Appendix—Hypervolume Maximization: A Geometric View of Pareto Set Learning

Anonymous Author(s)
Affiliation
Address
email

A Experiment Details

A.1 Metrics

As outlined in the main body of the paper, we utilize three metrics to evaluate the effectiveness of the learned solutions. In particular, we assess the performance of a Pareto neural model $x_\beta(\cdot)$ by examining the output of the model for N angles that are uniformly distributed. The output solution set $A = \{y^{(1)}, \ldots, y^{(N)}\}$, where $A = f \circ x_\beta(\widehat{\Theta})$. The three metrics are:

1. The *Hypervolume* indicator [30], which measures both the diversity and convergence of A;
2. The *Range* indicator, which measure the angular span of A;
3. The *Sparsity* indicator [4], which measures the distances between adjacent points.

A.1.1 The Hypervolume Indicator

The hypervolume indicator [30] used to measure A is standard, which has been defined in the main paper,

$$\mathcal{H}_r(A) = \Lambda(\{q \mid \exists p \in A : p \preceq q \text{ and } q \preceq r\}), \tag{11}$$

and r is a reference vector, $r \succeq y^{\text{nadir}}$. For bi-objective problems, the reference point r is set to [3.5, 3.5], whereas for three-objective problems, the reference point is set to [3.5, 3.5, 3.5].

A.1.2 The Range Indicator

The range indicator of a Pareto front is defined in polar coordinates and determines the angular span of the front. Let $(\rho^{(i)}, \theta^{(i)})$ be the polar coordinate of objective vectors $y^{(i)}$ with a reference point r. The relationship between of the Cartesian and polar coordinate is,

$$\begin{cases} y_1 = r_1 - \rho \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{m-1} \\ y_2 = r_2 - \rho \sin\theta_1 \sin\theta_2 \ldots \cos\theta_{m-1} \\ \ldots \\ y_m = r_m - \rho \cos\theta_1. \end{cases} \tag{12}$$

Then, the *Range* indicator is defined as,

$$\text{Range}(A) = \min_{i \in [m]} \max_{\substack{u \in [N], v \in [N], \\ u \neq v}} \left\{ |\theta_i^{(u)} - \theta_i^{(v)}| \right\}. \tag{13}$$

The *Range* indicator can be defined as the minimum angle span across all angles.

A.1.3 The Sparsity Indicator

The sparsity indicator first introduced in [4] measures how dense a set of solutions is. Small inter-solution distances result in a small sparsity indicator indicating a dense Pareto front can be found by the Pareto neural model. We make a modification for $m = 2$ since we find that the maximization operator is much more stable.

$$\text{Sparsity}(A) = \begin{cases} \max\limits_{i \in [N-1]} \sum\limits_{j=1}^{m} \left(\tilde{y}_j^{(i)} - \tilde{y}_j^{(i+1)}\right)^2 \ (m = 2) \\ \dfrac{1}{N-1} \sum\limits_{j=1}^{m} \sum\limits_{i=1}^{N-1} \left(\tilde{y}_j^{(i)} - \tilde{y}_j^{(i+1)}\right)^2 \ (m > 2). \end{cases} \tag{14}$$

where $\tilde{y}_j^{(i)}$ is the i-th solution, and the j-th objective values in the sorted list by the non-dominating sorting algorithm [9]. The unit of the Sparsity indicator is 10^{-3} for bi-objective problems and 10^{-7} for three objective problems.

A.2 Neural Model Architecture and Feasibility Guarantees

We use a 4-layer fully connected neural network similar to [37] for the Pareto neural model $x_\beta(\cdot)$. We optimize the network using Stochastic Gradient Descent (SGD) optimizer with a batch size of 64. The first three layers are,

$$\begin{aligned} x_\beta(\cdot) : \theta &\to \text{Linear}(m, 64) \to \text{ReLU} \\ &\to \text{Linear}(64, 64) \to \text{ReLU} \\ &\to \text{Linear}(64, 64) \to \text{ReLU} \to x_{\text{mid}}. \end{aligned} \tag{15}$$

For constrained problems, to satisfy the constraint that the solution $x_\beta(\lambda)$ must fall within the lower bound (l) and upper bound (u), a sigmoid activation function is used to map the previous layer's output to these boundaries,

$$\begin{aligned} x_{\text{mid}} &\to \text{Linear}(64, n) \to \text{Sigmoid} \\ &\to \odot(u - l) + l \to \text{Output } x_\beta(\lambda). \end{aligned} \tag{16}$$

For unconstrained problems, the output solution is obtained through a linear combination of x_{mid},

$$x_{\text{mid}} \to \text{Linear}(64, n) \to \text{Output } x_\beta(\lambda). \tag{17}$$

A.3 Benchmark Multiobjective Problems

Standard Multiobjective Optimization (MOO) problems. ZDT1-2 [42] and VLMOP1-2 [38] are widely recognized as standard multi-objective optimization (MOO) problems and are commonly employed in gradient-based MOO methods. ZDT1 exhibits a convex Pareto front described by $(y_2 = 1 - \sqrt{y_1},\ 0 \leq y_1 \leq 1)$. On the other hand, ZDT2 presents a non-convex Pareto front defined by $(y_2 = 1 - y_1^2,\ 0 \leq y_1 \leq 1)$, and the LS-based PSL approach can only capture a single Pareto solution.

Real world designing problem. Three real-world design problems with multi-objective optimization are the Four Bar Truss Design (RE21), Hatch Cover Design (RE24), and Rocket Injector Design (RE37). In order to simplify the optimization process, the objectives have been scaled to a range of zero to one.

Multiobjective Linear Quadratic Regulator. The Multiobjective Linear Quadratic Regulator (MO-LQR) problem is first introduced in [44]. MO-LQR is regarded as a specialized form of multi-objective reinforcement learning, where the problem is defined by a set of dynamics presented through the following equations:

$$\begin{cases} s_{t+1} = A s_t + B a_t \\ a_t \sim \mathcal{N}(K_{\text{LQR}} s_t, \Sigma). \end{cases} \tag{18}$$

Table 3: Problem information for multiobjective synthetic benchmarks, design, and LQR problems.

Problem	m	n
ZDT1	2	5
ZDT2	2	5
VLMOP1	2	5
VLMOP2	2	5
LQR2	2	2
Four Bar Truss Design	2	4
Hatch Cover Design	2	2
Rocket Injector Design	3	4
LQR3	3	3

In accordance with the settings discussed in the aforementioned work by Parisi et al. [44], the identity matrices A, B, and Σ are utilized. The initial state for the bi-objective problem is set to $s_0 = [10, 10]$, whereas for the three-objective problem, it is set to $s_0 = [10, 10, 10]$. The reward function is defined as $r_i(s_t, a_t)$, where i represents the respective objective. The function is formulated as follows:

$$r_i(s_t, a_t) = -(1 - \xi)(s_{t,i}^2 + \sum_{i \neq j} a_{t,i}^2) - \xi(a_{t,i}^2 + \sum_{i \neq j} s_{t,i}^2). \tag{19}$$

Here, ξ is the hyperparameter value that has been set to 0.1. The ultimate objective of the MO-LQR problem is to optimize the total reward while simultaneously taking into account the discount factor of $\gamma = 0.9$. The objectives are scaled with 0.01 for better illustration purposes.

Moreover, the control matrix K_{LQR} is assumed to be a diagonal matrix, and the diagonal elements of this matrix are treated as decision variables. Table 3 highlights the number of decision variables and objectives.

A.4 Results on All Problems

The results for all the examined problems are depicted in Figures 10-18, and combined with the results tabulated in Table 5 of the main paper, several conclusions can be made.

Behavior of LS-based PSL. A well-known fact of the linear scalarization method is, it can only learn the convex part of a Pareto front. This fact is validated by Figure 11(e), where LS-based PSL can only learn several solutions.

However, it is crucial to note that the connection between a solution and its corresponding preference vector, $\lambda(\theta)$, is *non-uniform*, though it is rarely discussed in previous literature. Therefore, a uniform sampling of preferences will not result in a uniform sampling of solutions. This observation is supported by the results depicted in Figures 10(e), 13(e), and 15(e), where the learned solutions by LS-based PSL are not uniformly distributed. And as a result, the sparsity indicators are rather high, which indicates the learned front is sparse.

Time Consumption of EPO-based PSL. In comparison to our approach, the Exact Pareto Optimization [6] algorithm, which serves as the foundation for EPO-based PSL [14], exhibits low efficiency due to two factors.

1. To execute the Exact Pareto Optimization (EPO) algorithm, it is necessary to compute the gradients of all objectives, $\nabla f_i(x)$'s. This prerequisite entails performing m back-propagations, resulting in higher computational costs. In contrast, our approach banks on just one back-propagation operation, rendering it a more efficient option in comparison to EPO.
2. For each iteration, the Exact Pareto Optimization (EPO) algorithm entails solving a complicated optimization problem based on the specific value of f_i's, utilizing the respective

Table 4: Licences.

Resource	Link	License
EPO	https://github.com/dbmptr/EPOSearch.git	MIT license
pymoo	https://pymoo.org/	Apache License 2.0
reproblems	https://ryojitanabe.github.io/reproblems/	None

gradients of $\nabla f_i(x)$'s. In contrast, our method does not rely on solving optimization problems for each iteration.

Emphasis on Boundary Solutions. Based on our empirical findings, it is crucial to put emphasis on boundary solutions when aiming to recover a complete Pareto set. As shown in Figure 12 and 14, if all coordinate θ are dealt with equally important, the neural model can only recover a partial part of the Pareto set. PSL-HV1 and PSL-HV2 have different behaviors on the three-objective Rocket Injector Design problem, as shown in Figure 18. PSL-HV2 algorithm has a tendency to accurately identify the complete boundary of the Pareto front, but it often overlooks intermediate solutions. In contrast, although PSL-HV1 method may not always recover the complete boundary, it generates a denser Pareto front.

A.5 Licences

In this paper, we utilized various licenses, which are outlined in Table 4. All methods were implemented using Python and the PyTorch framework, with the SMS-EMOA algorithm being aggregated in pymoo.

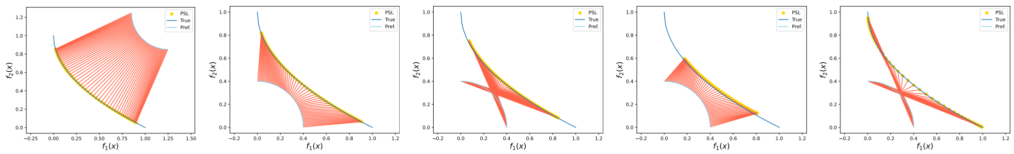
Table 5: Standard derivation (std) value of PSL results on all problems.

	ZDT1				ZDT2				VLMOP1			
Method	HV↑	Range↑	Sparsity↓	Time(s)↓	HV	Range	Sparsity	Time(s)	HV	Range	Sparsity	Time(s)
PSL-EPO	0.05	0.04	0.08	2.03	0.13	0.06	0.25	0.91	0.01	0.01	0.02	0.56
PSL-LS	0.0	0.0	0.2	0.43	0.0	0.0	0.0	0.36	0.0	0.0	0.05	0.76
PSL-Tche	0.01	0.0	0.01	0.56	0.01	0.0	0.22	0.79	0.01	0.01	0.02	0.54
PSL-HV1	0.01	0.0	0.05	0.22	0.03	0.01	0.04	0.2	0.0	0.0	0.03	0.48
PSL-HV2	0.01	0.0	0.04	0.29	0.01	0.0	0.21	0.95	0.01	0.0	0.04	1.15
	VLMOP2				Four Bar Truss Design				Hatch Cover Design			
PSL-EPO	0.08	0.04	0.19	0.48	0.02	0.01	0.01	1.53	0.0	0.02	0.06	4.96
PSL-LS	0.03	0.01	8.69	0.06	0.0	0.0	0.08	0.12	0.0	0.0	0.31	1.21
PSL-Tche	0.01	0.0	0.04	0.49	0.02	0.01	0.02	1.71	0.0	0.01	0.02	2.99
PSL-HV1	0.0	0.0	0.19	1.32	0.01	0.0	0.03	0.38	0.02	0.02	1.41	1.18
PSL-HV2	0.01	0.0	0.13	0.15	0.0	0.0	0.01	1.79	0.0	0.0	0.11	1.42
	LQR2				Rocket Injector Design				LQR3			
PSL-EPO	0.01	0.01	0.03	15.46	1.34	0.08	0.1	1.12	0.01	0.02	0.71	24.21
PSL-LS	0.0	0.0	0.08	3.7	0.0	0.0	0.02	0.11	0.0	0.01	0.05	5.79
PSL-Tche	0.01	0.01	0.1	4.63	0.01	0.0	0.02	1.17	0.01	0.01	0.27	8.9
PSL-HV1	0.0	0.0	0.22	1.83	0.09	0.01	0.18	0.14	0.0	0.02	0.68	1.34
PSL-HV2	0.0	0.0	0.13	9.86	0.03	0.01	1.53	1.31	0.0	0.01	0.72	11.95

B Characters of Hypervolume Maximization

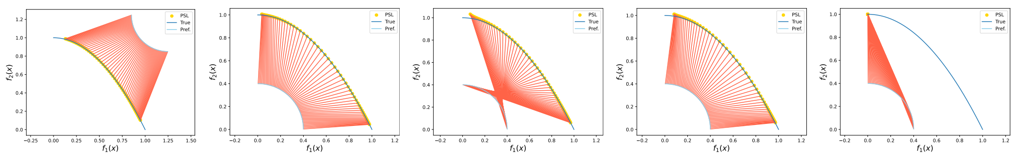
B.1 The Notation Table

To enhance the clarity of the paper, we have included a summary of the main notations in Table 6.



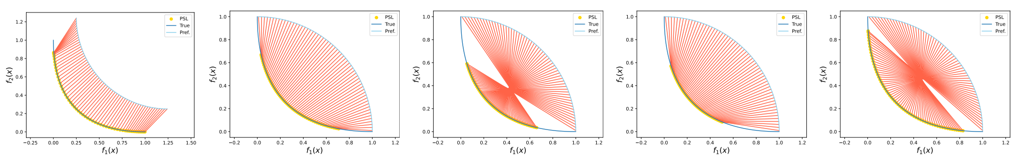
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 10: ZDT1.



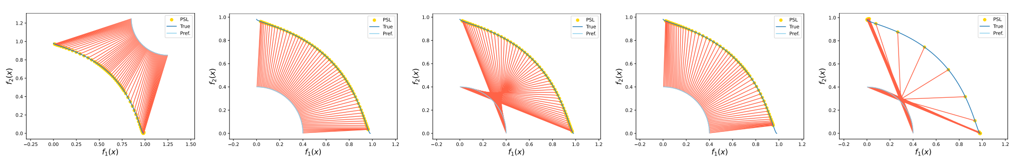
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 11: ZDT2.



(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 12: VLMOP1.



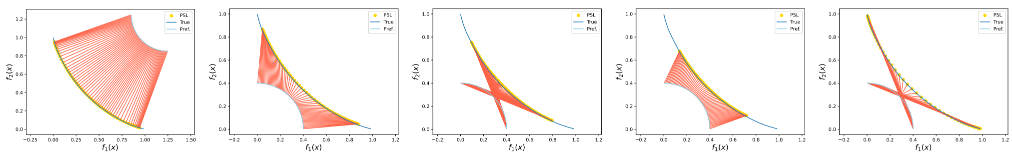
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 13: VLMOP2.

B.2 Hypervolume Calculation in the Polar Coordinate

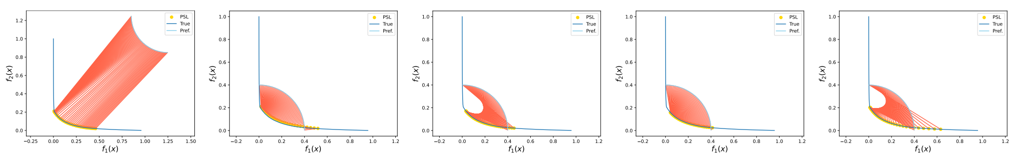
Proof. In this subsection, we provide the proof for Equation (5). $\mathcal{H}_r(\mathcal{F}^*)$ can be simplified by the following equations,

$$
\begin{aligned}
\mathcal{H}_r(\mathcal{F}^*) &= \int_{\mathbb{R}^m} I_\Omega dy_1 \ldots dy_m \\
&= \underbrace{\int_0^{\frac{\pi}{2}} \ldots \int_0^{\frac{\pi}{2}}}_{m-1} dv \\
&= \underbrace{\int_0^{\frac{\pi}{2}} \ldots \int_0^{\frac{\pi}{2}}}_{m-1} \overline{c}_m \cdot \frac{\rho_{\mathcal{X}}(\theta)^m}{2\pi \cdot \pi^{m-2}} \underbrace{d\theta_1 \ldots d\theta_{m-1}}_{d\theta} \\
&= \frac{\overline{c}_m}{2\pi^{m-1}} \underbrace{\int_0^{\frac{\pi}{2}} \ldots \int_0^{\frac{\pi}{2}}}_{m-1} \rho_{\mathcal{X}}(\theta)^m d\theta \\
&= \frac{\overline{c}_m}{2\pi^{m-1}} \cdot (\frac{\pi}{2})^{m-1} \cdot \mathbb{E}_{\theta \sim \text{Unif}(\Theta)}[\rho_{\mathcal{X}}(\theta)^m] \\
&= c_m \mathbb{E}_{\theta \sim \text{Unif}(\Theta)}[\rho_{\mathcal{X}}(\theta)^m].
\end{aligned}
\tag{20}
$$



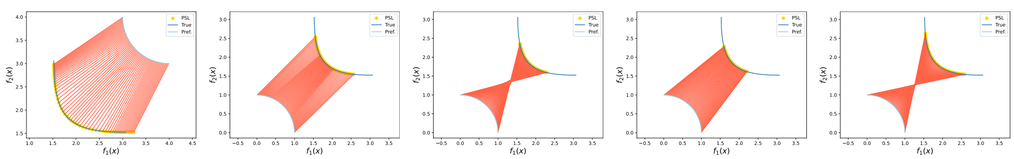
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 14: RE21.



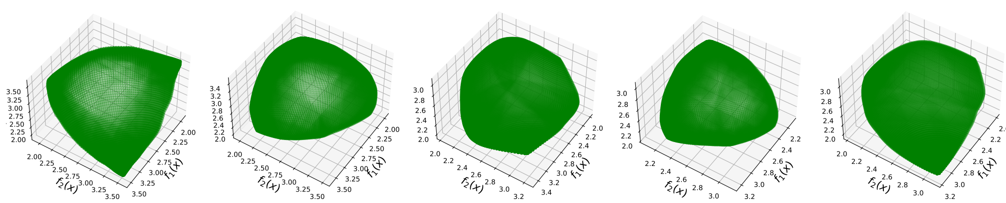
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 15: RE24.



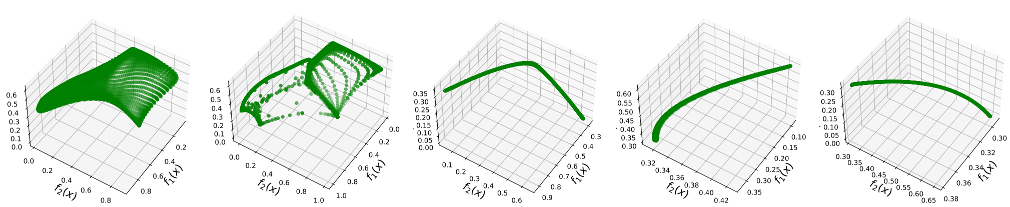
(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 16: MO-LQR2.



(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 17: MO-LQR3.



(a) HV1 (b) HV2 (c) Tche (d) EPO (e) LS

Figure 18: RE37.

Here, Ω denotes the region dominated by $\mathcal{F}^*$ with a reference point r, $\Omega = \{q \mid \exists p \in \mathcal{F}^* : p \preceq q \text{ and } q \preceq r\}$. I_Ω is the indicator function of Ω. $\overline{c}_m$ is the volume of a m-D unit sphere, $\overline{c}_m = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$. c_m is a constant defined in the main paper, $c_m = \frac{\pi^{m/2}}{2^m \Gamma(m/2+1)}$.

Line 2 holds since it represents the integral of Ω expressed in polar coordinates, wherein the element dv corresponds to the volume associated with a segment obtained by varying $d\theta$.

Line 3 calculates the infinitesimal volume of dv by noticing the fact that the ratio of dv to $\overline{c}_m$ is $\frac{\rho_{\mathcal{X}}(\theta)^m}{2\pi \cdot \pi^{m-2}}$. Line 4 is a simplification of Line 3. And Line 5 and 6 express the integral in its expectation form. $\square$

Table 6: The notation table.

Variable	Definition
x	The decision variable.
n	The number of the decision variables.
N	The number of samples.
m	The number of objectives.
θ	The angular polar coordinate.
$\lambda(\theta)$	An m-dimensional preference vector.
β	The model parameter.
$y^{\text{nadir}}/y^{\text{ideal}}$	The nadir/ideal point of a given MOO problem.
$\mathcal{F}^*$	The Pareto front, which is set of all Pareto non-dominated solutions.
$\mathcal{H}_r(A)$	The hypervolume of set A w.r.t a reference .
$\mathcal{S}_+^{m-1}$	The $(m$-1)-D positive unit sphere.

B.3 Proof of $\rho_{\mathcal{X}}(\theta)$ as a Max-Min Problem

We provide the proof of the following equation (Equation (6) in the main paper) in this subsection.

$$\rho_{\mathcal{X}}(\theta) = \max_{x \in \mathcal{X}} \rho(x, \theta) = \max_{x \in \mathcal{X}} \min_{i \in [m]} \{\frac{r_i - f_i(x)}{\lambda_i(\theta)}\}.$$

Proof. Let x^* be one of the optimal solutions of Problem $\max_{x \in \mathcal{X}} \rho(x, \theta)$. To begin, we define the attainment surface $\mathcal{S}_{\text{attain}}$, as detailed in [31], utilizing a reference point r. The sets of Pareto solutions and weakly Pareto solutions are denoted as $\mathcal{F}^*$ and $\mathcal{F}^*_{\text{weak}}$, respectively. Then, $\mathcal{S}_{\text{attain}}$ is defined as,

$$\mathcal{S}_{\text{attain}} = \mathcal{F}^* \cup \{p \mid p \preceq r, \ p \in \mathcal{F}^*_{\text{weak}}\}. \tag{21}$$

We denote $P(\theta)$ as the intersection point of the ray from the pole r along angle θ and the attainment surface $\mathcal{S}_{\text{attain}}$. $\rho_{\mathcal{X}}(\theta)$ is the distance from the reference point r to the intersection point $P(\theta)$. There are two cases, x^* is a Pareto solution or a weakly Pareto solution. Else, by contradiction, $f(x^*)$ can be improved in all objectives, x^* cannot be a solution of Problem (6).

When x^* is Pareto optimal. In such case, we should prove that $f(x^*) = P(\theta)$. If $x^* \neq P(\theta)$, then there exist at least one element j such that, $\frac{r_j - f_j(\theta)}{\lambda_j(\theta)} \leq \frac{r_i - P_i(\theta)}{\lambda_i(\theta)}$, $\forall i = 1, \ldots, m$. This is a contradiction with x^* is the optimal solution of Problem (6). So, $x^* = P(\theta)$.

When x^* is weakly Pareto optimal. In such case, $f(x^*)$ does not necessary equals to $P(\theta)$. In such case, since x^* is the solution of Problem (6), we have that there exist at least one index j, where $j = \arg\min \frac{r_j - f_j(x^*)}{\lambda_j(\theta)}$ such that $\frac{r_j - f_j(x^*)}{\lambda_j(\theta)} = \frac{r_i - P(\theta)}{\lambda_i(\theta)}$, $i = 1, \ldots m$. In such a case, $dist(P(\theta), r) = \frac{r_j - f_j(x^*)}{\lambda_j(\theta)}$. □

B.4 Proof of Proposition 2

This subsection provides the proof for Proposition 2, which builds the relationship between a polar angle θ and the corresponding solution of Problem (6).

Proof. There are two cases for x^*. x^* is Pareto optimal or x^* is weakly Pareto optimal. When x^* is neither Pareto optimal nor weakly Pareto optimal, there exists a solution x' which is better than x^* for all objectives. In such case, x^* is not a solution for Problem (6), which is a contradiction.

When x^* is Pareto optimal. Since we have $\rho_{\mathcal{X}}(\theta) = \frac{r_i - f_i(x^*)}{\lambda_i(\theta)}$, which indicates that for any other solution x', there exist at least one index j such that, $\frac{r_j - f_j(x')}{\lambda_j(\theta)} \leq \rho_{\mathcal{X}}(\theta)$, then x' is not the optimal solution of Problem (6). As a result x^* is the only solution of Problem (6), $\mathcal{X}_\theta = \{x^*\}$.

When x^* is weakly Pareto optimal. There can exist one solution x' such that, $x'_i \neq x^*_i$ for some i and therefore, $x' \in \mathcal{X}_\theta$. As a result, we can conclude that, $x^* \in \mathcal{X}_\theta$. $\square$

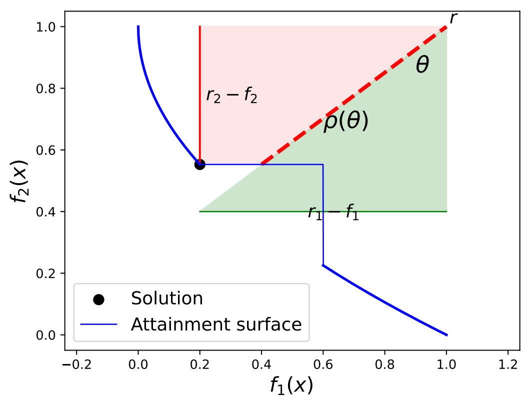


Figure 19: Case of a disjointed Pareto front.

B.5 Case of a Disjointed Pareto Front

In order to gain a more thorough comprehension of our approach to optimizing loss functions for Pareto set learning (PSL), we investigate a scenario where the Pareto front is disjointed. In such a scenario, it is noted that the preference vector still has an intersection point with the attainment surface (defined in Equation (21)), as illustrated by the blue curve in Figure 19. Equation (6) now measures the volume within the attainment surface and the reference point r, which is just the hypervolume of a disjointed Pareto front $\mathcal{H}_r(\mathcal{F}^*)$.

For a disjointed Pareto front, the quantity $\rho_{\mathcal{X}}(\theta)$ denotes the distance between r and the attainment surface associated with angle θ. Specifically, in Figure 19, the black dot represents the solution for this scenario. The integral of the distance function $\rho_{\mathcal{X}}(\theta)$ still returns the hypervolume of a disjointed Pareto front, which satisfies our purpose in this paper.

However, disjointed Pareto fronts in Pareto set learning overemphasize boundary solutions which may result in unpredictable outcomes. For disjointed Pareto fronts, it is recommended to adaptively adjust the preference distribution (which is set to be uniform in our experiments).

B.6 Pareto Front Hypervolume Calculation (Type2)

In this subsection, we define region A as the set of points dominating the Pareto front,

$$A = \{q \mid \exists p \in \mathcal{F}^* : p \leq q \text{ and } q \geq p^{\text{ideal}}\}. \tag{22}$$

To ensure consistency with the notation used in the main paper, we use the notation $\Lambda(\cdot)$ to represent the Lebesgue measure of a set. From a geometric perspective, as illustrated in Figure 20, it can be observed that:

$$\Lambda(A) + \mathcal{H}_r(\beta) = \prod_{i=1}^{m} (r_i - y_i^{\text{ideal}}). \tag{23}$$

The volume of A can be calculated in a polar coordinate as follows,

$$\Lambda(A) = c_m \int_{(0,\frac{\pi}{2})^{m-1}} \overline{\rho}_{\mathcal{X}}(\theta)^m d\theta, \tag{24}$$

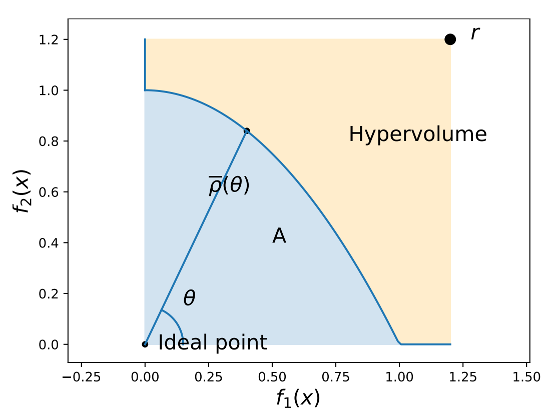


Figure 20: The hypervolume calculation (Type2).

where c_m is a constant and $\overline{\rho}_{\mathcal{X}}(\theta)$ represents the distance from the ideal point to the Pareto front at angle θ. This distance function $\overline{\rho}_{\mathcal{X}}(\theta)$ is obtained by solving the optimization problem assuming that any radius from θ intersects with the Pareto front.

Problem 1.

$$\overline{\rho}_{\mathcal{X}}(\theta) = \min_{x \in \mathcal{X}} \overline{\rho}_{\mathcal{X}}(\theta, x) = \min_{x \in \mathcal{X}} \max_{i \in [m]} \left\{ \frac{f_i(x) - y_i^{ideal}}{\lambda_i(\theta)} \right\}, \quad \theta \in (0, \frac{\pi}{2})^{m-1}. \tag{25}$$

The relationship between preference λ and the polar angle θ is as follows:

$$\begin{cases} \lambda_1(\theta) = \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{m-1} \\ \lambda_2(\theta) = \sin\theta_1 \sin\theta_2 \ldots \cos\theta_{m-1} \\ \ldots \\ \lambda_m(\theta) = \cos\theta_1. \end{cases} \tag{26}$$

Combining Equation (24) and (25) implies that $\overline{\mathcal{H}}_r(\beta)$ can be estimated as an expectation problem,

$$\overline{\mathcal{H}}_r(\beta) = \prod_{i=1}^{m}(r_i - y_i^{\text{ideal}}) - \frac{1}{m} c_m \mathbb{E}_{\theta \sim \text{Unif}(\Theta)}[\overline{\rho}_{\mathcal{X}}(x_\beta(\theta), \theta)^m]. \tag{27}$$

B.7 Proof of Proposition 3

Proof. It can be observed that Equation (6) in the main paper implies the following equation,

$$-\rho(x,\theta) = \max_{i\in[m]} \left\{ \frac{f_i(x) - r_i}{\lambda_i(\theta)} \right\}. \tag{28}$$

When all objectives f_i's are convex, function $-\rho(x,\theta)$ is also convex yet non-smooth, and hence $\rho(x,\theta)$ is concave. When f_i's are differentiable, $-\rho(x,\theta)$ possesses a natural subgradient denoted as d that is formulated as $d = \frac{\partial f_j(x)}{\partial x}\frac{1}{\lambda_i(\theta)}$, where $j = \arg\max_{i\in[m]}\{\frac{f_i(x)-r_i}{\lambda_i(\theta)}\}$. The subgradient d can be iteratively updated to converge on the global optima of $\rho_{\mathcal{X}}(\theta)$ in a $\mathcal{O}(1/\epsilon^2)$ rate, as described in [48, 49].

When all objectives f_i's are quasi-convex, $-\rho(x,\theta)$, which is a point-wise max of quasi-convex functions, is quasi-convex. And, hence $\rho(x,\theta)$ is quasi-concave. □

B.8 Proof of $\rho_\beta(\theta)$ is Quasi-Concave w.r.t. x

Proof. Proposition 3 rigorously demonstrates that the function $-\rho(x,\theta)$ is convex for any given value of θ. Furthermore, consider the function $h(x) : \mathbb{R} \to \mathbb{R}$ which may be defined as follows,

$$h(u) = \begin{cases} u^m & \text{if } u \geq 0 \\ u & \text{otherwise} \end{cases}. \tag{29}$$

It is clear $h(x)$ is a non-decreasing function, and $g(x) = -\rho_\beta(x) = h\circ(-\rho(x,\theta))$. Since $(-\rho(x,\theta))$ is convex, then, for any α, the set $S_\alpha(-\rho(x,\theta))$, as defined as follows, is convex.

$$S_\alpha(-\rho(x,\theta)) = \{x| - \rho(x,\theta) \leq \alpha\}. \tag{30}$$

Let $\gamma = h(\alpha)$. Then for any γ, the set $S_\gamma(h\circ(-\rho(x,\theta)))$, which equals to $S_\alpha(-\rho(x,\theta))$, is convex. This indicates that $h\circ(-\rho(x,\theta))$ is quasi-convex, and as a result $\rho_\beta(\theta)$ is quasi-convex w.r.t. x. □

B.9 Proof of Theorem 1

Definitions and preliminaries. The proof will heavily utilize the existing results on Rademacher complexity of MLPs. We will first provide some useful definitions and facts. We start with the definition of Rademacher complexity as follows:

Definition 2 (Rademacher complexity, Definition 13.1 in [50])**.** *Given a set of vectors $V \subseteq \mathbb{R}^n$, we define the (unnormalized) Rademacher complexity as*

$$\text{URad}(V) := \mathbb{E} \sup_{u\in V}\langle \epsilon, u\rangle,$$

where each coordinate ϵ_i is an i.i.d. Rademacher random variable, meaning $\Pr[\epsilon_i = +1] = \frac{1}{2} = \Pr[\epsilon_i = -1]$. *Furthermore, we can accordingly discuss the behavior of a function class $\mathcal{G}$ on $S = \{z_i\}_{i=1}^N$ by using the following set:*

$$\mathcal{G}_{|S} := \{(g(z_1), \ldots, f(z_N)) : g \in \mathcal{G}\} \subseteq \mathbb{R}^N,$$

and its Rademacher complexity is

$$\text{URad}\left(\mathcal{G}_{|S}\right) = \mathbb{E}_\epsilon \sup_{u\in\mathcal{G}_{|S}} \langle \epsilon, u\rangle = \mathbb{E}_\epsilon \sup_{g\in\mathcal{G}} \sum_i \epsilon_i g(z_i).$$

Utilizing Rademacher complexity, we can conveniently bound the generalization error via the following theorem:

Theorem 2 (Uniform Generalization Error, Theorem 13.1 and Corollary 13.1 in [50])**.** *Let $\mathcal{G}$ be given with $g(z) \in [a, b]$ a.s. $\forall g \in \mathcal{G}$. We collect i.i.d. samples $S = \{z_i\}_{i=1}^N$ from the law of random variable Z. With probability $\geq 1 - \delta$,*

$$\sup_{g \in \mathcal{G}} \mathbb{E} g(Z) - \frac{1}{N} \sum_i g\left(z_i\right) \leq \frac{2}{N} \operatorname{URad}\left(\mathcal{G}_{|S}\right) + 3(b-a) \sqrt{\frac{\ln(2/\delta)}{2N}}.$$

Specifically, the Rademacher complexity in using MLP is provided by the following theorem:

Theorem 3 (Rademacher complexity of MLP, Theorem 1 in [51])**.** *Let* 1*-Lipschitz positive homogeneous activation σ_i be given, and*

$$\mathcal{G}^{MLP} := \left\{\theta \mapsto \sigma_L\left(W_L \sigma_{L-1}\left(\cdots \sigma_1\left(W_1 \theta\right) \cdots\right)\right) : \left\|W_i\right\|_{\mathrm{F}} \leq B_w\right\}$$

Then

$$\operatorname{URad}\left(\mathcal{G}_{|S}^{MLP}\right) \leq B_w^L \|X_\theta\|_F (1 + \sqrt{2L \ln(2)}).$$

We can then utilize the following composition character of Rademacher complexity, to help induce the final Rademacher complexity of hypervolume.

Lemma 2 (Rademacher complexity of compositional function class, adapted from Lemma 13.3 in [50])**.** *Let $g : \Theta \to \mathbb{R}^n$ be a vector of n multivariate functions $g^{(1)}, g^{(2)}, \ldots, g^{(n)}$, $\mathcal{G}$ denote the function class of g, and further $\mathcal{G}^{(j)}$ be the function class of $g^{(j)}, \forall j$. We have a "partially Lipschitz continuous" function $\ell(g(\theta), \theta)$ so that $|\ell(g_1(\theta), \theta) - \ell(g_2(\theta), \theta)| \leq L_\ell \|g_1(\theta) - g_2(\theta)\|$ for all $g_1, g_2 \in \mathcal{G}$ and a certain $L_\ell > 0$; the associated function class of ℓ is denoted as $\mathcal{G}^\ell$. We then have*

$$\operatorname{URad}\left(\mathcal{G}_{|S}^{\ell}\right) \leq \sqrt{2} L_\ell \sum_{j=1}^n \operatorname{URad}\left(\mathcal{G}_{|S}^{(j)}\right).$$

Proof. This proof extends Lemma 13.3 in [50] for vector-valued g and "partially Lipschitz continuous" ℓ. We first similarly have

$$\begin{aligned}
\operatorname{URad}\left(\mathcal{G}_{|S}^{\ell}\right) &= \mathbb{E} \sup_{g \in \mathcal{G}} \sum_i \epsilon_i \ell(g(\theta_i), \theta_i) \\
&= \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:N}} \sup_{f, h \in \mathcal{G}} \left(\ell(f(\theta_1), \theta_1) - \ell(h(\theta_1), \theta_1) + \sum_{i=2}^N \epsilon_i \left(\ell(f(\theta_i), \theta_i) + \ell(h(\theta_i), \theta_i) \right) \right) \\
&\leq \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:N}} \sup_{f, h \in \mathcal{G}} \left(L_\ell \|f(\theta_1) - h(\theta_1)\| + \sum_{i=2}^N \epsilon_i \left(\ell(f(\theta_i), \theta_i) + \ell(h(\theta_i), \theta_i) \right) \right) \\
&\leq \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon} \sup_{f, h \in \mathcal{G}} \left(L_\ell \sqrt{2} \Big| \sum_{j=1}^n \epsilon_1^{(j)} (f^{(j)}(\theta_1) - h^{(j)}(\theta_1)) \Big| + \right. \\
&\qquad \left. \sum_{i=2}^N \epsilon_i \left(\ell(f(\theta_i), \theta_i) + \ell(h(\theta_i), \theta_i) \right) \right),
\end{aligned}$$

where $\epsilon_1^{(j)}$'s are new i.i.d. Rademacher variables; the last inequality comes from Proposition 6 in [52] (see Equations (5)-(10) in [52] for more details). We can then get rid of the absolute value by considering swapping f and h,

$$\begin{aligned}
&\sup_{f,h\in\mathcal{G}}\left(\sqrt{2}L_\ell|\sum_{j=1}^{n}\epsilon_1^{(j)}(f^{(j)}(\theta_1)-h^{(j)}(\theta_1))|+\sum_{i=2}^{N}\epsilon_i\left(\ell(f(\theta_i),\theta_i)+\ell(h(\theta_i),\theta_i)\right)\right)\\
&=\max\left\{\sup_{f,h\in\mathcal{G}}\left(\sqrt{2}L_\ell\sum_{j=1}^{n}\epsilon_1^{(j)}(f^{(j)}(\theta_1)-h^{(j)}(\theta_1))+\sum_{i=2}^{N}\epsilon_i\left(\ell(f(\theta_i),\theta_i)+\ell(h(\theta_i),\theta_i)\right)\right),\right.\\
&\qquad\left.\sup_{f,h\in\mathcal{G}}\left(\sqrt{2}L_\ell\sum_{j=1}^{n}\epsilon_1^{(j)}(h^{(j)}(\theta_1)-f^{(j)}(\theta_1))+\sum_{i=2}^{N}\epsilon_i\left(\ell(f(\theta_i),\theta_i)+\ell(h(\theta_i),\theta_i)\right)\right)\right\}\\
&=\sup_{f,h\in\mathcal{G}}\left(\sqrt{2}L_\ell\sum_{j=1}^{n}\epsilon_1^{(j)}(f^{(j)}(\theta_1)-h^{(j)}(\theta_1))+\sum_{i=2}^{N}\epsilon_i\left(\ell(f(\theta_i),\theta_i)+\ell(h(\theta_i),\theta_i)\right)\right).
\end{aligned}$$

We can thus upper bounded $\mathrm{URad}\left(\mathcal{G}^{\ell}_{|S}\right)$ by

$$\begin{aligned}
&\frac{1}{2}\mathop{\mathbb{E}}_{\epsilon}\sup_{f,h\in\mathcal{G}}\left(\sqrt{2}L_\ell\sum_{j=1}^{n}\epsilon_1^{(j)}(f^{(j)}(\theta_1)-h^{(j)}(\theta_1))+\sum_{i=2}^{N}\epsilon_i\left(\ell(f(\theta_i),\theta_i)+\ell(h(\theta_i),\theta_i)\right)\right)\\
=&\mathop{\mathbb{E}}_{\epsilon}\sup_{g\in\mathcal{G}}\left(\sqrt{2}L_\ell\sum_{j=1}^{n}\epsilon_1^{(j)}g^{(j)}(\theta_1)+\sum_{i=2}^{N}\epsilon_i\ell(g(\theta_i),\theta_i)\right),
\end{aligned}$$

Repeating this procedure for the other coordinates, we can further have

$$\mathrm{URad}\left(\mathcal{G}^{\ell}_{|S}\right)\le\sqrt{2}L_\ell\mathop{\mathbb{E}}_{\epsilon}\sup_{g\in\mathcal{G}}\left(\sum_{i=1}^{N}\sum_{j=1}^{n}\epsilon_i^{(j)}g^{(j)}(\theta_i)\right)\le\sqrt{2}L_\ell\sum_{j=1}^{n}\mathop{\mathbb{E}}_{\epsilon}\sup_{g^{(j)}\in\mathcal{G}^{(j)}}\left(\sum_{i=1}^{N}\epsilon_i^{(j)}g^{(j)}(\theta_i)\right),$$

which leads to our claim in the lemma. □

Proof of Theorem 1. We are now geared up for the complete proof.

Proof. We first introduce the sketch of the proof. We mainly utilize Theorem 2 to attain the claimed results in Theorem 1. Specifically, we set the random sample set $S=\{\theta_i\}_{i=1}^N$, the function class $\mathcal{G}$ as $\{\theta\mapsto c_m\rho(x_\beta(\theta),\theta)^m\}$ (the assumption $r_i-f_i(x)\in[b,B]$ indicates that $\rho(x,\theta)=\min_{i\in[m]}\{\frac{r_i-f_i(x)}{\lambda_i(\theta)}\}\ge b\ge 0$ and by the definition in Equation (7), $\rho_\beta(\theta)$ is thus always $\rho(x(\theta),\theta)^m$; $x_\beta(\cdot)$ is an L-layer MLP to be specified later). Applying Theorem 2, we can obtain that with probability at least $1-\frac{\delta}{2}$,

$$\sup_{g\in\mathcal{G}}\mathbb{E}_\theta g(\theta)-\frac{1}{N}\sum_i g\left(\theta_i\right)\le\frac{2}{N}\,\mathrm{URad}\left(\mathcal{G}_{|S}\right)+3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}},$$

where the definition of URad and $\mathcal{G}_{|S}$ can be found in Definition 2. Simply replacing $\mathcal{G}$ with $-\mathcal{G}:=\{-g:g\in\mathcal{G}\}$, we can have the inequality of the other direction with probability at least $1-\frac{\delta}{2}$:

$$\begin{aligned}
&\sup_{g\in-\mathcal{G}}\mathbb{E}_\theta g(\theta)-\frac{1}{N}\sum_i g\left(\theta_i\right)\le\frac{2}{N}\,\mathrm{URad}\left(-\mathcal{G}_{|S}\right)+3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}}\\
\Rightarrow&\sup_{g\in\mathcal{G}}\mathbb{E}_\theta-g(\theta)-\frac{1}{N}\sum_i -g\left(\theta_i\right)\le\frac{2}{N}\,\mathrm{URad}\left(-\mathcal{G}_{|S}\right)+3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}}\\
\Rightarrow&\sup_{g\in\mathcal{G}}\frac{1}{N}\sum_i g\left(\theta_i\right)-\mathbb{E}_\theta g(\theta)\le\frac{2}{N}\,\mathrm{URad}\left(\mathcal{G}_{|S}\right)+3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}},
\end{aligned}$$

where we apply the property $\mathrm{URad}\left(-\mathcal{G}_{|S}\right)=\mathrm{URad}\left(\mathcal{G}_{|S}\right)$. We thus, with probability at least $1-\delta$ (as a result of union bound), can upper bound $\sup_{g\in\mathcal{G}}\left|\mathbb{E}_\theta g(\theta)-\frac{1}{N}\sum_i g\left(\theta_i\right)\right|$ by

$$\begin{aligned}&\max\left\{\sup_{g\in\mathcal{G}}\mathbb{E}_\theta g(\theta)-\frac{1}{N}\sum_i g\left(\theta_i\right),\sup_{g\in\mathcal{G}}\frac{1}{N}\sum_i g\left(\theta_i\right)-\mathbb{E}_\theta g(\theta)\right\}\\ \leq&\frac{2}{N}\,\mathrm{URad}\left(\mathcal{G}_{|S}\right)+3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}}.\end{aligned}$$

For the next step, we will upper bound $\mathrm{URad}\left(\mathcal{G}_{|S}\right)$ by analyzing the structure of $c_m\rho(x_\beta(\theta),\theta)^m$ and utilizing the existing bound (see Theorem 3) for Rademacher complexity of MLP x_β.

The main idea of controlling $\mathrm{URad}\left(\mathcal{G}_{|S}\right)$ is to obtain the "partially Lipschitz continuity" that $|\rho(x_\beta(\theta),\theta)-\rho(x_{\beta'}(\theta),\theta)|\leq L_\rho\|x_\beta(\theta)-x_{\beta'}(\theta)\|$ for a certain $L_\rho>0$; with the "partially Lipschitz continuity" we can apply Lemma 2 and obtain the desired bound. For simplicity, we denote $x_\beta(\theta),x_{\beta'}(\theta)$ respectively as x,x', and use λ_j's as shorthand for $\lambda_j(\theta)$'s. We now expand the difference $|\rho(x_\beta(\theta),\theta)-\rho(x_{\beta'}(\theta),\theta)|$ as:

$$\begin{aligned}&\left|\min_{j\in[m]}\frac{r_j-f_j(x)}{\lambda_j}-\min_{k\in[m]}\frac{r_k-f_k(x')}{\lambda_k}\right|\\ =&\max\left\{\min_{j\in[m]}\frac{r_j-f_j(x)}{\lambda_j}-\min_{k\in[m]}\frac{r_k-f_k(x')}{\lambda_k},\min_{k\in[m]}\frac{r_k-f_k(x')}{\lambda_k}-\min_{j\in[m]}\frac{r_j-f_j(x)}{\lambda_j}\right\}.\end{aligned}$$

If we respectively denote the minima index of the two finite-term minimization as j^* and k^*, we can then upper bound $|\rho(x_\beta(\theta),\theta)-\rho(x_{\beta'}(\theta),\theta)|$ by

$$\begin{aligned}&\max\left\{\frac{r_{k^*}-f_{k^*}(x)}{\lambda_{k^*}}-\frac{r_{k^*}-f_{k^*}(x')}{\lambda_{k^*}},\frac{r_{j^*}-f_{j^*}(x')}{\lambda_{j^*}}-\frac{r_{j^*}-f_{j^*}(x)}{\lambda_{j^*}}\right\}\\ =&\max\left\{\frac{f_{k^*}(x')-f_{k^*}(x)}{\lambda_{k^*}},\frac{f_{j^*}(x)-f_{j^*}(x')}{\lambda_{j^*}}\right\}\leq\max_{j\in\{j^*,k^*\}}\frac{|f_j(x)-f_j(x')|}{\lambda_j}\\ \leq&\max_{j\in\{j^*,k^*\}}\frac{L_f|x-x'|}{\lambda_j}.\end{aligned}$$

We note there is a special property for λ_j when j is the minima index: as $\|\lambda\|=1$, there must be a certain $\lambda_j\geq 1/\sqrt{m}$, and since $b\leq r_j-f_j(x)\leq B,\forall j$, we have

$$\frac{b}{\lambda_{j^*}}\leq\frac{r_{j^*}-f_{j^*}(x')}{\lambda_{j^*}}\leq\frac{B}{1/\sqrt{m}}\Rightarrow\lambda_{j^*}\geq\frac{b}{\sqrt{m}B}.$$

With this special property, we obtain

$$|\rho(x_\beta(\theta),\theta)-\rho(x_{\beta'}(\theta),\theta)|\leq\frac{\sqrt{m}B}{b}L_f|x-x'|.$$

We further have

$$\begin{aligned}&|c_m\rho(x_\beta(\theta),\theta)^m-c_m\rho(x_{\beta'}(\theta),\theta)^m|\\ =&c_m\left|\rho(x_\beta(\theta),\theta)-\rho(x_{\beta'}(\theta),\theta)\right|\left(\sum_{k=1}^m\rho(x_\beta(\theta),\theta)^{m-k}\rho(x_{\beta'}(\theta),\theta)^{k-1}\right)\\ \leq&c_m\frac{\sqrt{m}B}{b}L_f|x-x'|m(B\sqrt{m})^{m-1}=c_m\frac{m}{b}(B\sqrt{m})^mL_f|x-x'|,\end{aligned}$$

which establishes the "partially Lipschitz continuity". We can then apply Lemma 2 and have

$$\begin{aligned}\mathrm{URad}\left(\mathcal{G}_{|S}\right)&\leq\sqrt{2}c_m\frac{m}{b}(B\sqrt{m})^mL_fn\,\mathrm{URad}\left(\mathcal{G}_{|S}^{\mathrm{MLP}}\right)\\ &\leq\sqrt{2}c_m\frac{m}{b}(B\sqrt{m})^mL_fn\cdot B_w^L\|X_\theta\|_F(1+\sqrt{2L\ln(2)}).\end{aligned}$$

Combining the pieces above, we finally have

$$
\begin{aligned}
&\sup_{g\in\mathcal{G}} |\mathbb{E}_\theta g(\theta) - \frac{1}{N}\sum_i g\left(\theta_i\right)| \\
\leq& \frac{2}{N}\,\mathrm{URad}\left(\mathcal{G}_{|S}\right) + 3c_m(B\sqrt{m})^m\sqrt{\frac{\ln(4/\delta)}{2N}} \\
\leq& c_m(B\sqrt{m})^m\left(\frac{2\sqrt{2}mn}{Nb}L_f\cdot B_w^L\|X_\theta\|_F(1+\sqrt{2L\ln(2)}) + 3\sqrt{\frac{\ln(4/\delta)}{2N}}\right),
\end{aligned}
$$

which is the generalization error bound we claim. □

B.10 Upper Bound of $\rho_{\mathcal{X}}(\theta)$

In this subsection, we prove that the distance function $\rho_{\mathcal{X}}(\theta)$ is bounded by the following inequality,

$$\rho_{\mathcal{X}}(\theta) \leq Bm^{1/2}, \tag{31}$$

when $r_i - f_i(x) \leq B,\ \forall x \in \mathcal{X}, \forall i \in [m]$ and $||\lambda(\theta)|| = 1$.

Proof. We show that the following inequalities hold,

$$
\begin{aligned}
\rho_{\mathcal{X}}(\theta) &\leq \max_{x\in\mathcal{X},||\lambda(\theta)||=1}\left(\min_{i\in[m]}\{\frac{r_i - f_i(x)}{\lambda_i(\theta)}\}\right) \\
&\leq \max_{||\lambda(\theta)||=1}\left(\min_{i\in[m]}\{\frac{B}{\lambda_i(\theta)}\}\right) \\
&\leq \frac{B}{m^{-1/2}} = Bm^{1/2}.
\end{aligned}
\tag{32}
$$

The transition from line one to line two is due to the fact that the inequality $r_i - f_i(x) \leq B$ holds for all $x \in \mathcal{X}$ and for all $i \in [m]$. The transition from line two to line three is $\max_{||\lambda(\theta)||=1}\left(\min_{i\in[m]}\{\frac{B}{\lambda_i(\theta)}\}\right)$ is an optimization problem under the constraint $||\lambda(\theta)|| = 1$. The upper bound for this optimization is when $\lambda_i = \ldots = \lambda_m = m^{-1/2}$. □

Let $\mathcal{Z}(\theta) = c_m\rho_{\mathcal{X}}(\theta)^m$, as a corollary, $\mathcal{Z}(\theta) \leq c_m B^m m^{m/2}$.

B.11 Gradients of HV-PSL

In this subsection, we present the analytical expression for $\nabla_\beta\mathcal{H}_r(\beta)$ to ensure completeness. The gradient for PSL-HV1 can be computed using the chain rule, which yields:

$$
\nabla_\beta\mathcal{H}_r(\beta) = \begin{cases}
mc_m\mathbb{E}_{\theta\sim\mathrm{Unif}(\Theta)}[\rho(x_\beta(\theta),\theta)^{m-1}\underbrace{\frac{\partial\rho(x_\beta(\theta),\theta)}{\partial x_\beta(\theta)}}_{1\times n}\underbrace{\frac{\partial x_\beta(\theta)}{\partial\beta}}_{n\times d}], & \rho(x_\beta(\theta),\theta)\geq 0. \\
c_m\mathbb{E}_{\theta\sim\mathrm{Unif}(\Theta)}[\underbrace{\frac{\partial\rho(x_\beta(\theta),\theta)}{\partial x_\beta(\theta)}}_{1\times n}\underbrace{\frac{\partial x_\beta(\theta)}{\partial\beta}}_{n\times d}], & \text{Otherwise.}
\end{cases}
\tag{33}
$$

The gradient of PSL-HV2 can be calculated by,

$$
\nabla_\beta\mathcal{H}_r(\beta) = -mc_m\mathbb{E}_{\theta\sim\mathrm{Unif}(\Theta)}[\overline{\rho}_{\mathcal{X}}(x_\beta(\theta),\theta)^{m-1}]\underbrace{\frac{\partial\overline{\rho}_{\mathcal{X}}(x_\beta(\theta),\theta)}{\partial x_\beta(\theta)}}_{1\times n}\underbrace{\frac{\partial x_\beta(\theta)}{\partial\beta}}_{n\times d}]. \tag{34}
$$

B.12 Relationship between Hypervolume and Decomposition based Multiobjective Optimization

In this subsection, we will explore the fundamental relationship between hypervolume-based and decomposition-based multiobjective optimization. Prior to our study, it was commonly acknowledged that there were three primary multiobjective optimization methods: Pareto-based [9], hypervolume-based [30], and decomposition-based methods [8].

The present paper yields a result by establishing a correlation between hypervolume and decomposition-based approach in scenarios where the number of preference $\lambda(\theta)$ is considerably high. Previous methods mainly consider two decomposition functions, namely linear scalarization and Tchebycheff. Actually, we only need to make two modifications for the classical decomposition-based method in [8],

1. Sampling the polar angles $\theta^{(i)}$ from S_+^{m-1}.
2. For each sampled angle $\theta^{(i)}$, maximizing the scalarization function $\rho_{\mathcal{X}}(\theta^{(i)}) = \max_{i\in[m]}\{\frac{r_i - f_i(x)}{\lambda_i(\theta^{(i)})}\}$.

Subsequently, upon optimizing each scalarization function, it becomes feasible to constrain the deviation between the empirical mean of $c_m \rho_{\mathcal{X}}(\theta^{(i)})^m$ and the hypervolume of the Pareto front to a small value with a high level of certainty. This is elaborated by Equation (9) in the main manuscript.

References

[1] Yong Zheng and David Xuejun Wang. A survey of recommender systems with multi-objective optimization. *Neurocomputing*, 474:141–153, 2022.

[2] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. A multi-objective optimization framework for multi-stakeholder fairness-aware recommendation. *ACM Transactions on Information Systems*, 41(2):1–29, 2022.

[3] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14610–14621, 2019.

[4] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10607–10616. PMLR, 2020.

[5] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 525–536, 2018.

[6] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6597–6607. PMLR, 2020.

[7] Michael Ruchte and Josif Grabocka. Scalable pareto front approximation for deep multi-objective learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1306–1311. IEEE, 2021.

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

[9] 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.

[10] Xingchao Liu, Xin Tong, and Qiang Liu. Profiling pareto front with multi-objective stein variational gradient descent. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14721–14733, 2021.

[11] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12037–12047, 2019.

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

[13] William Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

[14] Aviv Navon, Aviv Shamsian, Ethan Fetaya, and Gal Chechik. Learning the pareto front with hypernetworks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[15] Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable pareto multi-task learning. *ArXiv preprint*, abs/2010.06313, 2020.

[16] Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[17] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings 4*, pages 862–876. Springer, 2007.

[18] Michael Emmerich, Nicola Beume, and Boris Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 62–76. Springer, 2005.

[19] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28, 2007.

[20] Timo M Deist, Monika Grewal, Frank JWM Dankers, Tanja Alderliesten, and Peter AN Bosman. Multi-objective learning using hv maximization. In *Evolutionary Multi-Criterion Optimization: 12th International Conference, EMO 2023, Leiden, The Netherlands, March 20–24, 2023, Proceedings*, pages 103–117. Springer, 2023.

[21] Michael Emmerich and André 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.

[22] Moksh Jain, Sharath Chandra Raparthy, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. *ArXiv preprint*, abs/2210.12765, 2022.

[23] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2022.

[24] Alan Q Wang, Adrian V Dalca, and Mert R Sabuncu. Computing multiple image reconstructions with a single hypernetwork. *ArXiv preprint*, abs/2202.11009, 2022.

[25] Xi Lin, Zhiyuan Yang, and Qingfu Zhang. Pareto set learning for neural multi-objective combinatorial optimization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[26] Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 11–20. PMLR, 2019.

[27] Toygun Basaklar, Suat Gumussoy, and Umit Y Ogras. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. *ArXiv preprint*, abs/2208.07914, 2022.

[28] Joey Hong, Aviral Kumar, and Sergey Levine. Confidence-conditioned value functions for offline reinforcement learning. *ArXiv preprint*, abs/2212.04607, 2022.

[29] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. User-interactive offline reinforcement learning. *ArXiv preprint*, abs/2205.10629, 2022.

[30] Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. The hypervolume indicator: Problems and algorithms. *ArXiv preprint*, abs/2005.00515, 2020.

[31] Hisao Ishibuchi, Noritaka Tsukamoto, Yuji Sakane, and Yusuke Nojima. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In *2009 IEEE Congress on Evolutionary Computation*, pages 530–537. IEEE, 2009.

[32] Jingda Deng and Qingfu Zhang. Approximating hypervolume and hypervolume contributions using polar coordinate. *IEEE Transactions on Evolutionary Computation*, 23(5):913–918, 2019.

[33] Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11096–11105. PMLR, 2020.

[34] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[35] Hisao Ishibuchi, Ryo Imada, Naoki Masuyama, and Yusuke Nojima. Use of two reference points in hypervolume-based evolutionary multiobjective optimization algorithms. In *Parallel Problem Solving from Nature–PPSN XV: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part I 15*, pages 384–396. Springer, 2018.

[36] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, 26(3):411–440, 2018.

[37] Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. *arXiv e-prints*, pages arXiv–2210, 2022.

[38] David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithm test suites. In *Proceedings of the 1999 ACM symposium on Applied computing*, pages 351–357, 1999.

[39] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[40] Harvey J Greenberg and William P Pierskalla. A review of quasi-convex functions. *Operations research*, 19(7):1553–1570, 1971.

[41] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6231–6239, 2017.

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

[43] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

[44] Simone Parisi, Matteo Pirotta, and Jan Peters. Manifold-based multi-objective policy search with sample reuse. *Neurocomputing*, 263:3–14, 2017.

[45] Xiaoliang Ma, Qingfu Zhang, Guangdong Tian, Junshan Yang, and Zexuan Zhu. On tchebycheff decomposition approaches for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244, 2017.

[46] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[47] Xiangyi Chen, Zhiwei Steven Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[48] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

[49] Arkadi Nemirovski. Efficient methods in convex programming. *Lecture notes*, 1994.

[50] Matus Telgarsky. Deep learning theory lecture notes, 2020.

[51] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.

[52] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 3–17. Springer, 2016.