho^*}\exp(b\|g^*(x)\|^2) \leq 3\exp(16\alpha bMd/c_1).$$

The solution ρ_t to (7) *satisfies the following*

$$\mathbb{E}_{x\sim\rho_t}\|g^*(x)\|^2 \leq \exp(-\frac{1}{2}c_1 t)\mathbb{E}_{x\sim\rho_0}\|g^*(x_0)\|^2 + 4\alpha^2 Md/c_1,$$

$$\mathbb{E}_{x\sim\rho_t}\exp(b\|g^*(x)\|^2) \leq \exp(-\alpha^2 bMdt)\mathbb{E}_{x\sim\rho_0}\exp(b\|g^*(x)\|^2) + 3\exp(16\alpha bMd/c_1).$$

4 Related Works

We discuss related works on MOO and also introduce hypervolume (HV) indicator which we will use as the evaluation metric in our experiments.

Gradient-free MOO Algorithms A large volume of literature focuses on black box and derivative-free MOO, where $f_i(\boldsymbol{x}),\ \ i \in \{1, 2, \ldots, m\}$ are observed through the outputs [e.g., 6, 18, 11, 28, 18, 10]. These methods are typically based on evolutionary algorithm or Bayesian optimization. However, due to the lack of gradient information, they can perform well only in low dimensional problems and can not be applied to large-scale optimization problems in modern machine learning, such as learning neural networks.

Gradient-based MOO Algorithms Another important line of work is gradient-based methods. The foundation of these methods is MGD. Representative methods include Pareto Front Stochastic Multiple Gradient (PF-SMG) [7, 16], Pareto Multi-task Learning (PMTL) [13], and Exact Pareto Optimization (EPO) [20].

PF-SMG leverages the randomness in stochastic multiple gradient descent to profile the whole Pareto front. The method keeps a set of solutions. In every optimization step, the algorithm removes dominated solutions from the set, randomly sample new solutions from the neighborhood of the survived solutions, and run MGD for another multiple steps. PF-SMG is similar to the Langevin method we discuss in Section 3.2 in that both of them leverage random noise, but PF-SMG leverages the noise in stochastic gradient, which will vanish to zero when the step size decreases to zero. Therefore, PF-SMG would reduce to MGD except the selection step. In constrast, the diffusion noise converges to Brownian motion with small step size.

PMTL and EPO are designed to find local Pareto optimal solutions that satisfy specific user preference. They can also be used to trace the Pareto front by varying the user preference vector uniformly. However, sampling the preference vector uniformly will not necessarily result in a uniform solution set on the Pareto front, which is what we observe in the experiments. Compared with PTML and EPO, our method does not require pre-defined user preference and can obtain more diverse and uniformly distributed points as we show in experiments.

Hypervolume The hypervolume (HV) indicator is a standard metric for evaluating the quality of sets of solutions in MOO. Assume $r = [r_1, \ldots, r_m]$ is an reference point that is an upper bound of the objectives, such that $\sup_x f_i(x) \leq r_i,\ \forall i \in [m]$. For a given set of solutions $\mathcal{X} = \{x_i\}_{i=1}^n$, its hypervolume indicator $HV(\mathcal{X})$ is the measure of the region between $F(\mathcal{X})$ and r,

$$HV(\mathcal{X}) = \Lambda\left(\left\{q \in \mathbb{R}^d \mid \exists x \in \mathcal{X} :\ q \in \prod_{i=1}^m [f_i(x), r_i]\right\}\right), \tag{9}$$

where $\Lambda(\cdot)$ denotes the Lebesgue measure. Directly optimizing the hypervolume metric can be problematic, because it is piece-wise constant. Hypervolume indicator gradient ascent (HIGA) [26] relaxes the hypervolume indicator to get usable gradient for optimization, however, it has two main drawbacks compared with our sampling-based method:(1) its results depend on the choice of reference point(2) it may get stuck in bad local minima. See Appendix for more empirical comparison.

5 Experiments

We show our method, Multi-Objective Stein Variational Gradient Descent (MOO-SVGD) can efficiently profile the Pareto front on a variety of problems, including the ZDT problem set, tri-objective

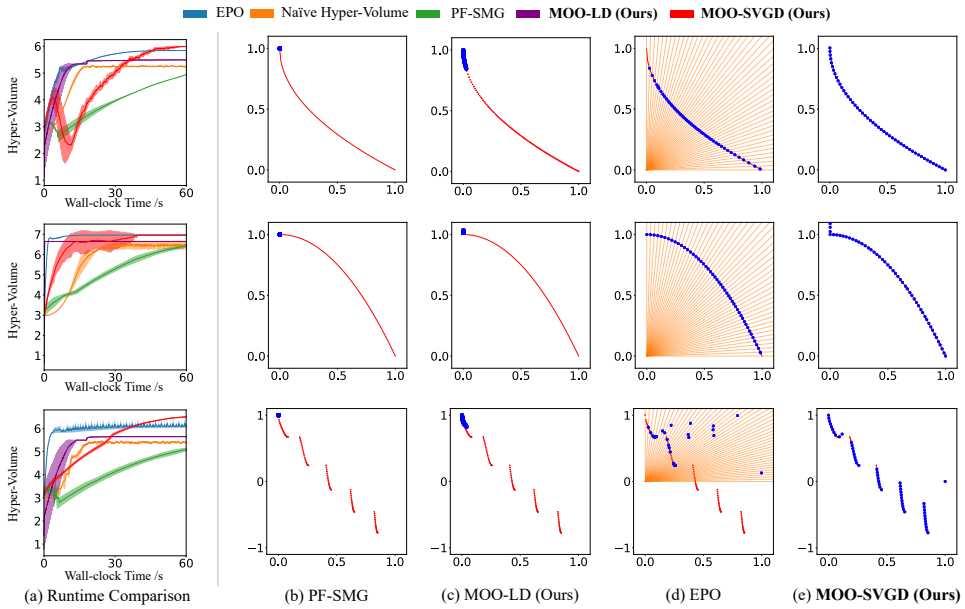

Figure 1: (a) The hypervolume of the results of different methods w.r.t. the wall clock time. (b) The solution set obtained by PF-SMG. All the points converge to the same solution because they do not interact with each other. (c) The solution set obtained by MOO-LD. Thanks to the random noise, it has slightly larger coverage than PF-SMG, but still does not widespread even with 200,000 samples. (d) The solution set obtained by EPO. The orange lines refer to the preference vectors. Note that in ZDT1, uniform preference vector causes non-uniform distribution of the final solution; in ZDT3, EPO cannot deal with negative objectives. (d) Benefited from the explicit repulsive force, MOO-SVGD obtains a uniform Pareto front. We also provide the detailed training dynamics of these methods in the Appendix due to limited space here. All the results are averaged over 5 runs.

problem MaF1, trade-off between accuracy and fairness metrics in fair ML, as well as multi-task learning on MultiMNIST, MultiFashion and MultiFashion+MNIST. Our method outperforms EPO and PF-SMG, two previous baselines for profiling the Pareto front with gradients. We also include Multi-Objective Langevin Dynamics (MOO-LD) for comparison, and find that MOO-SVGD yields much better empirical performance against MOO-LD. We leave the detailed algorithm configuration in the Appendix due to space limitation.

ZDT Bi-objective Problems We compare our methods with other baseline methods on the ZDT problem set [29]. The ZDT problems have a 30-dimensional variable and two objective functions. The details of ZDT problems can be found in the Appendix. In ZDT problems, both objective functions are differentiable. We choose ZDT-1, ZDT-2, and ZDT-3 to demonstrate the performance of our algorithm when the Pareto front is convex, concave, and discontinuous. For baselines, we compare with naive optimization on hypervolume and state-of-the-art gradient-based MOO algorithms (EPO [20] PF-SMG [16]). All the results of the baselines are reproduced using their official implementation. For fair comparison between black-box and white-box optimization algorithms, we report the increase of the hypervolume indicator (Eq. (9)) over the wall-clock time. For fairness of comparison, we use the same reference point when computing hypervolume indicator for every algorithm. Moreover, to make the comparison of wall-clock time fair, we run all the algorithms with CPU on the same computational platform with 192GB memory and 48 cores. The results are shown in Fig. 1. We constrain the maximal wall-clock time in the figure to be 60 seconds. We use 50 particles for MOO-SVGD. Because EPO needs user preference vectors, we uniformly sample 50 unit vectors from a semi-circle to trace a Pareto front, following the strategy provided by the authors [20].

As the figures shows, our method, MOO-SVGD, obtains the largest hyper-volume in all three problems. In the limit of 60 seconds, EPO generally has the best performance among all the three baselines. Naive optimization of hyper-volume get similar performance as the stochastic method, PF-SMG, though the latter is much slower than the former. In ZDT1 and ZDT3, MOO-SVGD converged to a better solution. In comparison, another sampling based method, MOO-LD, can only cover a limited subset of the whole Pareto front.

MaF1 Tri-objective Problem We test our algorithm on MaF1 [3], a problem with three objectives to showcase its ability when there are more than two objectives. The objective functions are defined on $[0,1]^d$. The specific expression of f_1, f_2 and f_3 can be found in Appendix due to limited space. The Pareto front of the problem can be analytically solved, which is $\{x\colon\ f_1(x)+f_2(x)+f_3(x)=2\}$. The reference point for calculating hypervolume is $(1.0, 1.0, 2.0)$. The results are shown in Fig 5 and Table. 1. All the three methods, EPO, MOO-LD, and MOO-SVGD can find Pareto optimal solutions. However, MOO-LD can only cover a very small subset of the Pareto front. EPO can find diverse Pareto optimal solutions, but they are not distributed uniformly. MOO-SVGD obtains the most visually uniform solution set, and hence reaches the highest performance w.r.t. hypervolume. MOO-LD is not tested in the following parts due to its poor performance on toy experiments.

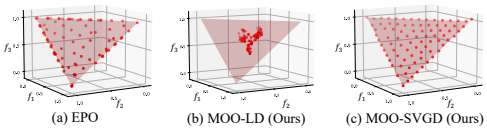

Figure 2: The solution sets obtained by EPO, MOO-LD and MOO-SVGD. The red plane is the Pareto front. The red dots are the final solutions.

[t]

Method	Hypervolume
EPO	0.5714
MOO-LD (Ours)	0.2613
MOO-SVGD (Ours)	**0.5867**

Table 1: The Hyper-volume of the solution sets obtained by EPO, MOO-LD and MOO-SVGD on MaF1.

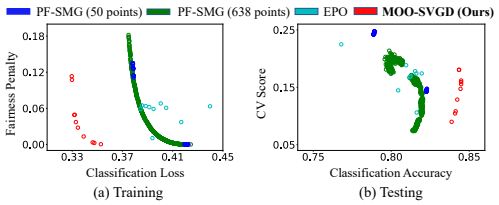

Figure 3: (a) The Pareto front of classification loss and fairness penalty in the training set. MOO-SVGD obtains a uniform solution set that substantially dominates the baselines. (b) Trade-off between classification accuracy and CV score in the test set. PF-SMG (k point) denotes PF-SMG with k solution.

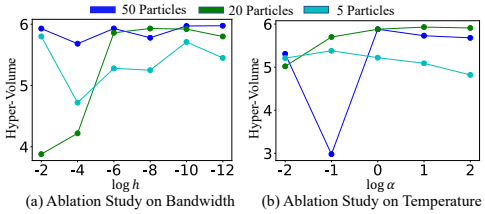

Figure 4: Ablation study of MOO-SVGD on ZDT1. (a) The final hypervolume w.r.t the bandwidth h when $\alpha = 1$. The optimal h is larger for smaller number of particles. (b) The final hypervolume w.r.t. the temperature α when $h = 1e-5$. All the results are averaged over 3 runs.

Trade-off Between Accuracy and Fairness When a ML system is applied to a real-world scenario, it may be biased against people with sensitive attributes, e.g., their genders or races. This leads to unfairness in ML, which is a key issue in real applications of ML. A common practice to avoid unfairness is to add fairness regularization during training. Unfortunately, introducing fairness regularization may hurt the prediction accuracy of the model. The user may have various requirements for the trade-off between accuracy and fairness in different environments. This can be framed into a MOO problem, which simultaneously optimize the typical training loss and the fairness penalties.

For this experiment, please refer to the appendix to see how we construct our dataset. We choose *gender* to be the sensitive attribute, and we use *disparate impact* as the fairness criterion. Disparate impact can be measured by the CV score [2], $\mathrm{CV}(f) = |p(\hat{y}=1|A=0) - p(\hat{y}=0|A=0)|$, where f is our classifier, $\hat{y}$ is the prediction, and A is the sensitive attribute. $\hat{y}=1$ if $f(x)>0$; otherwise, $\hat{y}=0$. CV score implies the gap between the probabilities of getting positive outcomes in different sensitive groups. As in [17], we use a linear classifier with 87 parameters. The classification loss ℓ_1 is the typical binary cross-entropy loss. For fairness penalty, instead of directly optimizing the CV score, we use a convex surrogate of CV score provided in [17], $\ell_2(f) = [\mathbb{E}_{x_i,a_i\in\mathcal{D}}(a_i-\bar{a})f(x_i)]^2$, where x_i is the feature of instance i; a_i is the sensitive attribute of instance i; $\mathcal{D}$ is the dataset, and $\bar{a}$ is the averaged value of all a_i in the dataset. Basically, it is minimizing the empirical covariance between the the prediction and the sensitive attribute.

We compare with PF-SMG [17] and EPO [20]. The number of particles is 10 for MOO-SVGD and EPO. For PF-SMG, we use two different versions. The first version has a maximal particle number of 600; the second is 50. The two versions are denoted by PF-AMG-50 and PF-SMG-600. The results are shown in Fig. 3. Generally, MOO-SVGD yields the best solution set at convergence. Among all the methods, only MOO-SVGD and PF-SMG-600 can get a solution set that is uniformly distributed on different preferences. PF-SMG-50, due to poor exploration ability of the random strategy, shrinks

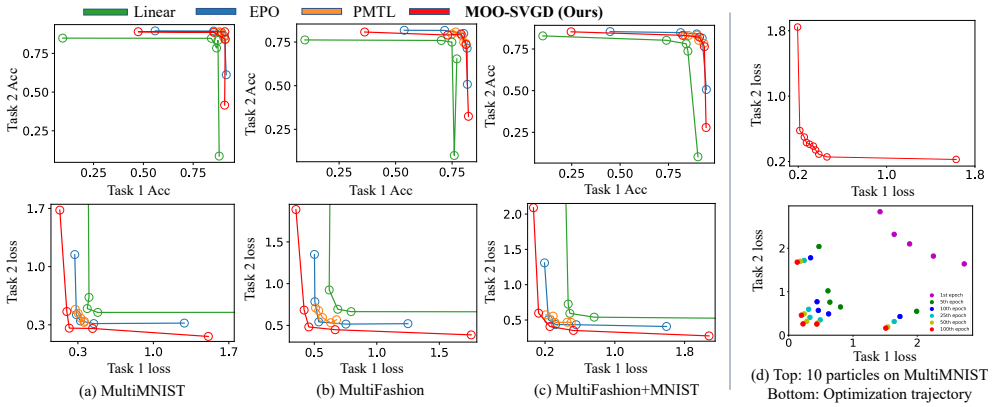

Figure 5: The solution set of the neural networks obtained by different methods on two tasks, w.r.t. the training loss (lower panel) and testing accuracy (upper panel). MOO-SVGD yields the best Pareto front on training loss and a widespread Pareto front on test accuracy.

to the two end points of PF-SMG-600. EPO totally fails in this task. Uniformly sampling preference vector from the semi-circle is problematic, since the Pareto front lies in a very small region of the semi-circle, and most of the preference vectors do not ends up with an 'exact' solution at all.

Multi-task Learning with Neural Networks MultiMNIST, MultiFashion and MultiFashion+MNIST, introduced in [24], are three standard benchmarks for multi-task deep learning. The datasets are created in a similar way: each image in the datasets is a combination of two different images. One of them is on the top-left, and the other is on the bottom-right. In MultiMNIST, both images are from the MNIST dataset [12]. Similarly, MultiFashion uses two images from the FashionMNIST dataset [27]. For MultiFashion+MNIST, the top-left image is from MNIST, while the bottom-right one is from FashionMNIST. All three datasets have 120,000 samples in the training set, and 20,000 samples for test set. These datasets are used in [13, 20, 19].

The baseline methods are linear scalarization, EPO [20] and Pareto Multi Task Learning (PMTL) [13]. EPO and PMTL both require preference vectors to work. We follow the setting in [20], where the authors use LeNet [12] and 5 preference vectors uniformly sampled from a semi-circle. We also use 5 particles for MOO-SVGD.

We find that training with MOO-SVGD can be problematic in the early stage, where networks with different initialization can be too close to each other and cause gradient explosion. Therefore, we initialize the networks with same parameters, then warm-up them by training with naive linear scalarization objective for one epoch, where the weights are sampled uniformly from $[0, 1]$. We switch back to MOO-SVGD in the remaining epochs. For all the experiments, all the methods are trained for 100 epochs with a batch size of 256. Moreover, they are optimized with SGD and the learning rate is 0.001.

6 Conclusion, Limits, Impacts, and Open Questions

In this paper, we propose Multi-objective Stein Variational Gradient Descent (MOO-SVGD). Empirically, it can trace the Pareto front evenly without user preference. There are a number of limits that deserve future efforts. On the practical side, since our methods leverage SVGD and Langevin dynamics, they necessarily inherent their limits, such as the sensitivity on the choice of kernel and step sizes; studies on automatizing the selection of these hyper-parameters can improve the method significantly. Theoretically, analyzing our methods are challenging due to the lack of explicit form of the invariant distributions. For example, although we show that the limiting distributions concentrate on the Pareto front, it is an open question to theoretically characterize in what sense they are evenly distributed in the Pareto front. Moreover, it would be interesting to draw theoretical understandings on why the MOO-SVGD performs better than the MOO-Langevin dynamics.

In terms of social impact, as a new computational tool for MOO, it allows us to better trade-off different objectives in machine learning and hence improve ML systems in both accuracy and various trustworthy metrics such as fairness, safety, as we demonstrate with our experiments. We will release code online to promote the applications of our approach, although it does open the possibility of malicious use of our techniques for adversarial purposes.

References

[1] Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. In *AAAI*, pages 10044–10052, 2020.

[2] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

[3] Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, 2017.

[4] Altannar Chinchuluun and Panos M Pardalos. A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, 154(1):29–50, 2007.

[5] Carlos A Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):269–308, 1999.

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

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

[8] Andrew B Duncan, Tony Lelievre, and GA Pavliotis. Variance reduction using nonreversible langevin samplers. *Journal of statistical physics*, 163(3):457–491, 2016.

[9] Zesong Fei, Bin Li, Shaoshi Yang, Chengwen Xing, Hongbin Chen, and Lajos Hanzo. A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems. *IEEE Communications Surveys & Tutorials*, 19(1):550–586, 2016.

[10] Daniel Golovin et al. Random hypervolume scalarizations for provable multi-objective black box optimization. *arXiv preprint arXiv:2006.04655*, 2020.

[11] Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. *arXiv preprint arXiv:1912.12854*, 2019.

[14] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3115–3123, 2017.

[15] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2370–2378, 2016.

[16] Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *arXiv preprint arXiv:1907.04472*, 2019.

[17] Suyun Liu and Luis Nunes Vicente. Accuracy and fairness trade-offs in machine learning: A stochastic multi-objective approach. *arXiv preprint arXiv:2008.01132*, 2020.

[18] Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems*, 2020.

[19] Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, pages 6522–6531. PMLR, 2020.

[20] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020.

[21] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[22] JE Mendoza, ME Lopez, CA Coello Coello, and EA Lopez. Microgenetic multiobjective reconfiguration algorithm considering power losses and reliability indices for medium voltage distribution network. *IET Generation, Transmission & Distribution*, 3(9):825–840, 2009.

[23] Hong Qian. A decomposition of irreversible diffusion processes without detailed balance. *Journal of Mathematical Physics*, 54(5):053302, 2013.

[24] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[25] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *arXiv preprint arXiv:1810.04650*, 2018.

[26] Hao Wang, André Deutz, Thomas Bäck, and Michael Emmerich. Hypervolume indicator gradient ascent multi-objective optimization. In *International conference on evolutionary multi-criterion optimization*, pages 654–669. Springer, 2017.

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

[28] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2009.

[29] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See section 2.
   (b) Did you describe the limitations of your work? [Yes] ??
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work provides a new way for different societies to collaborate.
   (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] They are listed in Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? See supplementary.
   (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)?
   (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?
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All data are simulated.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] All data are simulated.
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] Our research does not involve human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Our research does not involve human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Our research does not involve human subjects