Pessimistic Off-Policy Multi-Objective Optimization

Shima Alizadeh
AWS AI Labs

Aniruddha Bhargava
Amazon

Karthick Gopalswamy
AWS AI Labs

Lalit Jain
Amazon Visiting Scholar

Branislav Kveton
AWS AI Labs

Ge Liu
AWS AI Labs

Abstract

Multi-objective optimization is a class of optimization problems with multiple conflicting objectives. We study offline optimization of multi-objective policies from data collected by a previously deployed policy. We propose a pessimistic estimator for policy values that can be easily plugged into existing formulas for hypervolume computation and optimized. The estimator is based on inverse propensity scores (IPS), and improves upon a naive IPS estimator in both theory and experiments. Our analysis is general, and applies beyond our IPS estimators and methods for optimizing them.

1 INTRODUCTION

Multi-objective optimization (MOO) is a class of optimization problems with multiple conflicting objectives (Keeney and Raiffa, 1993; Emmerich and Deutz, 2018). Many real-world problems have multiple objectives, such as in economics (Ponsich et al., 2013), engineering (Marler and Arora, 2004), product design and manufacturing (Wang et al., 2011), and logistics (Xifeng et al., 2013). Therefore, MOO has many successful applications. MOO can help a system designer to trade off multiple objectives subject to their preferences. As an example, when designing a product, the form factor, cost, and failure rate need to be carefully balanced.

MOO has been usually studied under the assumption that the objective function is known, with a focus on optimizing it. When it is not known, the problem of learning to optimize it online can be formulated as a *contextual bandit* (Li et al., 2010; Chu et al., 2011), where the goal is to learn a *policy* that takes the most rewarding *action* in each *context*. In many applications, policies cannot be learned online by bandit algorithms because exploration can significantly impact user experience. However, offline data collected by a previously deployed policy are often available. *Offline*, or *off-policy*, optimization using such logged data is a practical way of learning policies without costly online interactions (Dudik et al., 2014; Swaminathan and Joachims, 2015a). In this work, we study offline optimization of multi-objective policies from logged data.

One motivating example for our work is the design of a movie recommendation policy at a movie streaming company. As a first step, the policy would be learned offline to maximize the click-through rate (CTR), for instance. However, after it is deployed online, it may recommend too many recent movies, which was not intended. To avoid the bias, a recent movie penalty is added to the objective and a new policy is learned offline. However, after it is deployed online, it may recommend mostly classic movies, which was again not intended. Therefore, the policy has to be redesigned again. Many iterations like this may be needed until a policy with a good balance between recency, popularity, and relevance is learned. We propose a framework for offline policy optimization that could prevent such costly interactions.

We study off-policy MOO from logged data and make the following contributions. First, we formalize offline optimization of multi-objective policies as hypervolume maximization. Second, we propose a pessimistic IPS estimator for the values of multi-objective policies that can be easily plugged into existing formulas for hypervolume computation. Third, we analyze the error of the estimator when used in optimization, and show its benefits over a naive IPS estimator. Our analysis is general, and applies beyond our IPS esti-

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

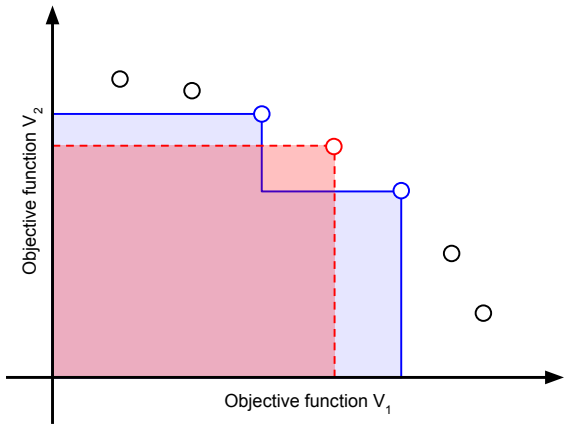

Figure 1: Each point is a value function $V(\pi)$ for one policy $\pi \in \Pi$. The red rectangle is the optimal hypervolume for $K = 1$. The union of the blue rectangles is the optimal hypervolume for $K = 2$.

mators (Section 3) and methods for optimizing them (Section 5). Finally, we show the benefit of pessimistic optimization empirically on all major multi-objective benchmarks: ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2005), and WFG (Huband et al., 2005).

2 SETTING

We formally introduce the problem of policy optimization with a single objective in Section 2.1 and generalize it to multiple objectives in Section 2.2.

2.1 Single-Objective Policy Optimization

We start with introducing our notation. Random variables are capitalized, except for Greek letters like θ. For any positive integer n, we define $[n] = \{1, \ldots, n\}$. The indicator function is $\mathbb{1}\{\cdot\}$. The i-th entry of vector v is denoted by v_i. If the vector is already indexed, such as v_j, we write $v_{j,i}$.

In the classic contextual bandit (Li et al., 2010), the agent observes a *context* $x \in \mathcal{X}$, where $\mathcal{X}$ is a *context set*; takes an *action* $a \in \mathcal{A}$, where $\mathcal{A}$ is an *action set*; and observes a *stochastic reward* $Y \sim P(\cdot \mid x, a)$, where $P(\cdot \mid x, a)$ is the *reward distribution* of action a in context x. We denote the *mean reward* of action a in context x by $r(x, a) = \mathbb{E}_{Y \sim P(\cdot|x,a)}[Y]$. A *policy* π maps actions to contexts, and we denote by $\pi(a \mid x)$ the probability of taking action a in context x.

Let $(x_t)_{t=1}^n$ be a sequence of n contexts. The *expected value of policy* π in contexts $(x_t)_{t=1}^n$ is

$$V(\pi) = \frac{1}{n} \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \pi(a \mid x_t)\, r(x_t, a)\,. \tag{1}$$

The optimal policy maximizes the expected value,

$$\pi_* = \arg\max_{\pi \in \Pi} V(\pi)\,, \tag{2}$$

where Π is a class of optimized policies. If the policy class is sufficiently expressive, π_* could take the action with the highest mean reward in each context.

2.2 Multi-Objective Policy Optimization

Now we extend the setting in Section 2.1 to multiple objectives. The main difference is that the stochastic reward $Y \sim P(\cdot \mid x, a)$ and its mean $r(x, a)$ are vectors of length m, where m is the *number of objectives*. We denote by Y_i and $r_i(x, a)$ the rewards in objective $i \in [m]$. The expected value of policy π, $V(\pi)$ in (1), is a vector of length m and we denote by $V_i(\pi)$ the value in objective i. We call $V(\pi)$ a *value function* because it maps policies to their values in multiple objectives. To simplify exposition, we assume that the stochastic rewards are bounded in $[0, 1]^m$. Thus $r(x, a) \in [0, 1]^m$ and $V(\pi) \in [0, 1]^m$.

Our motivating movie recommendation problem can be formulated in our setting as follows. The context set $\mathcal{X}$ is the set of all users and the action set $\mathcal{A}$ is the set of all movies that can be recommended. The user in interaction t, $x_t \in \mathcal{X}$, is recommended movie $a \in \mathcal{A}$ with probability $\pi(a \mid x_t)$. The mean reward $r(x_t, a)$ could be a 2-dimensional vector, where $r_1(x_t, a)$ is the probability of clicking on a movie and $r_2(x_t, a)$ is the probability of watching it.

The main challenge in extending the optimization in (2) to multiple objectives is that no policy may dominate others in all objectives. To address this problem, we adopt the standard approach in *a-posteriori* MOO (Miettinen, 1998): we cover the Pareto front of V by diverse policies $\pi \in \Pi$. This could be done through random scalarization (Murata and Ishibuchi, 1995), Pareto dominance (Deb et al., 2002), and hypervolume maximization (Emmerich et al., 2005). We adopt the last approach. In our motivating problem, the diverse set of policies would be learned and presented to a human decision maker, which would then select a policy that best matches their preferences.

We measure the diversity of policies by their *hypervolume indicator*, a popular metric in multi-objective optimization (Emmerich et al., 2005). The *hypervolume indicator* of policies $S \subseteq \Pi$ is defined as

$$\begin{aligned} \mathrm{vol}(S, V) &= \int_{y \in [0,1]^m} \mathbb{1}\left\{ \bigvee_{\pi \in S} \{y \le V(\pi)\} \right\} \mathrm{d}y \\ &= \bigcup_{\pi \in S} \bigtimes_{i=1}^{m} [0, V_i(\pi)]\,, \end{aligned} \tag{3}$$

where the inequality $y \leq V(\pi)$ is applied entry-wise. The first definition says that it is the fraction of points $y \in [0,1]^m$ such that $y \leq V(\pi)$ holds for at least one $\pi \in S$. The second definition says that it is the hypervolume of a union of hyperrectangles corresponding to each policy $\pi \in S$. To simplify terminology, we refer to (3) as the *hypervolume*.

Our goal is to identify $\hat{S} \subseteq \Pi$ such that $|\hat{S}| \leq K$ and $\mathrm{vol}(\hat{S}, V) \approx \mathrm{vol}(\Pi, V)$. Roughly speaking, $\hat{S}$ should be as diverse as Π, as measured by covering a similar space. Thus a natural generalization of (2) is the set of K policies that maximizes the hypervolume,

$$S_* = \arg\max_{S \subseteq \Pi : |S| = K} \mathrm{vol}(S, V)\,. \tag{4}$$

We note that (4) reduces to (2) when the number of objectives is $m = 1$. We illustrate solutions to (4) for $K \in \{1, 2\}$ in Figure 1.

In this work, we study a setting where the value function V, an input to $\mathrm{vol}(S, V)$, is estimated from logged data. We present several estimators of V in Section 3 and analyze them in Section 4. In Section 5, we solve (4) using the estimators.

3 OFF-POLICY MULTI-OBJECTIVE ESTIMATION

We estimate the unknown value function V from data collected by a logging policy. The data are logged as follows. Let $(x_t)_{t=1}^n$ be the same sequence of contexts as in (1) and π_0 be a data *logging policy*, which takes action $A_t \sim \pi_0(\cdot \mid x_t)$ in interaction $t \in [n]$. Let $Y_t = (Y_{t,i})_{i \in [m]}$ be the resulting reward, generated as $Y_t \sim P(\cdot \mid x_t, A_t)$. The rewards are *stochastic* and sampled independently, with means $\mathbb{E}\left[Y_{t,i} \mid x_t, A_t\right] = r_i(x_t, A_t)$ and σ^2-sub-Gaussian noise. This process generates a *logged dataset* $\mathcal{D} = \{(x_t, A_t, Y_t)\}_{t \in [n]}$ of size n, which we use to estimate V.

The rest of this section is organized as follows. We present an inverse propensity score estimator of the value function V in Section 3.1. Our main contribution is its pessimistic variant in Section 3.2. We focus on these estimators because they can be easily combined with differentiable policies (Swaminathan and Joachims, 2015a). Other estimators are discussed in Appendix C. We also have a separate estimator for each objective. Such estimators can be easily plugged into existing hypervolume estimators. For instance, if $\hat{V}_i(\pi)$ is an estimate of $V_i(\pi)$, we only need to replace $V_i(\pi)$ in (3) to compute the hypervolume under $\hat{V}_i(\pi)$. The per-objective design is due to Wang et al. (2022), and we are the first to incorporate confidence intervals and pessimism into it.

3.1 IPS Estimator

Inverse propensity scores (IPS) (Horvitz and Thompson, 1952) are arguably the most popular approach to estimating the mean value of a policy in the off-policy setting. In our setting, the IPS estimate for the value of policy π in objective i is

$$\hat{V}_i(\pi) = \frac{1}{n} \sum_{t=1}^{n} \frac{\pi(A_t \mid x_t)}{\pi_0(A_t \mid x_t)} Y_{t,i}\,. \tag{5}$$

While $\hat{V}_i(\pi)$ is an unbiased estimate of $V_i(\pi)$, it tends to have a high variance in practice. This can be mitigated by clipping. In particular, the *clipped IPS* estimate (Ionides, 2008; Strehl et al., 2010) for the value of policy π in objective i is

$$\hat{V}_i(\pi, M) = \frac{1}{n} \sum_{t=1}^{n} \min\left\{ \frac{\pi(A_t \mid x_t)}{\pi_0(A_t \mid x_t)}, M \right\} Y_{t,i}\,,$$

where $M \geq 0$ is a tunable clipping parameter. Lower values of M yield a lower variance in the estimate in exchange for a higher bias. When $M = \infty$, the estimator becomes the IPS. When $M = 0$, the estimator is heavily biased and returns 0 for any policy π.

3.2 Pessimistic IPS Estimator

Another approach to off-policy optimization is based on pessimism (Swaminathan and Joachims, 2015a; Jin et al., 2021; Hong et al., 2023), where a *lower confidence bound (LCB)* is optimized. The LCB can be derived based on a high-probability confidence interval, which we present below.

Lemma 1. *Let $c_i(\pi) = \beta \sigma M_\pi / n$ for $\beta > 0$ and*

$$M_\pi = \sqrt{\sum_{t=1}^{n} M_{t,\pi}^2}\,, \quad M_{t,\pi} = \max_{a \in \mathcal{A}} \frac{\pi(a \mid x_t)}{\pi_0(a \mid x_t)}\,. \tag{6}$$

Then for any objective $i \in [m]$ and policy $\pi \in \Pi$, the bound $|\hat{V}_i(\pi) - V_i(\pi)| \leq c_i(\pi)$ holds with probability at least $1 - 2\exp[-\beta^2/2]$.

Proof. First, note that $\hat{V}_i(\pi)$ is a weighted sum of independent σ^2-sub-Gaussian rewards $Y_{t,i}$ and its mean is $V_i(\pi)$. Second, each reward $Y_{t,i}$ is scaled by at most $M_{t,\pi}$. Therefore, $\hat{V}_i(\pi)$ is sub-Gaussian with variance proxy $\sigma^2 M_\pi^2 / n^2$, and the claim of the lemma follows from standard concentration bounds for sub-Gaussian random variables (Boucheron et al., 2013). □

The main novelty in our work is not in deriving a pessimistic estimator (Jin et al., 2022). It is in applying it to multiple objectives. Our LCB is

$$L_i(\pi) = \hat{V}_i(\pi) - c_i(\pi) \tag{7}$$

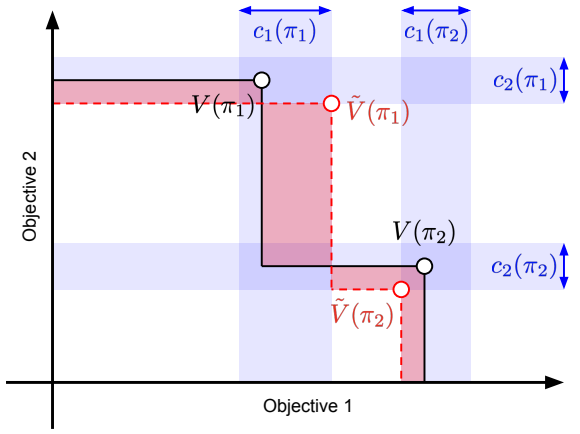

Figure 2: Illustration of Lemma 4 for $S = \{\pi_1, \pi_2\}$. The light blue rectangles represent $\sum_{\pi \in S} \sum_{i=1}^{2} c_i(\pi)$, which bounds the hypervolume difference (red area).

and we call it a *pessimistic IPS estimator*. When $\beta = \sqrt{2 \log(2/\delta)}$, the LCB holds with probability at least $1 - \delta$ for any objective $i \in [m]$ and policy $\pi \in \Pi$. We note that $L_i(\pi)$ can be negative. Therefore, it should be clipped from below by 0 before plugging it into (3) instead of $V_i(\pi)$.

One notable property of $c_i(\pi)$ is that it captures similarities of policies π and π_0. More specifically, $c_i(\pi) = O(M_\pi)$, where M_π^2 is the sum of maximum ratios between probabilities of taking actions under policies π and π_0 in (6). Thus $c_i(\pi)$ decreases as $\pi \to \pi_0$ and so does the uncertainty in the estimate of $V_i(\pi)$.

4 ANALYSIS

Next we analyze the benefit of acting pessimistically. Our analysis assumes access to α-approximate maximization oracles.

Definition 1. *Let $f : \mathcal{S} \to \mathbb{R}$ be a set function, where $\mathcal{S}$ is a set of sets. Then $\tilde{S} \in \mathcal{S}$ is an* α-approximation *for $\alpha \in (0, 1]$ if*

$$f(\tilde{S}) \geq \alpha \max_{S \in \mathcal{S}} f(S) \,.$$

This assumption allows us to study the statistical efficiency of our estimators without being worried about the computational cost of maximizing them. As shown later (Section 4.3), the quality of the oracle affects all our bounds identically. Thus the benefit of pessimism can be argued for any oracle and more abstract treatment is appropriate. Note that when Π is discrete and small, a computationally-efficient maximization oracle exists for $\alpha = 1 - 1/e$ (Section 5.1).

This section is organized as follows. In Section 4.1, we derive error bounds for approximate maximization of functions using their mean and pessimistic estimates. In Section 4.2, we specialize the bounds to approximate hypervolume maximization. Finally, we compare the bounds in Section 4.3. We want to point out that the analyses of mean and pessimistic off-policy estimators are common (Strehl et al., 2010; Swaminathan and Joachims, 2015a; Jin et al., 2021; Yin et al., 2021; Jin et al., 2022; Hong et al., 2023). The main novelty in this work is that we generalize these to MOO. To do this, we decompose the uncertainty of hypervolume into those of its points and show its effect on optimization. Our bounds are general, and apply beyond our IPS estimators in Section 3 and methods for optimizing them in Section 5. All omitted proofs are in Appendix A.

4.1 Approximate Maximization

We find it useful to have a more abstract treatment generalizing to arbitrary set functions. Let Π be a set of points and $\mathcal{S} \subseteq 2^\Pi$ be a subset of its power set. Let $g : \mathcal{S} \to \mathbb{R}$ be a set function with an approximation $\tilde{g} : \mathcal{S} \to \mathbb{R}$. For any set $S \in \mathcal{S}$, let $c(S) \geq |g(S) - \tilde{g}(S)|$ be an upper bound on the approximation error. Let $S_* = \arg\max_{S \in \mathcal{S}} g(S)$ be the optimal solution. Then an approximate maximization of $\tilde{g}$ can be related to it as follows.

Lemma 2. *Let*

$$\tilde{S} = \arg\max_{S \in \mathcal{S}} \tilde{g}(S)$$

and $\hat{S}$ be an α-approximation for $\alpha \in (0, 1]$. Then

$$g(\hat{S}) \geq \alpha g(S_*) - c(S_*) - c(\hat{S}) \,.$$

Optimization of the lower bound $L(S) = \tilde{g}(S) - c(S)$ can be analyzed similarly.

Lemma 3. *Let*

$$\tilde{S} = \arg\max_{S \in \mathcal{S}} L(S)$$

and $\hat{S}$ be an α-approximation for $\alpha \in (0, 1]$. Then

$$g(\hat{S}) \geq \alpha g(S_*) - 2c(S_*) \,.$$

Optimization of the estimated mean (Lemma 2) and the lower bound (Lemma 3) differ as follows. In the former, the error can be arbitrarily large if $\tilde{g}$ significantly overestimates g on some set S. In the latter, the error is bounded by the error at the optimal solution S_* only. If this one is high, S_* is arguably hard to identify. Therefore, maximization of a lower bound yields more robust solutions.

4.2 Hypervolume Maximization

Now we use the error bounds in Section 4.1 to analyze IPS hypervolume maximization (Section 3.1). The key step in our argument is to relate the hypervolume under an approximation $\tilde{V}$ and the true function V. We relate them through errors in individual objectives.

Lemma 4. *Let $V_i(\pi), \tilde{V}_i(\pi) \in [0,1]$ for all $i \in [m]$ and $\pi \in \Pi$. Suppose that $|V_i(\pi) - \tilde{V}_i(\pi)| \le c_i(\pi)$ holds for all $i \in [m]$ and $\pi \in \Pi$. Then*

$$|\mathrm{vol}(S,V) - \mathrm{vol}(S,\tilde{V})| \le c(S) = \sum_{\pi\in S}\sum_{i=1}^{m} c_i(\pi)\,.$$

The lemma says that the difference in the hypervolumes of S under V and $\tilde{V}$ is bounded by the sum of differences of V_i and $\tilde{V}_i$ at individual policies $\pi \in S$. We visualize this in Figure 2.

To obtain an error bound for IPS hypervolume maximization, we chain Lemmas 2 and 4. We assume that the IPS estimator is clipped to $[0,1]$.

Theorem 1. *Take the IPS estimator in (5). Suppose that $\hat{V}_i(\pi) \in [0,1]$ for all $i \in [m]$ and $\pi \in \Pi$, and that*

$$|V_i(\pi) - \hat{V}_i(\pi)| \le c_i(\pi) \tag{8}$$

holds jointly for all $i \in [m]$ and $\pi \in \Pi$ with probability at least $1-\delta$. Let $\hat{S}$ be an α-approximation to

$$\tilde{S} = \arg\max_{S\subseteq\Pi:|S|=K} \mathrm{vol}(S,\hat{V})$$

for $\alpha \in (0,1]$. Then with probability at least $1-\delta$,

$$\mathrm{vol}(\hat{S},V) \ge \alpha\mathrm{vol}(S_*,V) - c(S_*) - c(\hat{S})\,,$$

where $c(S) = \sum_{\pi\in S}\sum_{i=1}^{m} c_i(\pi)$.

Proof. Since (8) holds, we have by Lemma 4 that

$$|\mathrm{vol}(S,V) - \mathrm{vol}(S,\hat{V})| \le c(S)\,.$$

Now we apply Lemma 2, where $g(S) = \mathrm{vol}(S,V)$ and $\tilde{g}(S) = \mathrm{vol}(S,\hat{V})$. □

Theorem 1 says that $\mathrm{vol}(\hat{S},V)$ is within a multiplicative factor of α of $\mathrm{vol}(S_*,V)$. The additional error depends on the magnitude of confidence intervals $c_i(\pi)$ at $\pi \in S_* \cup \hat{S}$. We discuss this more in Section 4.3.

Now we use the error bounds from Section 4.1 to analyze pessimistic IPS hypervolume maximization (Section 3.2). The proof is analogous to Theorem 1, with the only difference that we apply Lemma 3 instead of Lemma 2. We assume that the pessimistic IPS estimator is clipped to $[0,1]$.

Theorem 2. *Take the pessimistic estimator in (7). Suppose that $\hat{V}_i(\pi), L_i(\pi) \in [0,1]$ holds for all $i \in [m]$ and $\pi \in \Pi$, and that (8) holds jointly for all $i \in [m]$ and $\pi \in \Pi$ with probability at least $1-\delta$. Let $\hat{S}$ be an α-approximation to*

$$\tilde{S} = \arg\max_{S\subseteq\Pi:|S|=K} \mathrm{vol}(S,L)$$

for $\alpha \in (0,1]$. Then with probability at least $1-\delta$,

$$\mathrm{vol}(\hat{S},V) \ge \alpha\mathrm{vol}(S_*,V) - 2c(S_*)\,,$$

where $c(S) = \sum_{\pi\in S}\sum_{i=1}^{m} c_i(\pi)$.

Proof. Since (8) holds, we have by Lemma 4 that

$$|\mathrm{vol}(S,V) - \mathrm{vol}(S,\hat{V})| \le c(S)\,.$$

Now we apply Lemma 3, where $g(S) = \mathrm{vol}(S,V)$ and $\tilde{g}(S) = \mathrm{vol}(S,L)$. □

Theorem 2 says that $\mathrm{vol}(\hat{S},V)$ is within a multiplicative factor of α of $\mathrm{vol}(S_*,V)$. The additional error depends on the magnitude of confidence intervals $c_i(\pi)$ at $\pi \in S_*$. We discuss this more in Section 4.3.

4.3 Discussion

Theorems 1 and 2 are similar in two ways. First, the solution $\hat{S}$ in both is α-approximate up to the uncertainty in the estimate of V. Second, the uncertainty is characterized in the same way, by the hypervolume confidence interval widths. When $M_{t,\pi} = O(1)$ in (6), the widths are

$$c(S) = O(\beta\sigma Km/\sqrt{n})\,. \tag{9}$$

The $O(Km)$ dependence arises because the difference of hypervolumes over K points in m objectives can be bounded by the sum of Km hypervolumes, for every point and objective. Figure 2 illustrates it for $m = 2$ objectives. As expected, the hypervolume confidence interval widths increase with the number of policies K, the number of objectives m, and reward noise σ. They decrease with a larger sample size n.

The last issue in (9) is relating β to the failure probability δ in Theorems 1 and 2. This can be done for specific policy classes. For the class in Section 5.2, we show in Appendix F that

$$\beta = \sqrt{2\left(d\log\left(\frac{L\sqrt{n}}{\sigma M}+1\right) + \log m + \log(2/\delta)\right)}\,,$$

where L denotes the maximum of Lipschitz factors of V_i and $\hat{V}_i$. We note that the number of policy parameters d is the only non-logarithmic quantity in β.

Theorems 1 and 2 differ only in where the confidence intervals are instantiated. In Theorem 2, it is the optimal solution S_*. Therefore, when the logging policy π_0 is near optimal, $c(S_*)$ is small and pessimistic IPS maximization is comparable to maximizing $\mathrm{vol}(\cdot, V)$, even if V is unknown and potentially poorly estimated everywhere else but at $\pi \in S_*$.

Such a guarantee cannot be proved for the IPS maximization in Theorem 1. To demonstrate this, we take a policy $\hat{\pi} \in \Pi$ such that $\hat{V}_i(\hat{\pi}) \approx 1$ and $c_i(\hat{\pi}) \approx 1$ for all $i \in [m]$. Based on $\hat{V}_i(\hat{\pi})$ alone, $\hat{\pi}$ has a high value. When the confidence intervals are considered, $\hat{V}_i(\hat{\pi})$ is clearly unreliably estimated, since $L_i(\hat{\pi}) \approx 0$. This is captured by $c(\hat{S})$ in Theorem 1, which would be $O(1)$ if $\hat{\pi} \in \hat{S}$. This would render the bound meaningless.

Finally, when the logging policy π_0 is uniform, we do not expect any benefit of pessimism because all confidence intervals, including $c(S_*)$ and $c(\hat{S})$, would have similar widths. Theorems 1 and 2 show this.

5 HYPERVOLUME OPTIMIZATION

The hypervolume optimization in (4) is a challenging problem because both maximization over policies and hypervolume computation are. We borrow from prior works to address them. All discussions in this section apply to the value function V in (1), its IPS estimator in (5), and its pessimistic IPS estimator in (7).

5.1 Discrete Optimization

When the policy class Π is finite, the hypervolume in (4) can be optimized greedily in K steps as follows. In step $k \in [K]$, a policy $\pi_k \in \Pi$ that increases the hypervolume the most, after being added to the previously selected policies $\{\pi_\ell\}_{\ell=1}^{k-1}$, is chosen,

$$\pi_k = \operatorname*{arg\,max}_{\pi \in \Pi} \mathrm{vol}(\{\pi_1, \dots, \pi_{k-1}, \pi\}, V) . \tag{10}$$

The greedy solution $\{\pi_\ell\}_{\ell=1}^{K}$ is $(1 - 1/e)$-optimal because $\mathrm{vol}(S, V)$ is both monotone and submodular in S (Ulrich and Thiele, 2012). The shortcoming of this approach is that each greedy step requires $O(|\Pi|)$ hypervolume evaluations. Therefore, it is computationally costly when Π is large and cannot even be applied when Π is continuous.

5.2 Policy Gradient

Rather than being restricted to discrete optimization, we optimize a general policy class using policy gradients (Williams, 1992; Sutton et al., 2000; Baxter and Bartlett, 2001). Let

$$\pi(a \mid x; \theta) = \frac{\exp[\phi(x, a)^\top \theta]}{\sum_{a' \in \mathcal{A}} \exp[\phi(x, a')^\top \theta]} \tag{11}$$

be the probability of taking action $a \in \mathcal{A}$ in context $x \in \mathcal{X}$ parameterized by *policy parameter* $\theta \in \Theta$. Here $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is an arbitrary feature mapping and $\Theta \subseteq \mathbb{R}^d$ is the space of policy parameters.

Algorithm 1 Poligy gradient optimization.

Inputs:
Value function V
Number of optimized policies K
Initialize policy parameters $\theta_0 = (\theta_{0,k})_{k \in [K]}$
$\ell \leftarrow 0$
repeat
  $\theta_{\ell+1} \leftarrow \theta_\ell + \alpha_\ell \nabla \mathrm{vol}(\{\pi(\cdot \mid \cdot; \theta_{\ell,k})\}_{k=1}^{K}, V)$
  $\ell \leftarrow \ell + 1$
until convergence
Output: Policy parameters $\theta_\ell = (\theta_{\ell,k})_{k \in [K]}$

To solve (4), we optimize all policies jointly by policy gradients. Our algorithm is presented in Algorithm 1. Its inputs are the value function or its estimate, and the number of optimized policies K. The policies are represented by dK-dimensional vectors. In particular, $\theta_\ell = (\theta_{\ell,k})_{k \in [K]}$ is the vector of all policy parameters in iteration ℓ. We update θ_ℓ as

$$\theta_{\ell+1} = \theta_\ell + \alpha_\ell \nabla \mathrm{vol}(\{\pi(\cdot \mid \cdot; \theta_{\ell,k})\}_{k=1}^{K}, V) ,$$

where α_ℓ is an adaptable learning rate in iteration ℓ. The gradient is with respect to all policy parameters. It exists as long as each $V_i(\pi(\cdot \mid \cdot; \theta))$ is differentiable in θ. This is true for any policy of form (11) plugged into the value function in (1), its IPS estimator in (5), or its pessimistic IPS estimator in (7).

In our experiments, we implement Algorithm 1 using automatic differentiation with Adam (Kingma and Ba, 2015). This choice is motivated by the popularity of Adam and its good initial performance. We discuss other potential choices in Appendix D.

5.3 Hypervolume Computation

Exact computation of the hypervolume of K points is exponential in K, because it corresponds to computing the union of K hyperrectangle volumes. Such computations are only feasible when K is small (Appendix E.1). Efficient exact algorithms also exist for $m = 2$ objectives (Appendix E.2). The general case can be reduced to Klee's measure problem, which has the best known computational complexity of $\tilde{O}(K^{\frac{m}{3}})$

(Chan, 2008). Despite this, many efficient approximations exist (Appendix E).

6 EXPERIMENTS

We also evaluate the benefit of pessimism empirically. Due to space constraints, we only show representative trends and defer the rest to Appendix B.

6.1 Benchmarks

No standardized benchmarks exist for evaluating off-policy MOO. Therefore, we adapt three popular MOO bechmarks, which have been used in numerous works: ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2005), and WFG (Huband et al., 2005). ZDT is a set of bi-objective problems where the number of features can vary. Both the number of objectives and features can vary in DTLZ and WFG problems.

At a high level, we use multi-objective functions in existing benchmarks to define the mean rewards $r(x, a)$ in (1). The mean rewards can be controlled through actions a and we optimize policies over them. Specifically, let $\mathbb{R}^d$ be the feature space of an existing benchmark and $f : \mathbb{R}^d \to \mathbb{R}^m$ be its multi-objective function. We split the feature space as $\mathbb{R}^d = \mathcal{X} \times \mathcal{A}$, where $\mathcal{X} = \mathbb{R}^{\frac{d}{2}}$ and $\mathcal{A} = \mathbb{R}^{\frac{d}{2}}$ are the context and action sets, respectively. The mean reward of action $a \in \mathcal{A}$ in context $x \in \mathcal{X}$ is $r(x, a) = f(x \oplus a)$, where $u \oplus v$ is the concatenation of vectors u and v. We discretize $\mathcal{A}$ by 20 random points to guarantee that the probability of taking actions can be normalized. The features are

$$\phi(x, a) = x \oplus a \oplus \mathrm{vec}(xa^\top) \oplus (1)\,,$$

where $\mathrm{vec}(M)$ denotes the vectorization of matrix M. We introduce the cross-interaction term $xa^\top$ to allow for context-dependent policies.

6.2 Evaluation Protocol

Compared methods. Our approach is a policy gradient (Section 5.2) with the pessimistic IPS estimator in (7). We call it `pessHVI`. We choose $\beta = 0.2$ in the confidence interval $c_i(\pi)$ (Lemma 1), which performed well in our first experiments. To show the benefit of pessimism, we compare `pessHVI` to a policy gradient with the IPS estimator in (5). We call it `meanHVI`.

We consider four additional baselines. The first baseline selects K random policies of form (11), where θ is randomly chosen from a unit ball. We call it `Random`. This baseline shows what is possible with a minimal computational cost. The next two baselines are state-of-the-art genetic algorithms for multi-objective optimization: NSGA-II (Deb et al., 2002) and SMS-EMOA (Beume et al., 2007). We implement both algorithms with the IPS estimator in (5).

The last baseline is a state-of-the-art approach of *expected hypervolume improvement (EHVI)* (Emmerich et al., 2005; Emmerich, 2005; Ernst et al., 2005; Yang et al., 2019). The main challenge in implementing it was that our setting is not Bayesian. Therefore, there is no posterior to sample from. At the end, we implemented it using bootstrapping (Efron and Tibshirani, 1986), which is known to be equivalent to posterior sampling in several notable cases (Lu and Van Roy, 2017; Vaswani et al., 2018). Specifically, we take the logged dataset and resample it N times with replacement. Let $\tilde{V}_j(\pi)$ be the IPS estimate of value function $V(\pi)$ from the resampled dataset $j \in [N]$. Using it as a posterior sample, we can approximate the expected hypervolume of solution S as $\frac{1}{N}\sum_{j=1}^{N} \mathrm{vol}(S, \tilde{V}_j)$. We optimize EHVI using policy gradients over the policy class in (11) and call this algorithm `EHVI`.

Hypervolume computation. All methods are described in Appendix E. We use the exact formula in Appendix E.2 for $m = 2$ objectives. For $m > 2$, we use the scalarized approximation in Appendix E.3.

Logging policy. Let

$$\mathcal{D}_x = \{a \in \mathcal{A} : r(x, a) \le r(x, a') \text{ for some } a' \in \mathcal{A}\}$$

be actions in context $x \in \mathcal{X}$ whose mean rewards are dominated. Then

$$\pi_0(a \mid x) \propto \frac{\varepsilon}{|\mathcal{A}|} + (1 - \varepsilon)\frac{\mathbb{1}\{a \notin \mathcal{D}_x\}}{\sum_{a \in \mathcal{A}} \mathbb{1}\{a \notin \mathcal{D}_x\}}\,. \qquad (12)$$

The policy π_0 is random with probability ε and takes near-optimal actions otherwise. We set $\varepsilon = 0.1$. This mimics a real-world setting where the deployed policy is already of a high quality.

Evaluation. We evaluate all methods by their *recovered hypervolume*, which is the hypervolume of their solutions over the estimated maximum. Since all experiments are simulations, V is known and hence the true hypervolume can be computed. We approximate the maximum hypervolume as $\mathrm{vol}(\tilde{S}, V)$, where $\tilde{S}$ are 10 000 randomly chosen policies as in `Random`. Since the maximum is only approximated, the recovered hypervolume can be more than 1. In all experiments, we average the recovered hypervolume over 20 runs and also report the standard error of its estimate. In each run, we log up to $n = 30\,000$ interactions with reward noise $\sigma = 1$.

6.3 Results

In Figures 3 and 4, we report the performance of all methods on selected ZDT, DTLZ, and WFG benchmarks with $m = 2$ objectives and $d = 6$ features. In

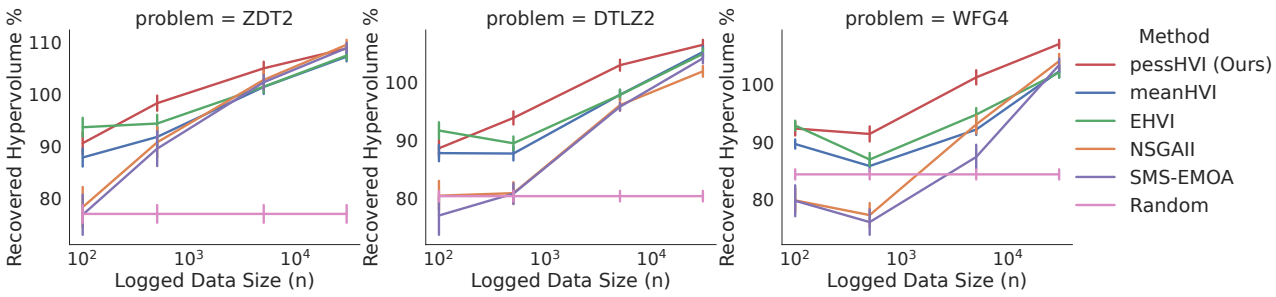

Figure 3: Comparison of `pessHVI` to baselines for $K = 10$ while varying logged dataset size n.

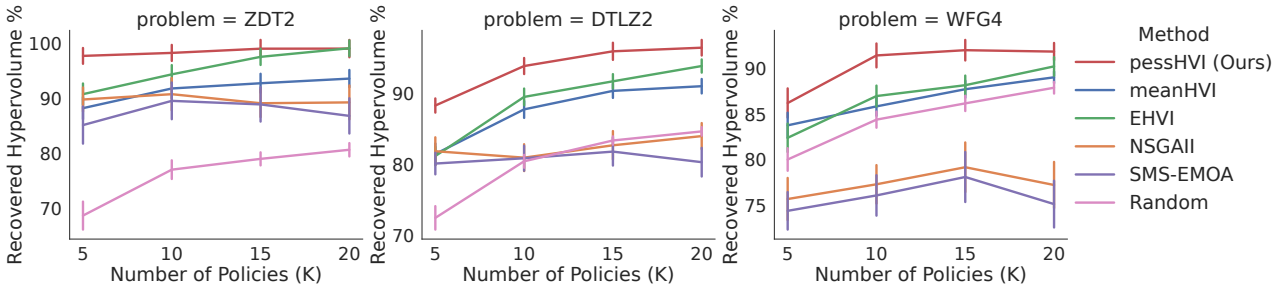

Figure 4: Comparison of `pessHVI` to baselines at $n = 500$ while varying the number of optimized policies K.

Figure 3, we fix the number of optimized policies at $K = 10$ and vary the logged dataset size n. In Figure 4, we fix the logged dataset size at $n = 500$ and vary the number of optimized policies K. We observe two major trends. First, most methods improve as n or K increases. This is because the estimate of V improves with a larger sample size n and hypervolume optimization is easier for larger K. Second, `pessHVI` consistently outperforms `meanHVI` and all other baselines. This clearly shows the value of pessimistic optimization on logged data, which was suggested by our analysis in Section 4.

We present additional results on 5 ZDT, 7 DTLZ, and 9 WFG problems in Appendix B; and observe similar trends to Figures 3 and 4. We also experiment with 7 DTLZ problems with more objectives and features.

6.4 Ablation Study

We conduct an ablation study of recovered hypervolume by `pessHVI` on DTLZ2 problem in Figure 5. In Figure 5a, we vary the logged dataset size n and the number of optimized policies K. We observe that the recovered hypervolume improves in both. This is because the estimate of V improves with larger sample sizes n and hypervolume optimization becomes easier for larger K.

In Figure 5b, we vary ε in the logging policy π_0, while $n = 5\,000$ and $K = 10$. We observe two trends. First, the recovered hypervolume by `pessHVI` gets closer to that of `meanHVI` as π_0 becomes more uniform, $\varepsilon \to 1$. This validates one theoretical insight from Section 4, that the pessimism is less beneficial when the logging policy becomes more uniform. Second, the recovered hypervolumes by both `pessHVI` and `meanHVI` improve as $\varepsilon \to 1$. This is because the maximization of a function becomes easier when the space is uniformly explored, and the maximizer cannot be fooled by poorly estimated parts of the function.

Finally, in Figure 5c, we compare policy-gradient optimization (Section 5.2) to discrete optimization over 1 000 random policies (Section 5.1). The discrete optimization yields worse results, possibly due to poor discretization. This is why we propose continuous optimization by policy gradients in this work.

7 RELATED WORK

Our work lies at the intersection of several fields and we review prior works in detail in Appendix G. Here we discuss only some. Our method is an instance of a-posteriori methods, which cover the Pareto front by a diverse set of solutions. Notable approaches include random scalarization in ParEGO (Knowles, 2006) and evolutionary methods, such as MOEA/D and NSGA-II (Zhang and Li, 2007; Deb et al., 2002). The hypervolume indicator has become the metric of choice in several recent works that provide guarantees (Auer et al., 2016; Zhang and Golovin, 2020).

MOO has been studied extensively in the online setting, where the learning agent interactively explores

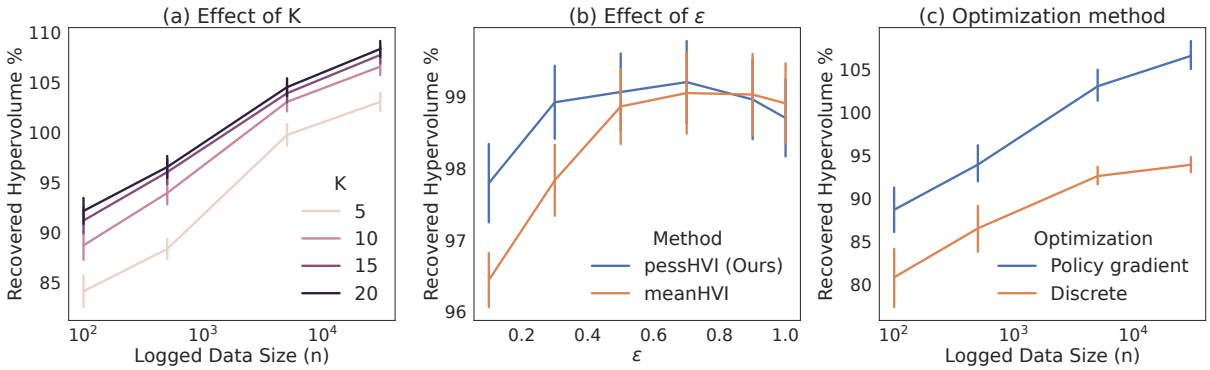

Figure 5: (a) Recovered hypervolume by `pessHVI` as a function of K and logged dataset size n. (b) The benefit of pessimism diminishes as the logging policy becomes more uniform, $\varepsilon \to 1$. (c) Comparison of the recovered hypervolume by policy gradients and discrete optimization in `pessHVI` with $K = 10$.

the Pareto front (Drugan and Nowe, 2013). Both upper confidence bound and posterior sampling methods were proposed (Auer et al., 2016; Yahyaa and Manderick, 2015), even for Gaussian processes (Zuluaga et al., 2013; Paria et al., 2019). Multi-objective reinforcement learning (RL) is a natural generalization of a single-step optimization and an active research area (Hayes et al., 2022).

MOO is understudied in the off-policy setting. Wang et al. (2022) formalized this problem as optimizing a scalarized objective, where the scalarization is learned by interacting with a policy designer. In comparison, our method is a-posteriori, produces a set of diverse policies without any human input, and incorporates pessimism. A common assumption in RL with multiple objectives is that the objectives can be scalarized (Satija et al., 2021; Wu et al., 2021). Therefore, these methods are a-priori. Our method is a-posteriori and does not assume that the context set is finite. Finally, Zhu et al. (2023) used hypervolume to obtain expert trajectories in offline multi-objective RL. This work is empirical and does not use pessimism. In comparison, we show the value of pessimistic optimization in both theory and experiments.

8 CONCLUSIONS

We propose a practical a-posteriori approach to offline optimization of multi-objective policies. The key idea is to maximize a pessimistic hypervolume estimate for a diverse set of policies. The maximization problem is solved by policy gradients. We showcase the benefit of pessimism both theoretically and empirically.

This is one of the first works on offline optimization of multi-objective policies (Section 7). We analyze the benefit of pessimism generally (Section 4), beyond our IPS estimators (Section 3) and methods for optimizing them (Section 5). Therefore, our results should be of a general interest, and lay ground for analyzing other notions of diversity and performance indicators in a-posteriori MOO. One shortcoming of our approach is that each objective is modeled separately. Therefore, we do not leverage correlations among the objectives, which could improve statistical efficiency.

References

C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. *European journal of operational research*, 292(2):397–422, 2021.

P. Auer, C.-K. Chiang, R. Ortner, and M. Drugan. Pareto front identification from stochastic bandit feedback. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016.

J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

J. Branke, J. Branke, K. Deb, K. Miettinen, and R. Slowiński. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media, 2008.

T. M. Chan. A (slightly) faster algorithm for klee's measure problem. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 94–100, 2008.

W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

S. Daulton, M. Balandat, and E. Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.

T. M. Deist, S. C. Maree, T. Alderliesten, and P. A. Bosman. Multi-objective optimization by uncrowded hypervolume gradient ascent. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II 16*, pages 186–200. Springer, 2020.

T. M. Deist, M. Grewal, F. J. Dankers, T. Alderliesten, and P. A. Bosman. Multi-objective learning to predict pareto fronts using hypervolume maximization. *arXiv preprint arXiv:2102.04523*, 2021.

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

M. Drugan and A. Nowe. Designing multi-objective multi-armed bandits algorithms: A study. In *Proceedings of the 2013 International Joint Conference on Neural Networks*, 2013.

M. Dudik, D. Erhan, J. Langford, and L. Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

B. Efron and R. Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1): 54–75, 1986.

M. Emmerich. *Single-and multi-objective evolutionary design optimization assisted by gaussian random field metamodels*. PhD thesis, Dortmund, Univ., Diss., 2005, 2005.

M. Emmerich and A. Deutz. Time complexity and zeros of the hypervolume indicator gradient field. In *EVOLVE-a bridge between probability, set oriented numerics, and evolutionary computation III*, pages 169–193. Springer, 2014.

M. Emmerich and A. Deutz. A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. *Natural Computing*, 17:585–609, 2018.

M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76, 2005.

D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

J. Hong, B. Kveton, M. Zaheer, S. Katariya, and M. Ghavamzadeh. Multi-task off-policy learning from bandit feedback. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47 (260):663–685, 1952.

S. Huband, L. Barone, L. While, and P. Hingston. A scalable multi-objective test problem toolkit. In *Proceedingd of the 3rd International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295, 2005.

E. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline RL? In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Y. Jin, Z. Ren, Z. Yang, and Z. Wang. Policy learning "without" overlap: Pessimism and generalized empirical bernstein's inequality. *CoRR*, abs/2212.09900, 2022. URL https://arxiv.org/abs/2212.09900.

R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

J. 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.

L. Li, W. Chu, J. Langford, and R. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

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

X. Lu and B. Van Roy. Ensemble sampling. In *Advances in Neural Information Processing Systems 30*, pages 3258–3266, 2017.

T. Marler and J. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, 1998.

T. Murata and H. Ishibuchi. MOGA: Multi-objective genetic algorithms. In *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, pages 289–294, 1995.

B. Paria, K. Kandasamy, and B. Poczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

A. Ponsich, A. Jaimes, and C. Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, 2013.

F. P. Preparata and M. I. Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012.

J. Robins, A. Rotnitzky, and L. P. Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.

D. Roijers, L. Zintgraf, and A. Nowe. Interactive Thompson sampling for multi-objective multi-armed bandits. In *Proceedings of the 5th International Conference on Algorithmic Decision Theory*, 2017.

H. Satija, P. S. Thomas, J. Pineau, and R. Laroche. Multi-objective SPIBB: Seldonian offline policy improvement with safety constraints in finite MDPs. In *Advances in Neural Information Processing Systems 34*, pages 2004–2017, 2021.

O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

A. Strehl, J. Langford, L. Li, and S. Kakade. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems 23*, 2010.

R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

A. Swaminathan and T. Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 814–823, 2015a.

A. Swaminathan and T. Joachims. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems 28*, 2015b.

T. Ulrich and L. Thiele. Bounding the effectiveness of hypervolume-based ($\mu+ \lambda$)-archiving algorithms. In *International Conference on Learning and Intelligent Optimization*, pages 235–249. Springer, 2012.

S. Vaswani, B. Kveton, Z. Wen, A. Rao, M. Schmidt, and Y. Abbasi-Yadkori. New insights into bootstrapping for bandits. *CoRR*, abs/1805.09793, 2018. URL http://arxiv.org/abs/1805.09793.

H. Wang, A. Deutz, T. Bäck, and M. Emmerich. Hypervolume indicator gradient ascent multi-objective optimization. In *Evolutionary Multi-Criterion Optimization: 9th International Conference, EMO 2017, Münster, Germany, March 19-22, 2017, Proceedings 9*, pages 654–669. Springer, 2017.

L. Wang, A. Ng, and K. Deb. *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*. Springer, 2011.

N. Wang, H. Wang, M. Karimzadehgan, B. Kveton, and C. Boutilier. IMO^3: Interactive multi-objective off-policy optimization. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 2022.

R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

R. Wu, Y. Zhang, Z. Yang, and Z. Wang. Offline constrained multi-objective reinforcement learning via pessimistic dual value iteration. In *Advances in Neural Information Processing Systems 34*, pages 25439–25451, 2021.

T. Xifeng, Z. Ji, and X. Peng. A multi-objective optimization model for sustainable logistics facility location. *Transportation Research Part D: Transport and Environment*, 22:45–48, 2013.

S. Yahyaa and B. Manderick. Thompson sampling for multi-objective multi-armed bandits problem. In

Proceedings of the 23rd European Symposium on Artificial Neural Networks, 2015.

K. Yang, M. Emmerich, A. Deutz, and T. Bäck. Multi-objective bayesian global optimization using expected hypervolume improvement gradient. *Swarm and evolutionary computation*, 44:945–956, 2019.

M. Yin, Y. Bai, and Y.-X. Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6): 712–731, 2007.

R. Zhang and D. Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

S. Zhou, W. Zhang, J. Jiang, W. Zhong, J. Gu, and W. Zhu. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115, 2022.

B. Zhu, M. Dang, and A. Grover. Scaling pareto-efficient decision making via offline multi-objective RL. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of pareto set approximations. *Multiobjective optimization*, pages 373–404, 2008.

M. Zuluaga, G. Sergent, A. Krause, and M. Püschel. Active learning for multi-objective optimization. In *International Conference on Machine Learning*, pages 462–470. PMLR, 2013.

A Proofs for Section 4 (ANALYSIS)

Lemma 2. *Let*

$$\tilde{S} = \arg\max_{S \in \mathcal{S}} \tilde{g}(S)$$

and $\hat{S}$ *be an* α*-approximation for* $\alpha \in (0, 1]$*. Then*

$$g(\hat{S}) \geq \alpha g(S_*) - c(S_*) - c(\hat{S})\,.$$

Proof. The claim is proved as

$$\begin{aligned}
\alpha g(S_*) - g(\hat{S}) &= \alpha g(S_*) - \alpha \tilde{g}(S_*) + \alpha \tilde{g}(S_*) - g(\hat{S}) \\
&\leq \alpha(g(S_*) - \tilde{g}(S_*)) + \alpha \tilde{g}(\tilde{S}) - g(\hat{S}) \\
&\leq \alpha(g(S_*) - \tilde{g}(S_*)) + \tilde{g}(\hat{S}) - g(\hat{S}) \\
&\leq c(S_*) + c(\hat{S})\,.
\end{aligned}$$

The first inequality holds because $\tilde{S}$ maximizes $\tilde{g}$. The second inequality uses that $\hat{S}$ is an α-approximation. The last inequality follows from the definition of function c and $\alpha \in (0, 1]$. □

Lemma 3. *Let*

$$\tilde{S} = \arg\max_{S \in \mathcal{S}} L(S)$$

and $\hat{S}$ *be an* α*-approximation for* $\alpha \in (0, 1]$*. Then*

$$g(\hat{S}) \geq \alpha g(S_*) - 2c(S_*)\,.$$

Proof. The claim is proved as

$$\begin{aligned}
\alpha g(S_*) - g(\hat{S}) &= \alpha g(S_*) - \alpha L(S_*) + \alpha L(S_*) - g(\hat{S}) \\
&\leq \alpha(g(S_*) - L(S_*)) + \alpha L(\tilde{S}) - g(\hat{S}) \\
&\leq \alpha(g(S_*) - L(S_*)) + L(\hat{S}) - g(\hat{S}) \\
&\leq 2c(S_*)\,.
\end{aligned}$$

The first inequality holds because $\tilde{S}$ maximizes L and the second inequality uses that $\hat{S}$ is an α-approximation. The last inequality follows from $L(S_*) = \tilde{g}(S_*) - c(S_*)$ and $L(\hat{S}) - g(\hat{S}) \leq 0$. After that, we use the definition of function c and $\alpha \in (0, 1]$. □

Lemma 4. *Let* $V_i(\pi), \tilde{V}_i(\pi) \in [0, 1]$ *for all* $i \in [m]$ *and* $\pi \in \Pi$*. Suppose that* $|V_i(\pi) - \tilde{V}_i(\pi)| \leq c_i(\pi)$ *holds for all* $i \in [m]$ *and* $\pi \in \Pi$*. Then*

$$|\mathrm{vol}(S, V) - \mathrm{vol}(S, \tilde{V})| \leq c(S) = \sum_{\pi \in S} \sum_{i=1}^{m} c_i(\pi)\,.$$

Proof. We start with the observation that for any two vectors $a, b \in \{0, 1\}^d$,

$$\left| \prod_{i=1}^{d} a_i - \prod_{i=1}^{d} b_i \right| \leq \sum_{i=1}^{d} |a_i - b_i|\,, \quad \left| 1 - \prod_{i=1}^{d} (1 - a_i) - \left(1 - \prod_{i=1}^{d} (1 - b_i) \right) \right| \leq \sum_{i=1}^{d} |a_i - b_i|\,. \tag{13}$$

In plain English, the difference in the logical "and" and "or" over entries of these vectors is bounded by the sum of the differences of their entries. The definition of the hypervolume together with these inequalities yields

$$\begin{aligned}
|\mathrm{vol}(S, V) - \mathrm{vol}(S, \tilde{V})| &\leq \int_{y \in [0,1]^m} \left| \mathbb{1}\left\{ \bigvee_{\pi \in S} \{y \leq V(\pi)\} \right\} - \mathbb{1}\left\{ \bigvee_{\pi \in S} \{y \leq \tilde{V}(\pi)\} \right\} \right| \mathrm{d}y \\
&\leq \sum_{\pi \in S} \int_{y \in [0,1]^m} \left| \mathbb{1}\{y \leq V(\pi)\} - \mathbb{1}\left\{ y \leq \tilde{V}(\pi) \right\} \right| \mathrm{d}y \\
&\leq \sum_{\pi \in S} \sum_{i=1}^{m} \int_{y \in [0,1]} \left| \mathbb{1}\{y \leq V_i(\pi)\} - \mathbb{1}\left\{ y \leq \tilde{V}_i(\pi) \right\} \right| \mathrm{d}y \\
&= \sum_{\pi \in S} \sum_{i=1}^{m} \left| V_i(\pi) - \tilde{V}_i(\pi) \right| \leq \sum_{\pi \in S} \sum_{i=1}^{m} c_i(\pi) = c(S)\,.
\end{aligned}$$

In the second and third inequalities, we use the "or" and "and" inequalities in (13), respectively. The rest follows from basic integration identities and that we integrate over a $[0, 1]^m$ hypercube. □

B Additional Experiments

We conduct additional experiments on ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2005), and WFG (Huband et al., 2005) problems. The setting is the same as in Section 6.3.

B.1 ZDT Problems

We experiment with 5 ZDT problems out of 6, with $m = 2$ objectives and $d = 6$ features. ZDT5 is excluded since it is a discrete optimization problem. The remaining 5 problems are continuous. Our results are reported in Figure 6. We observe that `pessHVI` consistently improves upon all baselines when $n \geq 500$.

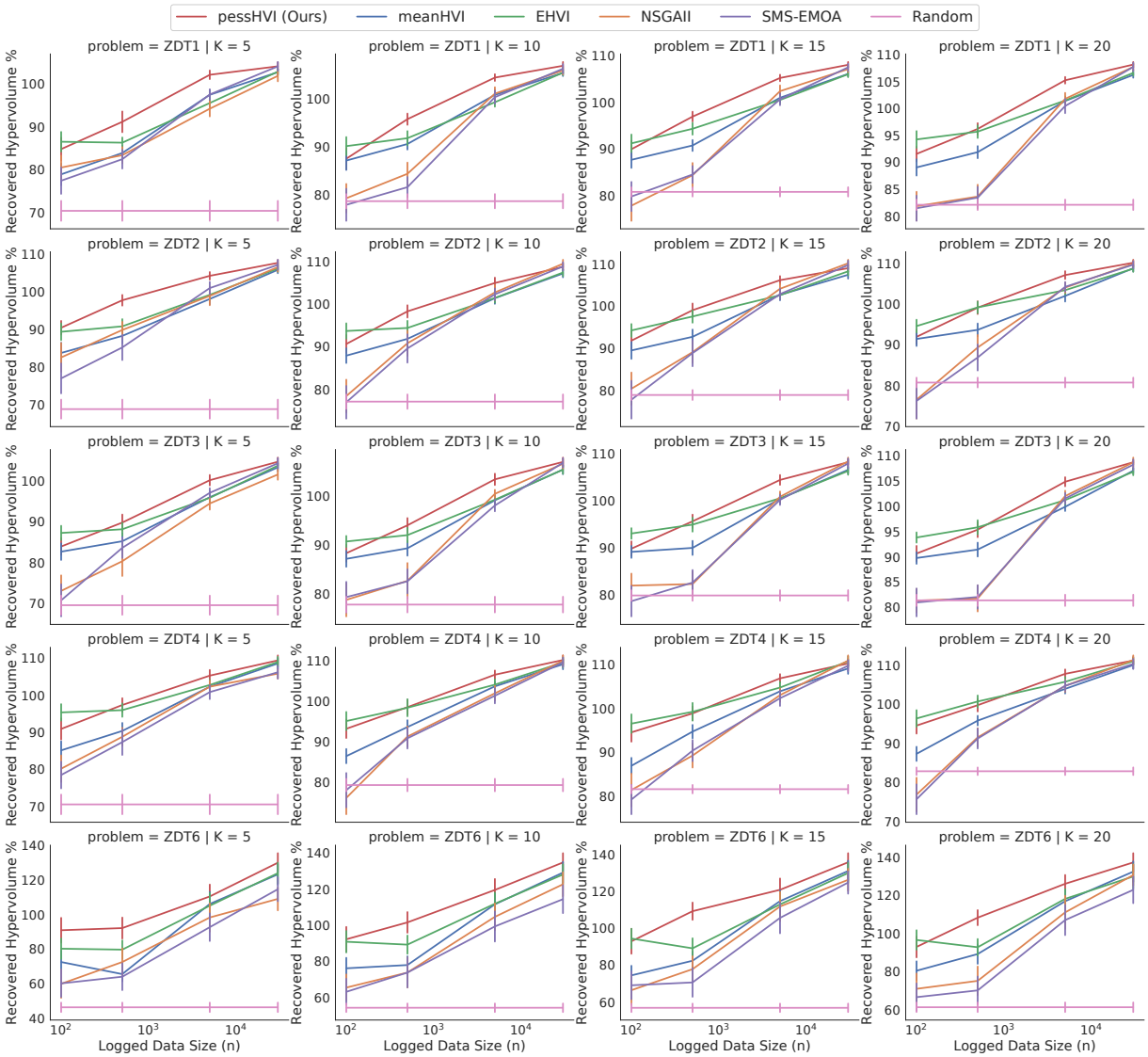

Figure 6: Evaluation of `pessHVI` and all baselines on 5 ZDT problems, for different values of K and n.

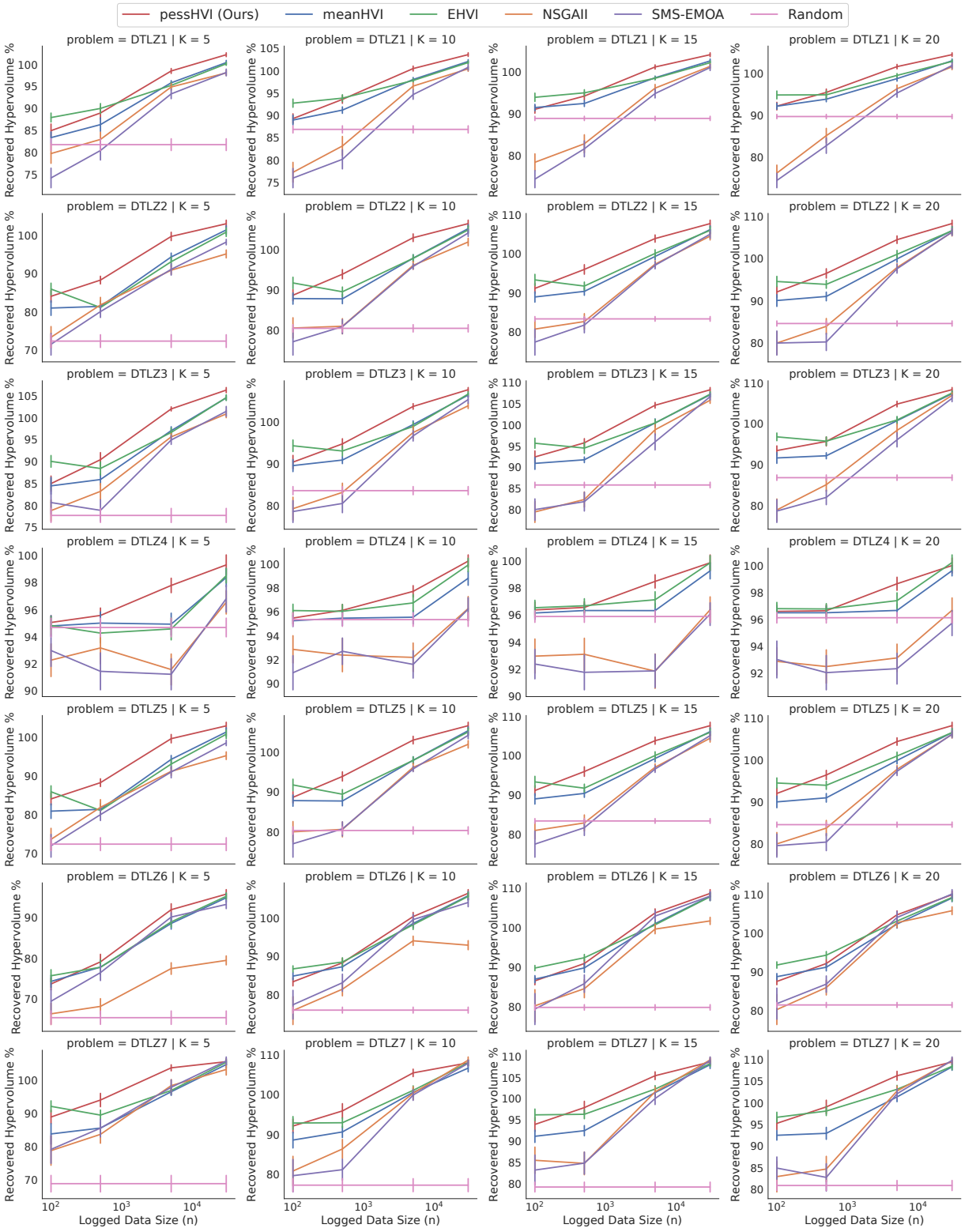

Figure 7: Evaluation of `pessHVI` and all baselines on 7 DTLZ problems, for different values of K and n.

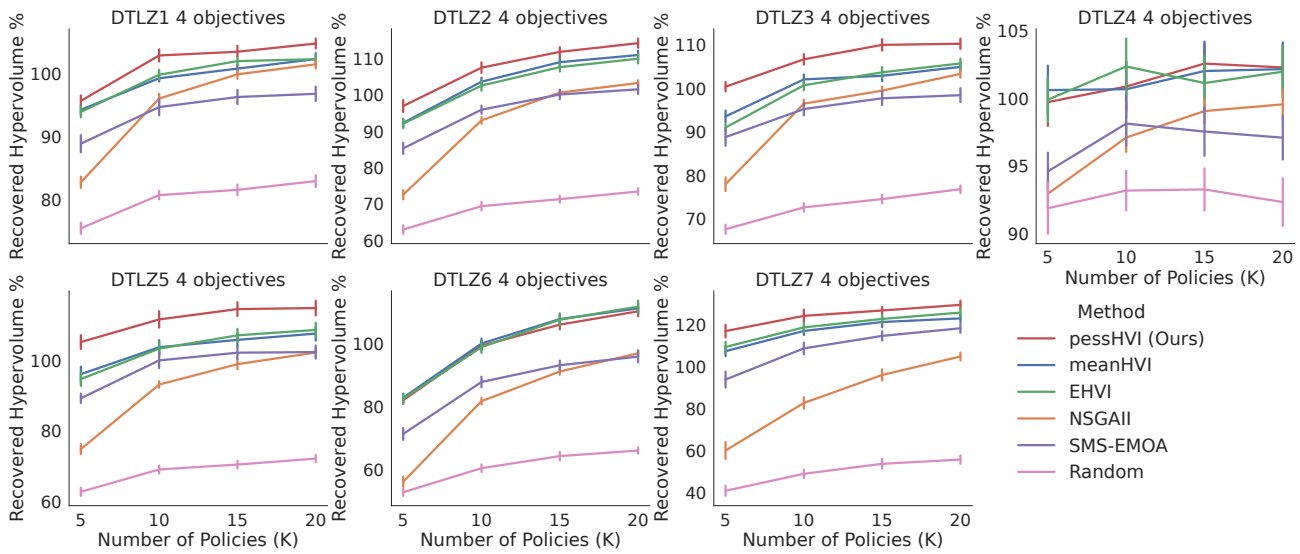

Figure 8: Evaluation of `pessHVI` and all baselines on 7 DTLZ problems with 4 objectives. We set $n = 5\,000$ and vary K.

B.2 DTLZ Problems

We experiment with 7 DTLZ problems out of 9, with $m = 2$ objectives and $d = 6$ features. We exclude DTLZ8 and DTLZ9 because these problems are constrained. The remaining 7 problems are unconstrained. Our results are reported in Figure 7. We observe that `pessHVI` consistently improves upon all baselines when $n \geq 500$. The only exception is DTLZ6, where many methods perform well.

B.3 DTLZ Problems with 4 Objectives

Similarly to Appendix B.2, we experiment with 7 DTLZ problems, with $m = 4$ objectives and $d = 10$ features. The hypervolume is computed as described in Appendix E.3. Our results are reported in Figure 8. We observe that `pessHVI` consistently improves upon all baselines in 5 problems. In DTLZ4 and DTLZ6, `pessHVI` performs comparably to `meanHVI` and `EHVI`.

B.4 WFG Problems

We experiment with 9 WFG problems, with $m = 2$ objectives and $d = 6$ features. Our results are reported in Figure 9. We observe that `pessHVI` consistently improves upon all baselines when $n \geq 500$. The only exception is WFG2, where many methods perform well.

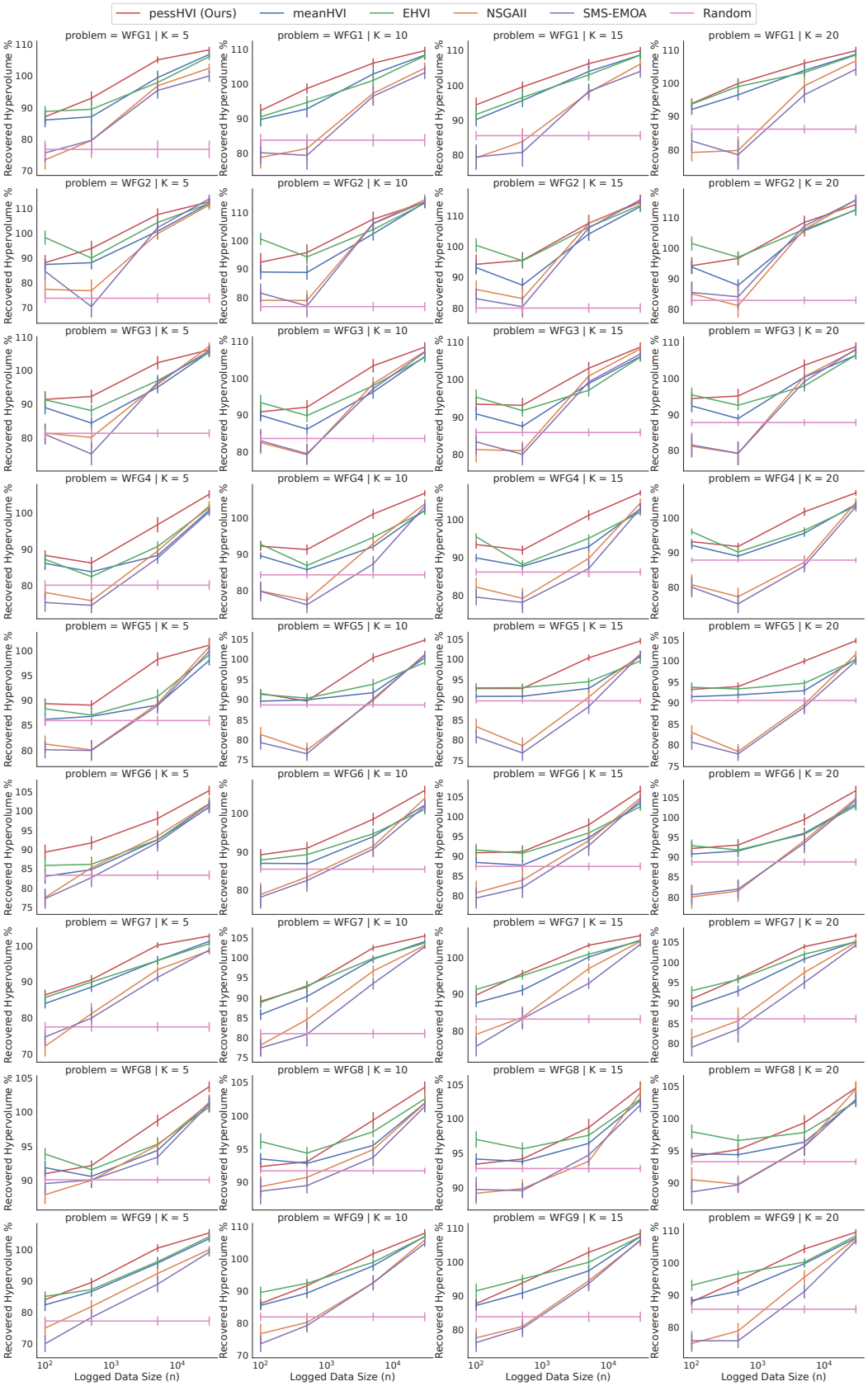

Figure 9: Evaluation of `pessHVI` and all baselines on 9 WFG problems, for different values of K and n.

C Other Estimators

The *direct method (DM)* (Dudik et al., 2014) is a popular approach to off-policy evaluation. Using the DM, the value of policy π in objective i can be computed as

$$\hat{V}_i^{\mathrm{DM}}(\pi) = \frac{1}{n}\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi(a \mid x_t)\,\hat{r}_i(x_t,a)\,,$$

where $\hat{r}_i(x,a)$ is the empirical mean estimate of $r_i(x,a)$.

The *doubly-robust method (DR)* (Robins et al., 1994; Dudik et al., 2014) combines the DM and IPS as

$$\hat{V}_i^{\mathrm{DR}}(\pi) = \frac{1}{n}\sum_{t=1}^{n}\frac{\pi(A_t \mid x_t)}{\pi_0(A_t \mid x_t)}(Y_{t,i} - \hat{r}_i(x_t,A_t)) + \hat{V}^{\mathrm{DM}}(\hat{\pi})\,.$$

It is popular because it combines the advantages of the DM and IPS: it is unbiased when the DM estimator is unbiased or the propensities in the IPS estimator are correctly specified.

Another popular approach is a *self-normalized IPS (SNIPS)* estimator (Swaminathan and Joachims, 2015b),

$$\hat{V}_i^{\mathrm{SNIPS}}(\pi) = \frac{1}{\sum_{t=1}^{n}\frac{\pi(A_t|x_t)}{\pi_0(A_t|x_t)}}\sum_{t=1}^{n}\frac{\pi(A_t \mid x_t)}{\pi_0(A_t \mid x_t)}Y_{t,i}\,.$$

Unlike IPS, the estimator is bounded but biased. However, if the logging policy is supported for all actions in each context, it is consistent.

D Gradient-Based Methods for Diverse Points

Gradient-based methods for finding a diverse set of points have been previously studied in the multi-task and multi-objective optimization literature. They fall into two categories: K points are optimized directly or their hypervolume is. An early approach in the former direction is algorithm MGDA of Désidéri (2012), which uses KKT conditions to compute the direction in which all objectives increase. Various extensions of MGDA have been proposed (Sener and Koltun, 2018; Zhou et al., 2022; Liu et al., 2021). These works do not directly target diversity. Gradient ascent on the hypervolume, as in our work, has been studied. One of the first works on this topic is Wang et al. (2017), who used hypervolume gradient derivations of Emmerich and Deutz (2014). The main challenge for this approach is that the hypervolume indicator is locally constant if any point is dominated. To avoid this, various modifications to steer dominated points to the boundary have been proposed (Deist et al., 2020, 2021). We believe that these methods could improve our policy-gradient optimization in Section 5.2.

E Hypervolume Computation

We review several existing hypervolume estimators. An exact formula based on the inclusion-exclusion principle is presented in Appendix E.1. Unfortunately, it is computationally intractable when S is large. In Appendix E.2, we present an exact formula for two objectives that has $O(|S|\log|S|)$ computation time. Finally, we present an approximation with $O(|S|)$ computation time in Appendix E.3. All formulas are stated for any multi-objective function $f : \Pi \to \mathbb{R}^m$.

E.1 Inclusion-Exclusion Estimator

The key insight in the inclusion-exclusion estimator (Daulton et al., 2020) is that the area of the union of two rectangles is the sum of their areas minus the area of their intersection, which is also a rectangle. In general, for hyperrectangles $\bigtimes_{i=1}^{m}[a, f_i(\pi)]$, where $a \in \mathbb{R}$ is a reference point for the beginning of the coordinate system, the hypervolume can be computed as follows. Let 2^S be the power set of $S \subseteq \Pi$. Then

$$\mathrm{vol}(S,f) = \sum_{C\in 2^S\setminus\emptyset}(2\,(|C| \bmod 2) - 1)\prod_{i=1}^{m}\left(\min_{\pi\in C} f_i(\pi) - a\right)\,. \tag{14}$$

The computation of (14) takes $O(2^{|S|})$ time and therefore is costly even for relatively small S.

E.2 Two Objectives

For $m = 2$ objectives, algorithms with $O(|S| \log |S|)$ computation time exist. More specifically, let $S = \{\pi_k\}_{k=1}^K$ and suppose that $f_1(\pi_1) \leq \cdots \leq f_1(\pi_K)$ holds. The latter can be guaranteed by sorting π_k according to the first objective in $O(|S| \log |S|)$ time (Preparata and Shamos, 2012). Then f can be viewed as a function with a single objective, where $f_1(\pi_k)$ is its input and $f_2(\pi_k)$ is its output, and integrated along the first objective as

$$\mathrm{vol}(S, f) = (f_1(\pi_1) - a)(\max_{k \in [K]} f_2(\pi_k) - a) + \sum_{k=1}^{K-1} (f_1(\pi_{k+1}) - f_1(\pi_k))(\max_{\ell \in [K] \setminus [k]} f_2(\pi_\ell) - a)\,.$$

E.3 Random Hypervolume Scalarization

Scalarization is a mapping $s_\lambda : \mathbb{R}^m \to \mathbb{R}$ for some $\lambda \in \mathbb{R}^m$. The key idea in all scalarization methods is to reduce multiple objectives into a scalar and optimize it. The most common scalarizations are linear and Chebyshev,

$$s_\lambda(f(\pi)) = \sum_{i=1}^m \lambda_i f_i(\pi)\,, \quad s_\lambda(f(\pi)) = \min_{i \in [m]} \lambda_i (f_i(\pi) - a_i)\,,$$

where $a \in \mathbb{R}^m$ is a reference point. Random hypervolume scalarization approximates the hypervolume indicator with random scalarizations sampled from a distribution. Specifically, Zhang and Golovin (2020) showed that the hypervolume $\mathrm{vol}(S, f)$ can be rewritten as

$$\mathrm{vol}(S, f) \propto \mathbb{E}_{\lambda \sim B_m} \left[\max_{\pi \in S} s_\lambda(f(\pi) - a) \right],$$

where $s_\lambda(y) = \min_{i \in [m]} \max\{0, y_i/\lambda_i\}^m$ and B_m is a unit sphere in $\mathbb{R}^m$. The expectation is approximated by an average over multiple sampled λ.

F Joint High-Probability Confidence Interval

Take the policy class in Section 5.2. Let $\Theta = [0,1]^d$ be a policy parameter space and G be a uniform ε-grid over Θ. The grid contains $(1/\varepsilon + 1)^d$ points and the maximum distance of any $\theta \in \Theta$ to the closest point $\theta' \in G$ is $\|\theta - \theta'\|_2 \leq \sqrt{d}\varepsilon$. Let $V_i(\theta) = V_i(\pi(\cdot \mid \cdot; \theta))$ and $\hat{V}_i(\theta) = \hat{V}_i(\pi(\cdot \mid \cdot; \theta))$. Let M be the maximum possible value of $M_{t,\pi}$ in Lemma 1. Let L be the maximum of Lipschitz factors of V_i and $\hat{V}_i$, so that $|V_i(\theta) - V_i(\theta')| \leq L\|\theta - \theta'\|_2$ and $|\hat{V}_i(\theta) - \hat{V}_i(\theta')| \leq L\|\theta - \theta'\|_2$ hold for any $\theta, \theta' \in \Theta$. We note that L is well defined and bounded, because $V_i(\theta) \in [0,1]$ and $\hat{V}_i \in [0, M]$, and π is a continuous function of θ.

By Lemma 1, where $\beta = \sqrt{2 \log(2(1/\varepsilon + 1)^d m/\delta)}$, and the union bound applied to the grid, we get that

$$|\hat{V}_i(\theta) - V_i(\theta)| \leq \sigma M \sqrt{2(d \log(1/\varepsilon + 1) + \log m + \log(2/\delta))/n}$$

holds jointly for all $\theta \in G$ and $i \in [m]$ with probability at least $1 - \delta$. For off-the-grid points, we introduce the closest grid point. Then, jointly over all $\theta \in \Theta$ with probability at least $1 - \delta$,

$$\begin{aligned} |\hat{V}_i(\theta) - V_i(\theta)| &= |\hat{V}_i(\theta) - \hat{V}_i(\theta') + \hat{V}_i(\theta') - V_i(\theta') + V_i(\theta') - V_i(\theta)| \\ &\leq |\hat{V}_i(\theta) - \hat{V}_i(\theta')| + |V_i(\theta') - V_i(\theta)| + \sigma M \sqrt{2(d \log(1/\varepsilon + 1) + \log m + \log(2/\delta))/n}\,, \end{aligned}$$

where $\theta' \in G$ is the closest grid point for θ. We bound the first two terms by $2L\sqrt{d}\varepsilon$, using the Lipschitz factor and that θ' is the closest grid point, and then set $\varepsilon = \sigma M/(L\sqrt{n})$ to make all terms in the bound similar up to logarithmic factors. The final bound, up to logarithmic factors, is $O(\sigma M \sqrt{d/n})$. This completes the proof.

G Additional Related Work

In multi-objective optimization, the decision maker must choose a candidate x from a set of potential candidates $\mathcal{X}$. For each $x \in \mathcal{X}$, there are m objective values $f(x) = (f_i(x))_{i=1}^m$, where $f_i : \mathcal{X} \to \mathbb{R}$. Since the objectives can be traded off in many ways, many algorithms for MOO exist (Emmerich and Deutz, 2018).

In the *a-priori setting* (Branke et al., 2008), the utility of a decision maker is known in advance and used to find the optimal candidate. It is common to represent the utility function as belonging to a family of *scalarizations* of the objectives, where the objectives are weighted separately and then combined. Arguably the most popular approach is linear scalarization $s_\lambda(f(x)) = \sum_{i=1}^{m} \lambda_i f_i(x)$, where $\lambda \in \mathbb{R}^m$ is a weight vector. In many real-world problems, λ is unknown in advance. In such cases, it is natural to present potential candidates to the decision maker that approximate the Pareto front well. This is known as the *a-posteriori setting* (Branke et al., 2008) and many algorithms exist for it. One popular approach is to cover the Pareto front using random scalarization (Zhang and Golovin, 2020). This was done in ParEGO (Knowles, 2006) and an evolutionary algorithm MOEAD (Zhang and Li, 2007). Other evolutionary algorithms, such as NSGA-II (Deb et al., 2002), iteratively refine a population of candidates based on various fitness metrics. Unlike our approach, none of these methods provide guarantees on the quality of the approximation and additionally do not handle uncertainty in objectives.

Regardless of the MOO method, the quality of the resulting solution needs to be measured. Intuitively, a good approximation contains a set of points that are close to the Pareto front and sufficiently diverse. Metrics that capture these two qualities are called performance indicators (Zitzler et al., 2008; Audet et al., 2021). Popular indicators are the Hausdorff distance of the approximation from the Pareto front, R_2, and hypervolume (Zitzler et al., 2000). The last has been increasingly popular and considered in several recent works (Zhang and Golovin, 2020; Auer et al., 2016). As discussed in Appendix E, hypervolume can be challenging to compute.

In the *online setting*, the decision maker interactively explores the Pareto front. Drugan and Nowe (2013) was the first work to apply bandits to MOO. They proposed a `UCB1` algorithm with a scalarized objective and also a Pareto `UCB1` algorithm. Auer et al. (2016) formulated the problem of the Pareto front identification as best-arm identification where each point x is an arm and its rewards are noisy realizations of $f(x)$. Thompson sampling in MOO was studied in Yahyaa and Manderick (2015). Several works assumed that the objective functions are drawn from a Gaussian process (GP). Zuluaga et al. (2013) is an early work with theoretical guarantees and a similar observation model to Auer et al. (2016). Two recent works that applied GP bandits to MOO are Paria et al. (2019) and Zhang and Golovin (2020). Paria et al. (2019) minimized the regret with respect to a known distribution of scalarization vectors. Zhang and Golovin (2020) showed that this algorithm maximizes random hypervolume scalarization. All above works are in the online setting, where the learning agent can interactively probe the environment to learn about objective functions. Our setting is offline.

Arguably the two closest works are Roijers et al. (2017) and Wang et al. (2022). Roijers et al. (2017) treated online MOO as a two-stage problem, where the objective functions are estimated using initial interactions with the environment and the scalarization vector is then estimated by interacting with the designer. This approach was further refined by Wang et al. (2022), who used state-of-the-art off-policy estimation techniques to estimate the objectives and analyzed their approach. The key difference in our work is that we do not put any interaction burden on the policy designer, and simply give them a diverse set of policies to choose from.