Offline Multi-Objective Optimization

Ke Xue [*1 2] **Rong-Xi Tan** [*1 2] **Xiaobin Huang** [1 2] **Chao Qian** [1 2]

Abstract

Offline optimization aims to maximize a black-box objective function with a static dataset and has wide applications. In addition to the objective function being black-box and expensive to evaluate, numerous complex real-world problems entail optimizing multiple conflicting objectives, i.e., multi-objective optimization (MOO). Nevertheless, offline MOO has not progressed as much as offline single-objective optimization (SOO), mainly due to the lack of benchmarks like Design-Bench for SOO. To bridge this gap, we propose a first benchmark for offline MOO, covering a range of problems from synthetic to real-world tasks. This benchmark provides tasks, datasets, and open-source examples, which can serve as a foundation for method comparisons and advancements in offline MOO. Furthermore, we analyze how the current related methods can be adapted to offline MOO from four fundamental perspectives, including data, model architecture, learning algorithm, and search algorithm. Empirical results show improvements over the best value of the training set, demonstrating the effectiveness of offline MOO methods. As no particular method stands out significantly, there is still an open challenge in further enhancing the effectiveness of offline MOO. We finally discuss future challenges for offline MOO, with the hope of shedding some light on this emerging field. Our code is available at https://github.com/lamda-bbo/offline-moo.

1. Introduction

Creating new designs to optimize specific properties is a widespread challenge, encompassing various domains such as real-world engineering design (Tanabe & Ishibuchi, 2020), protein design (Khan et al., 2023), and molecule design (Stanton et al., 2022). Many methods generate new designs by iteratively querying an unknown objective function that maps a design to its property score. However, in real-world situations, evaluating the objective function can be time-consuming, costly, or even dangerous (Dara et al., 2022). To optimize for the next candidate design based on accumulated data, a rational approach prefers to build a model, use it to guide the search, and select a suitable candidate for the evaluation. This approach is known as offline model-based optimization (Trabucco et al., 2022).

Offline model-based optimization solely permits access to an offline dataset and does not permit iterative online evaluation (i.e., only one batch of real evaluations), which presents notable challenges in comparison to more commonly studied online optimization. A common approach is to train a deep neural network (DNN) model $f_\theta(\cdot)$ on a static dataset and use the trained DNN as a proxy (also known as a surrogate model). The DNN proxy enables gradient descent on existing designs, which can result in an improved solution that is even better than the previously seen best one sometimes. However, this approach has a drawback: the trained proxy is prone to out-of-distribution problems, i.e., it makes inaccurate predictions when applied to data points that deviate significantly from the training distribution. Besides, in some cases, the learned proxy has a non-smooth landscape, posing challenges to optimize in it. Many recent studies try to address these issues from different perspectives, e.g., COMs (Trabucco et al., 2021) uses adversarial training to create a smooth proxy; RoMA (Yu et al., 2021) employs a local smoothness prior to alleviate the fragility of the proxy and achieves robust estimation by model adaptation; Tri-Mentoring (Chen et al., 2023a) effectively utilizes weak ranking supervision signals among proxies and achieves a robust ensemble of proxies by an adaptive soft-labeling module; just to name a few.

In addition to the objective function being black-box and the evaluations being costly, numerous complex real-world problems entail optimizing multiple objectives, frequently with conflicting requirements, which can be formulated as multi-objective optimization (MOO) problems (Miettinen, 1998; Ehrgott, 2005). The goal of MOO is to find a set of solutions that represent the optimal trade-offs among

[*]Equal contribution [1]National Key Laboratory for Novel Software Technology, Nanjing University, China [2]School of Artificial Intelligence, Nanjing University, China. Correspondence to: Chao Qian <qianc@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

the various objectives, thereby significantly augmenting the complexity of the problem compared to single-objective optimization (SOO) which aims to obtain a single optimal solution. Indeed, MOO is a more prevalent problem than SOO. Many single-objective problems are essentially multi-objective in nature, but they are often converted into a single objective by assigning weights to multiple objectives, primarily due to the challenges associated with solving MOO (Stanton et al., 2022; Chen & Li, 2023).

Recently, researchers have recognized the significance of directly modeling MOO problems (Deb et al., 2002a; Daulton et al., 2020). The demand for offline MOO is also gradually increasing. However, the progress of offline MOO is far behind compared to offline SOO. Thanks to the remarkable benchmark Design-Bench (Trabucco et al., 2022), several advanced offline SOO algorithms have been proposed, which can perform well even in high-dimensional and complex search spaces (Chen et al., 2022; Qi et al., 2022; Yuan et al., 2023; Chen et al., 2023a; Krishnamoorthy et al., 2023; Kim et al., 2023; Chemingui et al., 2024; Yu et al., 2024; Uehara et al., 2024). Unfortunately, there has been no such benchmark available for offline MOO, which hinders its progress. Even for online MOO, most works conduct evaluations on synthetic functions with a few exceptions that include real-world applications. This calls for a much-needed push towards more challenging benchmarks for reliable evaluation of MOO, especially in the offline setting.

In this paper, we propose a first benchmark for offline MOO, where the tasks range from synthetic functions to real-world science and engineering problems, as shown in Figure 1. To facilitate future research, we release our benchmark tasks and datasets with a comprehensive evaluation of different approaches and open-source examples. Specifically, we analyze an offline MOO method from four fundamental perspectives including data, model architecture, learning algorithm, and search algorithm, and propose two types of potential methods, i.e., DNN-based and Gaussian process-based offline MOO, by learning techniques from related areas such as offline SOO and multi-objective Bayesian optimization. Experimental results show that the proposed methods can achieve better results than the optimal ones in the training set. However, as no single method stands out significantly, how to enhance the effectiveness of offline MOO remains open. Our work serves as a starting point for offline MOO, and we hope it can encourage more explorations in this emerging area.

Our contributions can be summarized as follows:

- We propose a first benchmark for offline MOO, providing not only a large amount of offline data but also commonly used MOO interfaces. This facilitates the integration of a wider range of problems and algorithms.

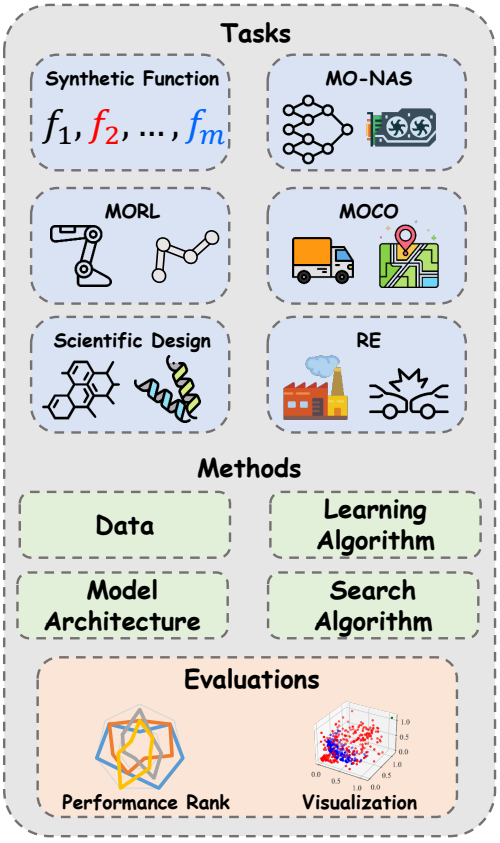

Figure 1. Benchmarks for offline MOO.

- We analyze an offline MOO method from four fundamental perspectives, including data, model architecture, learning algorithm, and search algorithm, and compare various implementations within a unified framework, making it convenient for researchers to compare their performance in a clear manner.
- We provide extensive empirical studies, and also discuss challenges and future directions of offline MOO.

2. Background

2.1. Offline Optimization

Given an offline-collected static dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, offline model-based optimization aims to find an optimal solution (also called "design" in many scenarios) $\boldsymbol{x}^*$ that minimizes the black-box objective function $f(\cdot)$, i.e., $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$.

A common approach for solving offline optimization problems is approximating the black-box objective function $f(\cdot)$ using a surrogate model, e.g., DNN. The parameters of DNN can be trained by minimizing the mean squared error between the predictions and the true scores. After that, the trained DNN model is used as a surrogate evaluator to optimize using a search algorithm, e.g., gradient descent.

Since offline optimization does not allow iterative real evaluations, the algorithm is expected to output a proper solution that is better than the best solution seen in the dataset. However, in practice, producing a single better design entirely from offline data is very difficult, so offline optimization methods are more commonly evaluated in terms of "P percentile of top K" performance (Kumar & Levine, 2020), where the algorithm produces K candidates and the P percentile objective value determines the final performance.

Many real-world tasks are inherently multi-objective, but they are usually simplified and formulated as single-objective problems. For example, neural architecture search (NAS) should not only maximize accuracy, but also minimize the scale of the model (Lu et al., 2023); protein design should take efficacy, toxicity, and yield into consideration simultaneously (Stanton et al., 2022). In this paper, we aim to highlight the importance and challenges of offline MOO, and provide a benchmark and comprehensive empirical studies on it.

2.2. Multi-Objective Optimization

First, we give a brief introduction to multi-objective optimization problems, which can be defined as

$$\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{x} = (x_1, \dots, x_D)$ is a solution, $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^m$ constitutes m objective functions, $\mathcal{X}$ is the solution space, and $\mathbb{R}^m$ is the objective space. For a non-trivial problem, no single solution can optimize all objectives at the same time, and we have to make a trade-off among them (Qian et al., 2013; Bian et al., 2023).

Definition 2.1. A solution $\boldsymbol{x}^*$ is Pareto-optimal with respect to Eq. (1), if $\nexists \boldsymbol{x} \in \mathcal{X}$ such that $\forall i : f_i(\boldsymbol{x}) \le f_i(\boldsymbol{x}^*)$ and $\exists i : f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$. The set of all Pareto-optimal solutions is called Pareto-optimal set (PS). The set of the corresponding objective vectors of PS, i.e., $\{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \text{PS}\}$, is called Pareto front (PF).

Instead of focusing on a single optimal solution in SOO, the goal of MOO is to find *a set of solutions* that can approximate the PF well. Next, we will briefly introduce two main kinds of methods for solving MOO problems.

Multi-objective evolutionary algorithm (MOEA). Evolutionary algorithms (Bäck, 1996; Zhou et al., 2019) have demonstrated their effectiveness in solving MOO problems. MOEA follows the population-based search by iterative parent selection, reproduction, and survivor selection, which can approximate the Pareto optimal solutions within one execution, with each solution in the population representing a unique trade-off among the objectives (Deb, 2001). Over the last decades, there have been a lot of well-known MOEAs developed (Coello et al., 2007). NSGA-II (Deb et al., 2002a) is a typical Pareto dominance-based MOEA, using fast non-dominated sorting for selecting solutions. MOEA/D (Zhang & Li, 2007) is a decomposition-based MOEA, converting an MOO problem into multiple SOO sub-problems through a number of weights, where neighboring solutions work cooperatively for the optimal solutions of the single-objective sub-problems. NSGA-III (Deb & Jain, 2013) is proposed to handle MOO problems with many objectives (having four or more objectives), by using reference points to assist the selection within non-dominated solutions.

Multi-objective Bayesian optimization (MOBO). Many real-world MOO tasks are expensive to evaluate. MOBO is suitable for these tasks due to its high sample-efficiency. Based on the observed data, MOBO learns a surrogate model, e.g., Gaussian process (GP) (Rasmussen & Williams, 2006), searches for new promising candidates based on an acquisition function built on the surrogate model, and queries the quality of these candidates with the ground truth black-box objectives. Existing MOBO methods mainly fall into the following three types. Hypervolume based methods consider the widely-used hypervolume metrics in acquisition function (Emmerich et al., 2006; Konakovic Lukovic et al., 2020; Daulton et al., 2021; 2023). Scalaraization based methods reduce the MO acquisition function into one or multiple SO problems via scalarization (Knowles, 2006; Zhang et al., 2009; Paria et al., 2020; Zhang & Golovin, 2020). Information-theoretic methods select points to reduce the uncertainty of the unknown Pareto front (Hernández-Lobato et al., 2016; Suzuki et al., 2020; Hvarfner et al., 2022; Qing et al., 2023). Besides these methods, there are also works addressing MOBO in other scenarios, such as high-dimensional space (Zhao et al., 2022) and sequence space (Stanton et al., 2022).

While MOO has made significant progress, most existing methods either use handcrafted mechanism and lack a learning mechanism (e.g., MOEA) or are unable to leverage a large amount of offline data for scalable learning (e.g., MOBO), restricting their applications in offline MOO tasks. Additionally, there is a lack of benchmark for offline MOO. Note that a good benchmark plays a crucial role in the advancement of a research field and the development of state-of-the-art algorithms, such as NASBench (Ying et al., 2019) and HPO-B (Arango et al., 2021) for BBO, D4RL (Fu et al., 2020) and NeoRL (Qin et al., 2022) for offline RL, and Design-bench (Trabucco et al., 2022) for offline SOO. In the following, we will propose the problem of offline MOO and provide a large-scale benchmark, covering a wide range of tasks and methods.

3. Offline MOO Benchmark

We present the problem formulation in Section 3.1 and the process of collecting the dataset for our Offline MOO Bench-

mark (Off-MOO-Bench) in Section 3.2. We will introduce the tasks and methods in our benchmark in Sections 4 and 5, respectively.

3.1. Offline MOO

Given an offline-collected static dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ denotes a solution and its objective vector, respectively, offline MOO aims to find a set of solutions to approximate the Pareto front of the MOO problem in Eq. (1). Similar to offline SOO, offline MOO only allows access to the offline dataset and does not permit iterative online evaluation. Besides, the MOO nature makes offline MOO more challenging.

Due to the goal of finding a set of solutions rather than a single solution, the commonly used measure "P percentile of top K" in offline SOO cannot be directly applied for offline MOO. In our experiments, each offline MOO algorithm first outputs a certain number of solutions (e.g., 256 and 32) to be evaluated. To report the "P percentile" measure, we use the NSGA-II selection procedure (i.e., first applying non-dominated sorting then selecting the top solutions) (Deb et al., 2002a) to eliminate the top $1 - P\%$ of solutions and report the remaining solutions' metrics as the evaluation results. There are two commonly used metrics in MOO, i.e., inverted generational distance (IGD) (Bosman & Thierens, 2003), which measures the distance between a solution set and the true Pareto front, and hypervolume (HV) (Zitzler & Thiele, 1998), which measures the volume of the objective space between a reference point and the objective vectors of a solution set, reflecting both convergence and diversity of the solution set. Because the calculation of IGD requires knowing the true Pareto front, which cannot be obtained in real-world tasks, we use HV as the metric in our benchmark. The reference point required to calculate HV is set to the nadir point, each dimension of which corresponds to the worst value of one objective. Details are provided in Appendix A.

3.2. Dataset Collection

We use three representative MOEAs, i.e., NSGA-II, MOEA/D, and NSGA-III, introduced in Section 2.2 to collect the data for all the tasks. For each problem, we run these three expert algorithms independently and collect the data as our dataset. However, only using the expert algorithms may result in a significant difference between our data distribution and diverse reality distribution. Thus, we introduce a probability of accepting inferior solutions during the survivor selection process of the expert algorithms. In addition, to solve problems with different search spaces, we also employ various types of evolutionary operators. Detailed settings are provided in Appendix A.

The complex objective space of real-world problems presents significant challenges for offline MOO. We provide the visualizations of the objective space in Appendix C. Compared to Design-Bench for offline SOO, our benchmark includes more data due to the inherent challenge of MOO. Additionally, our framework presents many easy-to-use interfaces to facilitate the integration with other algorithm implementations, including sub-problem generation, weight decomposition, HV evaluation, etc.

4. Tasks

In this section, we describe the set of tasks included in our benchmark. An overview of the tasks is provided in Table 1. Each task in our benchmark suite comes with a dataset $\mathcal{D}$, along with a ground-truth oracle objective function $\boldsymbol{f}$ that can be used for evaluation. An offline algorithm should not query the ground-truth oracle function during training, even for hyperparameter tuning. We first discuss the tasks in our benchmark. Detailed information about these tasks are provided in Appendix B due to space limitation.

4.1. Synthetic Function

We first use various synthetic functions as our tasks, which encompass several popular MOO problem sets, i.e., DTLZ (Deb et al., 2002b), ZDT (Zitzler et al., 2000), Omnitest (Deb & Tiwari, 2008), and VLMOP (Van Veldhuizen & Lamont, 1999). The search space is continuous, and the objectives are predetermined by the function designers. Although these synthetic functions may not be considered "realistic", they possess certain advantages and are worth considering for the following reasons: a) Their analytical expressions are known, allowing us to obtain the actual Pareto front for better understanding the problem's characteristic and the algorithm's behavior; b) They can be easily configured to any input dimension and any number of objectives, making them suitable for testing large-scale and many-objective optimization algorithms; c) They are computationally efficient to evaluate, enabling us to collect lots of data and assess the scalability of offline MOO algorithms. We implement these synthetic functions and collect the data.

4.2. Multi-Objective Neural Architecture Search

NAS has paved a promising path towards alleviating the unsustainable process of designing DNN architectures by automating the pipeline. Apart from the prediction error, recent NAS works also consider other objectives, e.g., the number of parameters. These NAS tasks are intrinsically MOO problems, aiming to achieve trade-offs of the multiple design criteria (Lu et al., 2023). We provide a toy example named NAS-Bench-201-Test, which uses a categorical cell-based search space (Dong & Yang, 2020). Besides, C-10/MOP and IN-1K/MOP from Lu et al. (2023) are also included, where both *micro* and *macro* search spaces are

Table 1. Properties of the tasks in offline MOO Benchmark.

Task Name	Dataset size	Dimensions	# Objectives	Search space
Synthetic Function	60000	2-30	2-3	Continuous
MO-NAS	9735-60000	5-34	2-3	Categorical
MO-Swimmer	8571	9734	2	Continuous
MO-Hopper	4500	10184	2	Continuous
MO-TSP	60000	20-500	2-3	Permutation
MO-CVRP	60000	20-100	2-3	Permutation
MO-KP	60000	50-200	2-3	Permutation
MO-Portfolio	60000	20	2	Continuous
Molecule	49001	32	3	Continuous
Regex	42048	4	2	Sequence
RFP	4937	4	2	Sequence
ZINC	48000	4	2	Sequence
Real-world Application	60000	3-6	2-6	Continuous & Mixed

used. For these tasks, there are three objectives to be minimized, i.e., error, number of parameters and edge GPU latency, which measure the model's performance, scale, and the GPU's efficiency during model execution, respectively. The data is from Lu et al. (2023). Detailed information about tasks is provided in Appendix B.2.

4.3. Multi-Objective Reinforcement Learning

Decision making in practical applications usually involves reasoning about multiple, often conflicting, objectives (Zhu et al., 2023). For example, when designing a control policy for a running quadruped robot, we need to consider two conflicting objectives: running speed and energy efficiency. Multi-objective reinforcement learning (MORL) aims to learn agents that can handle such a challenging task. We consider two locomotion tasks in the popular MORL benchmark MuJoCo (Todorov et al., 2012), i.e., MO-Swimmer and MO-Hopper. Their search space is the parameters of an agent, which is much larger than other tasks. The two objectives in MO-Swimmer are speed and energy efficiency, and MO-Hopper considers two objectives related to running and jumping. The data is collected by us via running PG-MORL (Xu et al., 2020).

4.4. Multi-Objective Combinatorial Optimization

Multi-objective combinatorial optimization (MOCO) commonly exists in industries, such as transportation, manufacturing, energy, and telecommunication (Chen et al., 2023b). We consider three typical MOCO problems that are commonly studied, i.e., multi-objective traveling salesman problem (MO-TSP), multi-objective capacitated vehicle routing problem (MO-CVRP), and multi-objective knapsack problem (MO-KP), and a multi-objective portfolio allocation (MO-Portfolio) problem. **MO-TSP** has n nodes, where each node has two sets of 2-dimensional coordinates. There are two objectives, each of which corresponds to the travel cost calculated using one set of 2-dimensional coordinates of all nodes. **MO-CVRP** has n customer nodes and a depot node, with each node featured by a 2-dimensional coordinate and each customer node associated with a demand. Following the common practice, we consider two objectives, i.e., the total tour length and the longest length of the route. **MO-KP** has n items, with each taking a weight and two separate values. The goal is to maximize the sum of their 2-dimensional objective vectors (corresponding to two objectives) under the constraint that the sum of weights does not exceed a capacity. The search space of these problems is a permutation space, and we use the corresponding operator in MOEA to search in it. The **MO-Portfolio** task is continuous and it is based on the Markowitz Mean-Variance Portfolio Theory (Fabozzi et al., 2008), where the two objectives, i.e., expected returns and variance of returns, are used to illustrate the relations between beliefs and choice of portfolio. The data is collected by us.

4.5. Scientific Design

Many real-world scientific problems also involve MOO. We consider molecule design and protein design, which are two important sequence optimization problems. The data of these tasks is collected by us.

Molecule design is critical to pharmaceutical drug discovery (Dara et al., 2022). Previous research has typically required the generated molecules to fulfill several objectives, e.g., new drugs should generally be non-toxic and ideally easy-to-synthesize, in addition to their primary purpose. In this task, we consider two objectives based on prior work in molecular design (Zhao et al., 2022), i.e., activity against biological targets GSK3β and JNK3, respectively. The solution is optimized in a pretrained 32-dimensional continuous latent space (Jin et al., 2020), which is then decoded into

molecular strings and fed into the property evaluators.

Protein design is the process of creating new or improved protein structures for use as biomarkers, therapeutics, etc. We consider the following three tasks. **Regex** is a basic task (around 32 tokens) for protein design, where the objectives are to maximize the counts of multiple bigrams. **ZINC** is a small scale task (around 128 tokens) to optimize the chemical properties of a small molecule. The two objectives are to maximizing the logP (the octanol-water partition coefficient) and QED (quantitative estimate of druglikeness). **RFP** is a large-scale task (around 200 tokens) designed to simulate searching for improved red fluorescent protein (RFP) variants, a problem of significant interest to biomedical researchers. The two objectives are maximizing the solvent-accessible surface area and the stability of RFP, respectively.

4.6. Real-World Application

MOO has applications in many real-world tasks. We select several real-world multi-objective engineering design problems from RE suite (Tanabe & Ishibuchi, 2020), including four bar truss design, pressure vessel design, disc brake design, vehicle crashworthiness design, rocket injector design, etc. These tasks provide various challenges for offline MOO, e.g., they have different number of objectives and different types of variables. We use the evaluation interface from RE and collect the data ourselves.

5. Offline MOO Method

This section introduces the approaches for offline MOO. Though no specific approach has yet been developed to address offline MOO problems, we can adapt the techniques from other related topics, such as offline SOO, MOBO, and surrogate-assisted evolutionary algorithm (SAEA) (Jin et al., 2019). All of these methods use a surrogate model and conduct searches within it. Offline SOO uses a neural network to build a surrogate model, while MOBO typically uses a Gaussian process (GP). SAEA may use both, with a focus on *"How to properly use the surrogate during the iterative search process"*, which, however, is not consistent with offline settings that do not support iterative search. As a result, we consider modifying offline SOO and MOBO methods to address offline MOO tasks, by using the DNN-based and GP-based surrogate models, respectively. We will consider four fundamental components of an offline MOO method: data, model architecture, learning algorithm, and search algorithm, which are shown below.

5.1. DNN-Based Offline MOO Method

DNN-based methods (Yu et al., 2021; Trabucco et al., 2021; 2022; Chen et al., 2023a) have shown impressive performance in offline SOO due to its ability to learn from a large amount of historical data, while also being able to perform search using gradient ascent within it. Model architecture design is a key aspect in this kind of method, especially for offline MOO. We consider the following three models.

End-to-end model is a straightforward approach, using a DNN to learn an approximation of m objectives simultaneously, where the model takes $\boldsymbol{x}$ as input and outputs an m-dimensional objective vector directly.

Multiple models maintains m independent surrogate models for an m-objective problem, which is a common practice in MOBO. Each individual model learns an objective function independently, allowing for the natural use of offline SOO techniques such as COMs (Trabucco et al., 2021), RoMA (Yu et al., 2021), IOM (Qi et al., 2022), ICT (Yuan et al., 2023), and Tri-Mentoring (Chen et al., 2023a).

Multi-head models. We observe that learning multiple objective functions simultaneously is similar to multi-task learning (MTL) (Zhang & Yang, 2022), whose aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks. As a commonly used model in MTL, multi-head models can also serve as a fundamental model for offline MOO. Furthermore, we propose utilizing training techniques (e.g., GradNorm (Chen et al., 2018) and PcGrad (Yu et al., 2020)) from MTL to assist model training of offline MOO.

Data pruning. During the training process, we find that using all the data for model training results in a significant inferior performance: the search algorithm can only obtain few solutions. This phenomenon occurs across all model structures, which may be attributed to the training data quality. Thus, we use data pruning, i.e., selecting some solutions with better scores for training. The corresponding experimental validation will be presented in Section 6.3.

Search algorithm. After training the surrogate model, various methods can be used to obtain the final solution set. We default to using the popular NSGA-II (Deb et al., 2002a) as the search algorithm. Additionally, we also consider employing other MOO algorithms, such as MOEA/D (Zhang & Li, 2007), NSGA-III (Deb & Jain, 2013), and MOBO (Daulton et al., 2021).

5.2. GP-Based Offline MOO Method

GP-based methods, often used in MOBO, are also promising for solving offline MOO problems. However, GP has a much higher computational complexity compared to DNN. Specifically, the computational complexity of learning a GP model is $\mathcal{O}(N^3 + N^2D)$ (Rasmussen & Williams, 2006),

where N is the number of data points and D is the dimension of search space. Thus, directly using GP to offline MOO is not realistic, and data pruning is required. Similar to the data pruning approach in DNN-based methods, we use the non-dominated sorting in NSGA-II (Deb et al., 2002a) to select K data points at the front layers for learning. We will examine the impact of hyper-parameter K on the performance in Figure 4 of Appendix C. We use three mainstream MOBO frameworks for comparison: 1) Hypervolume-based method qNEHVI (Daulton et al., 2021) selects solutions that can maximize the expected improvement in hypervolume, which is our default MOBO. 2) Scalarization-based method qParEGO (Daulton et al., 2020) randomly samples q weight vectors to scalarize the objectives into q single-objective problems and uses expected improvement to select points within each single-objective problem. 3) Information-theoretic-based method JES (Hvarfner et al., 2022) considers the information gain that maximally reduces the uncertainty in both the input and output spaces. On the special discrete tasks, we use Kendall kernel (Deshwal et al., 2022) and transformed overlap kernel (Khan et al., 2023) as the kernel function of GP for permutation and sequence space, respectively. MOBO employs NSGA-II (Deb et al., 2002a) to generate a batch of solutions. MOBO-qParEGO uses a single-objective genetic algorithm with specific operators to optimize the q single-objective problem. MOBO-JES requires a stationary kernel, therefore Kendall kernel (Deshwal et al., 2022) and transformed overlap kernel (Khan et al., 2023) cannot be utilized. Details are provided in Appendix A.4.

6. Experiment

In this section, we empirically examine the performance of different methods on our benchmark. We first introduce the experimental settings, and then report the main results on all the tasks. We also conduct additional experiments to show the challenges of offline MOO, compare training curves, study the effectiveness of data pruning, analyze the volume of data needed for MOBO, and investigate the influence of the search algorithm.

6.1. Experimental Settings

The compared methods are introduced as follows. For NN-based methods, we consider the three types of models discussed in Section 5.1. 1) End-to-End models, including End-to-End, End-to-End + GradNorm (Chen et al., 2018), and End-to-End + PcGrad (Yu et al., 2020). 2) Multi-Head models, including Multi-Head, Multi-Head + GradNorm, and Multi-Head + PcGrad. 3) Multiple models, including Multiple Models, Multiple Models + COMs (Trabucco et al., 2021), Multiple Models + RoMA (Yu et al., 2021), Multiple Models + IOM (Qi et al., 2022), Multiple Models + ICT (Yuan et al., 2023), and Multiple Models + Tri-Mentoring (Chen et al., 2023a). For GP-based methods, we use the three main types, including MOBO (i.e., qNEHVI (Daulton et al., 2021)), MOBO-qParEGO (Knowles, 2006), and MOBO-JES (Hvarfner et al., 2022). All the advanced methods use data pruning by default. After training the model, a search algorithm (which is NSGA-II (Deb et al., 2002a) by default) is run in the model to generate a set of 256 solutions, which are then conducted by one batched evaluation and used to calculate the HV value (i.e., 100th percentile evaluations). The results of other settings, including 256 solutions with 50th percentile evaluations and 32 solutions with 100th percentile evaluations are provided in Appendix C. The model architecture and hyperparameters are consistently maintained across all tasks. Different operators are used for different search spaces, while the operator remains the same within the same search space across all the methods. We report the mean performance and standard deviation over five identical seeds (1000, 2000, ..., 5000) for all algorithms on all the tasks. Note that not all methods can be applied to every task in Off-MOO-Bench due to the long running time and high computational resource cost (for example, running out of GPU memory due to the high complexity), and we indicate this with "N/A". Detailed experimental settings are provided in Appendix A.

6.2. Main Results

Table 2 shows the average rank of all the compared methods on each type of task. Note that $\mathcal{D}(\text{best})$ denotes the best solution set (having the largest HV value) in the training set. Based on the average rank of all tasks (i.e., the last column of the table), we can observe that all the NN-based offline MOO methods outperform the best solution set in the training set, i.e., $\mathcal{D}(\text{best})$, showcasing the feasibility and effectiveness of offline MOO. Multiple Models + IOM is the generally the best method (i.e., winner of Tables 2 and 21, runner-up of Table 14), demonstrating the effectiveness of advanced offline SOO techniques. However, optimizing in a discrete space can indeed be challenging, e.g., no method can achieve a better rank than $\mathcal{D}(\text{best})$ on MOCO, whose search space is a permutation space. This is primarily due to the complexity of modeling the discrete space in the surrogate model, which is also a significant challenge in offline SOO (Kim et al., 2023). Although GP-based methods can be applied to offline MOO by setting the number of iterations to 1 with a large batch size, their performance remains unsatisfactory. MOBO achieves only an average ranking of 8.64, while MOBO-qParEGO and MOBO-JES perform even worse than $\mathcal{D}(\text{best})$. Targeted MOBO algorithms for offline MOO, such as selecting training data more effectively based on the statistics of the offline dataset and designing improved kernel functions, can be proposed based

Table 2. Average rank of different offline MOO methods on each type of task in Off-MOO-Bench, where the best and runner-up ranks are **bolded** and underlined, respectively. Note that $\mathcal{D}(\text{best})$ denotes the best set in the training dataset, and the last column reports the average rank of each method on all the tasks.

Methods	Synthetic	MO-NAS	MORL	MOCO	Sci-Design	RE	Average Rank
$\mathcal{D}$(best)	12.17 ± 0.27	12.11 ± 0.05	9.00 ± 0.50	$\mathbf{2.00 \pm 0.14}$	8.38 ± 0.38	13.13 ± 0.07	10.03 ± 0.07
End-to-End	6.91 ± 0.03	8.37 ± 0.05	7.50 ± 2.00	6.75 ± 0.46	6.75 ± 1.12	7.50 ± 0.57	7.32 ± 0.01
End-to-End + GradNorm	8.25 ± 0.56	7.71 ± 0.08	4.50 ± 1.00	7.61 ± 0.18	8.62 ± 0.50	10.53 ± 0.07	8.34 ± 0.01
End-to-End + PcGrad	7.88 ± 0.06	7.18 ± 0.39	10.50 ± 1.50	6.07 ± 0.64	8.69 ± 2.69	8.23 ± 0.17	7.51 ± 0.14
Multi-Head	6.38 ± 0.50	5.37 ± 0.37	6.25 ± 2.25	8.29 ± 0.21	9.19 ± 0.44	8.33 ± 0.40	7.00 ± 0.38
Multi-Head + GradNorm	7.78 ± 0.53	10.20 ± 0.04	11.00 ± 3.00	9.98 ± 0.30	9.06 ± 1.19	10.63 ± 0.17	9.63 ± 0.04
Multi-Head + PcGrad	8.61 ± 0.14	6.92 ± 0.55	10.50 ± 3.50	8.21 ± 0.36	9.38 ± 0.50	8.50 ± 0.17	8.09 ± 0.20
Multiple Models	$\mathbf{4.05 \pm 0.11}$	4.93 ± 0.28	9.75 ± 0.75	6.34 ± 0.27	5.62 ± 0.75	4.50 ± 0.10	5.02 ± 0.03
Multiple Models + COMs	9.81 ± 0.31	5.92 ± 0.34	7.00 ± 2.00	6.36 ± 0.50	8.38 ± 2.00	10.50 ± 0.50	8.09 ± 0.32
Multiple Models + RoMA	8.95 ± 0.05	5.00 ± 0.00	4.75 ± 2.25	8.14 ± 0.21	8.00 ± 1.38	6.30 ± 0.10	7.07 ± 0.02
Multiple Models + IOM	6.11 ± 0.36	$\mathbf{4.34 \pm 0.34}$	$\mathbf{3.75 \pm 2.75}$	4.25 ± 0.04	7.19 ± 0.44	$\mathbf{3.23 \pm 0.03}$	$\mathbf{4.61 \pm 0.05}$
Multiple Models + ICT	9.11 ± 0.27	11.92 ± 0.29	4.75 ± 0.25	9.89 ± 0.46	8.62 ± 0.75	8.43 ± 0.30	9.64 ± 0.11
Multiple Models + Tri-Mentoring	7.83 ± 0.05	11.37 ± 0.47	5.25 ± 2.75	9.50 ± 0.00	9.38 ± 1.00	6.73 ± 0.20	8.77 ± 0.21
MOBO	9.09 ± 0.47	7.18 ± 0.55	10.50 ± 0.00	13.69 ± 0.08	$\mathbf{5.44 \pm 0.56}$	6.11 ± 0.29	8.64 ± 0.37
MOBO-qParEGO	10.27 ± 0.23	11.47 ± 0.32	N/A	13.62 ± 0.04	9.44 ± 0.44	12.71 ± 0.33	11.68 ± 0.20
MOBO-JES	12.48 ± 0.05	16.00 ± 0.00	N/A	3.00 ± 0.00	7.50 ± 6.50	8.04 ± 0.37	10.30 ± 0.44

on our benchmark, which represents an interesting direction for future research. We can also find that no single method demonstrates a significant advantage, and even the best-performing method only has an average rank of 4.61. These findings indicate that there is still an ongoing challenge to further enhance the effectiveness of offline MOO.

6.3. Additional Results

In this section, we mainly aim to answer the question: What matters to the performance of offline MOO methods? Other results, including the analysis of data pruning, and the analysis on the influence of number of initial points of MOBO and the search algorithms, are provided in Appendix C due to space limitation.

A key challenge of offline MOO is that an inaccurate surrogate model will destroy the final performance. The output of offline MOO is a set of solutions that are non-dominated to each other. If the surrogate model is inaccurate, the Pareto-dominance relationship will be largely influenced. For example, if the model wrongly predicts that one solution is very good, then the solution will dominate all the other solutions, resulting in only few solutions in the final solution set and an extremely low HV value. In our experiments, we have found that the main reason of inaccurate surrogate model lies in the poor performance of learning those solutions with better objective values, i.e., elites. To address this issue, we use data pruning to remove the solutions with worse objective values, allowing the model to focus more on learning from good regions and then obtaining a more accurate model. This will lead to a better final performance, as shown in Figure 2. The left and right columns denote the Multi-Head model without and with data pruning on the task of RE21, respectively. The upper-row shows the search process in the objective space of the surrogate model (i.e., proxy objective space), and the bottom-row shows their mapping in the real oracle objective space. We can observe that the model without data pruning has a phenomenon we discussed before, i.e., there are only two solutions in the final solution set. The model with data pruning performs much better, but still exhibits a certain degree of over-estimation which is also quite common in offline SOO (Trabucco et al., 2021). Thus, finding ways to mitigate such phenomenon is an important future direction in offline MOO. Detailed experiments and discussions about model collapse and data pruning are provided in Appendix C.1.

Learning curves. Based on the above analysis, we have found that the prediction quality of elites has a significant impact on the final performance. To verify this, we compare the vanilla Multi-Head model with the Multi-Head model with GradNorm on two tasks, namely DTLZ1 (from synthetic function) and MO-NAS. Figure 3 shows the changes of the elites loss during the training phase and the visualization of the final solution set in the objective space. It can be clearly observed from the upper subfigures that GradNorm achieves smaller elites loss than vanilla Multi-Head. As a result, the solution set obtained by GradNorm has a generally better distribution, as shown in the bottom subfigures, and also has a better HV value.

7. Discussion

Conclusion. In this paper, we emphasize the significance of offline MOO and provide the first benchmark that encompasses a range of crucial offline MOO tasks, from synthetic functions to real-world applications. Additionally, we introduce a framework of offline MOO methods

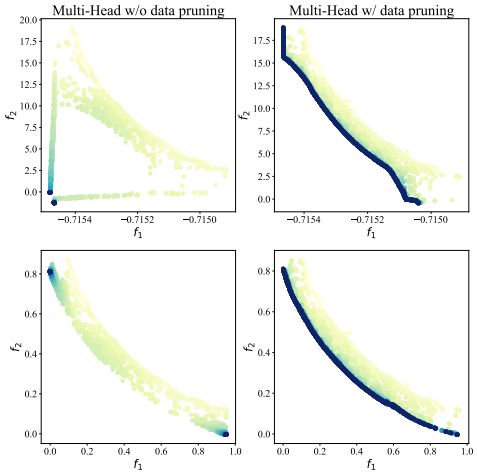

Figure 2. Objective space visualization of Multi-Head model without (left column) and with (right column) data pruning on RE21, where the upper and bottom rows correspond to the surrogate objective space and real objective space, respectively. Each point denotes a solution in the search history, whose color gradually changes from yellow to blue based on the iteration rounds of the search algorithm.

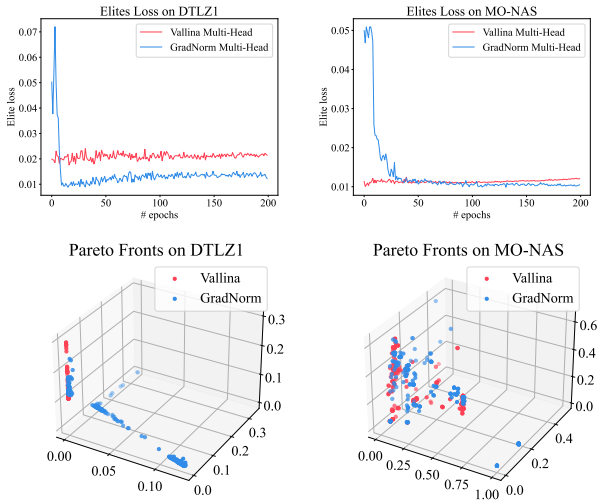

Figure 3. Elites loss changes (upper) and objective space visualizations of the final solution set (bottom), for Vanilla Multi-head model and Multi-head model with GradNorm on the two tasks, DTLZ1 and NAS-Bench-201-Test.

and analyze the different components. Extensive experiments validate the efficacy of these methods. In the future, we will incorporate more analysis (e.g., the influence of reference points (Ishibuchi et al., 2018; 2022)), more demanding tasks (e.g., DNA sequence designs and industrial applications), more learning-based MOO algorithms (e.g., Pareto set learning (Lin et al., 2022; Zhang et al., 2023) and MO-GFlowNets (Jain et al., 2023)), and more advanced offline SOO algorithms (e.g., LEO (Yu et al., 2024), and BRAID (Uehara et al., 2024)) into our benchmark.

Future works of offline MOO. Based on our experimental results and analyses, there are many worthwhile directions for future exploration. Here we discuss some challenges of offline MOO and hope to shed some light on future works.

1. **Mixed search space.** Most search spaces of the current problems are either continuous or discrete. However, in practice, many problems involve mixed variables (Thebelt et al., 2022), which pose significant challenges for offline MOO, especially in constructing accurate surrogate models.

2. **Large-scale (high-dimensional) optimization.** The high-dimensionality of the search space is a common challenge of black-box optimization (Binois & Wycoff, 2022). Our experimental results indicate that the current offline MOO methods do not perform well on large-scale problems, e.g., no method surpasses the best value of the training set on MOCO. Exploring effective techniques such as dimensionality reduction (Wang et al., 2016; Song et al., 2022) to efficiently solve large-scale problems is an important future direction.

3. **Constrained optimization.** Many real-world MOO tasks come with strict constraints (Afshari et al., 2019), making surrogate model learning and search challenging. Our current approach is rather simplistic, which directly discards solutions that do not satisfy the constraints. Employing more efficient constraint handling strategies would significantly improve the performance.

4. **Noisy optimization.** The black-box evaluations of numerous real-world problems involve intricate processes, which often suffer from inaccuracies due to the inevitable presence of noise (Goh & Tan, 2007; Qian et al., 2018). The noise may have a detrimental effect on the quality of the offline dataset, presenting a significant challenge that needs to be addressed.

5. **Few-shot optimization.** Some application scenarios do not have strict limitations on the number of evaluations but allow for a few batches (Wistuba & Grabocka, 2021). How to utilize a limited number of iterative evaluations to adapt the surrogate model is indeed a crucial task for future work.

Impact Statement

This paper presents work whose goal is to advance the field of multi-objective optimization and machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

The authors want to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science and Technology Major Project (2022ZD0116600) and National Science Foundation of China (62276124).

References

Afshari, H., Hare, W., and Tesfamariam, S. Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. *Applied Soft Computing*, 83:105631, 2019.

Arango, S. P., Jomaa, H. S., Wistuba, M., and Grabocka, J. HPO-B: A large-scale reproducible benchmark for black-box HPO based on OpenML. In *Proceedings of 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Bäck, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press., 1996.

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. BoTorch: A Framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, Virtual, 2020.

Bian, C., Zhou, Y., Li, M., and Qian, C. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5513–5521, Macao, SAR, China, 2023.

Binois, M. and Wycoff, N. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

Blank, J. and Deb, K. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8:89497–89509, 2020.

Bosman, P. A. N. and Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Cai, H., Gan, C., Wang, T., Zhang, Z., and Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv:1908.09791*, 2019.

Chemingui, Y., Deshwal, A., Hoang, T. N., and Doppa, J. R. Offline model-based optimization via policy-guided gradient search. In *Proceddings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 11230–11239, Vancouver, Canada, 2024.

Chen, C., Zhang, Y., Fu, J., Liu, X. S., and Coates, M. Bidirectional learning for offline infinite-width model-based optimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, 2022.

Chen, C., Beckham, C., Liu, Z., Liu, X., and Pal, C. Parallel-mentoring for offline model-based optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, New Orleans, LA, 2023a.

Chen, J., Zhang, Z., Cao, Z., Wu, Y., Ma, Y., Ye, T., and Wang, J. Neural multi-objective combinatorial optimization with diversity enhancement. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 39176–39188, New Orleans, LA, 2023b.

Chen, M., Peng, H., Fu, J., and Ling, H. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12270–12280, Virtual, 2021.

Chen, T. and Li, M. The weights can be harmful: Pareto search versus weighted search in multi-objective search-based software engineering. *ACM Transactions on Software Engineering and Methodology*, 32(1):5:1–5:40, 2023.

Chen, Z., Badrinarayanan, V., Lee, C., and Rabinovich, A. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 793–802, Stockholm, Sweden, 2018.

Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. *Evolutionary Algorithms for Solving Multi-objective Problems*. Springer, 2007.

Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M., and Ahsan, M. J. Machine learning in drug discovery: A review. *Artificial Intelligence Review*, 55(3):1947–1999, 2022.

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

Daulton, S., Balandat, M., and Bakshy, E. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 2187–2200, Virtual, 2021.

Daulton, S., Balandat, M., and Bakshy, E. Hypervolume knowledge gradient: A lookahead approach for multi-objective Bayesian optimization with partial information. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 7167–7204, Honolulu, HI, 2023.

Deb, K. *Multi-objective optimization using evolutionary algorithms*. Wiley, 2001.

Deb, K. and Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2013.

Deb, K. and Tiwari, S. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3): 1062–1087, 2008.

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

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, pp. 825–830, 2002b.

Deshwal, A., Belakaria, S., Doppa, J. R., and Kim, D. H. Bayesian optimization over permutation spaces. In *Proceedings of 36th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6515–6523, Virtual, 2022.

Dong, X. and Yang, Y. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

Dong, X., Liu, L., Musial, K., and Gabrys, B. NATS-Bench: Benchmarking NAS algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3634–3646, 2021.

Ehrgott, M. *Multicriteria Optimization*. Springer, 2005.

Emmerich, M. T., Giannakoglou, K. C., and Naujoks, B. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

Fabozzi, F. J., Markowitz, H. M., and Gupta, F. Portfolio selection. *Handbook of Finance*, 7(1):77, 2008.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv:2004.07219*, 2020.

Goh, C. K. and Tan, K. C. An investigation on noisy environments in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 11(3): 354–381, 2007.

Hernández-Lobato, D., Hernandez-Lobato, J., Shah, A., and Adams, R. Predictive entropy search for multi-objective Bayesian optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1492–1501, New York City, NY, 2016.

Hvarfner, C., Hutter, F., and Nardi, L. Joint entropy search for maximally-informed Bayesian optimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 11494–11506, New Orleans, LA, 2022.

Ishibuchi, H., Akedo, N., and Nojima, Y. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, 2014.

Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation*, 26:411–440, 2018.

Ishibuchi, H., Pang, L. M., and Shang, K. Difficulties in fair performance comparison of multi-objective evolutionary algorithms. *IEEE Computational Intelligence Magazine*, 17(1):86–101, 2022.

Jain, M., Raparthy, S. C., Hernández-García, A., Rector-Brooks, J., Bengio, Y., Miret, S., and Bengio, E. Multi-Objective GFlowNets. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 14631–14653, Honolulu, HI, 2023.

Jin, W., Barzilay, R., and Jaakkola, T. S. Hierarchical generation of molecular graphs using structural motifs. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 4839–4848, Virtual, 2020.

Jin, Y., Wang, H., Chugh, T., Guo, D., and Miettinen, K. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458, 2019.

Khan, A., Cowen-Rivers, A. I., Grosnit, A., Robert, P. A., Greiff, V., Smorodina, E., Rawat, P., Akbar, R., Dreczkowski, K., Tutunov, R., et al. Toward real-world automated antibody design with combinatorial Bayesian optimization. *Cell Reports Methods*, 3(1), 2023.

Kim, M., Berto, F., Ahn, S., and Park, J. Bootstrapped training of score-conditioned generator for offline design of biological sequences. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 67643–67661, New Orleans, LA, 2023.

Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Konakovic Lukovic, M., Tian, Y., and Matusik, W. Diversity-guided multi-objective Bayesian optimization with batch evaluations. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 17708–17720, Virtual, 2020.

Krishnamoorthy, S., Mashkaria, S. M., and Grover, A. Diffusion models for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 17842–17857, Honolulu, HI, 2023.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

Kumar, A. and Levine, S. Model inversion networks for model-based optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, Virtual, 2020.

Li, C., Yu, Z., Fu, Y., Zhang, Y., Zhao, Y., You, H., Yu, Q., Wang, Y., Hao, C., and Lin, Y. HW-NAS-Bench: Hardware-aware neural architecture search benchmark. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual, 2021.

Lin, X., Yang, Z., Zhang, X., and Zhang, Q. Pareto set learning for expensive multi-objective optimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, 2022.

Lu, Z., Cheng, R., Jin, Y., Tan, K. C., and Deb, K. Neural architecture search as multiobjective optimization benchmarks: Problem formulation and performance assessment. *IEEE Transactions on Evolutionary Computation*, 28:328–337, 2023.

Lust, T. and Teghem, J. The multiobjective traveling salesman problem: A survey and a new approach. In *Advances in Multi-Objective Nature Inspired Computing*, pp. 119–141. Springer, 2010.

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

Paria, B., Kandasamy, K., and Póczos, B. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 766–776, Toronto, Canada, 2020.

Qi, H., Su, Y., Kumar, A., and Levine, S. Data-driven offline decision-making via invariant representation learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 13226–13237, New Orleans, LA, 2022.

Qian, C., Yu, Y., and Zhou, Z. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 204:99–119, 2013.

Qian, C., Yu, Y., and Zhou, Z. Analyzing evolutionary optimization in noisy environments. *Evolutionary Computation*, 26(1), 2018.

Qin, R., Zhang, X., Gao, S., Chen, X., Li, Z., Zhang, W., and Yu, Y. NeoRL: A near real-world benchmark for offline reinforcement learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, 2022.

Qing, J., Moss, H. B., Dhaene, T., and Couckuyt, I. PF^{2}ES: Parallel feasible Pareto frontier entropy search for multi-objective Bayesian optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2565–2588, Valencia, Spain, 2023.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

Song, L., Xue, K., Huang, X., and Qian, C. Monte Carlo tree search based variable selection for high dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, 2022.

Stanton, S., Maddox, W. J., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 20459–20478, Baltimore, MD, 2022.

Suzuki, S., Takeno, S., Tamura, T., Shitara, K., and Karasuyama, M. Multi-objective Bayesian optimization using Pareto-frontier entropy. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 9279–9288, Virtual, 2020.

Tanabe, R. and Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Thebelt, A., Tsay, C., Lee, R. M., Sudermann-Merx, N., Walz, D., Shafei, B., and Misener, R. Tree ensemble kernels for Bayesian optimization with known constraints over mixed-feature spaces. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, 2022.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, Vilamoura, Portugal, 2012.

Trabucco, B., Kumar, A., Geng, X., and Levine, S. Conservative objective models for effective offline model-based optimization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 10358–10368, Virtual, 2021.

Trabucco, B., Geng, X., Kumar, A., and Levine, S. Design-Bench: Benchmarks for data-driven offline model-based optimization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 21658–21676, Baltimore, MD, 2022.

Uehara, M., Zhao, Y., Hajiramezanali, E., Scalia, G., Eraslan, G., Lal, A., Levine, S., and Biancalani, T. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. *arXiv:2405.19673*, 2024.

Van Veldhuizen, D. A. and Lamont, G. B. Multiobjective evolutionary algorithm test suites. In *Proceedings of the 1999 ACM Symposium on Applied Computing*, pp. 351–357, 1999.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55(1):361–387, 2016.

Wistuba, M. and Grabocka, J. Few-shot Bayesian optimization with deep kernel surrogates. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual, 2021.

Xu, J., Tian, Y., Ma, P., Rus, D., Sueda, S., and Matusik, W. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 10607–10616, Virtual, 2020.

Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., and Hutter, F. NAS-Bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 7105–7114, Long Beach, CA, 2019.

Yu, P., Zhang, D., He, H., Ma, X., Miao, R., Lu, Y., Zhang, Y., Kong, D., Gao, R., Xie, J., et al. Latent energy-based odyssey: Black-box optimization via expanded exploration in the energy-based latent space. *arXiv:2405.16730*, 2024.

Yu, S., Ahn, S., Song, L., and Shin, J. RoMA: Robust model adaptation for offline model-based optimization. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 4619–4631, Virtual, 2021.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, Virtual, 2020.

Yuan, Y., Chen, C., Liu, Z., Neiswanger, W., and Liu, X. Importance-aware co-teaching for offline model-based optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, New Orleans, LA, 2023.

Zajac, S. and Huber, S. Objectives and methods in multi-objective routing problems: A survey and classification scheme. *European Journal of Operational Research*, 290 (1):1–25, 2021.

Zela, A., Siems, J., Zimmer, L., Lukasik, J., Keuper, M., and Hutter, F. Surrogate NAS benchmarks: Going beyond the limited search spaces of tabular NAS benchmarks. *arXiv:2008.09777*, 2020.

Zhang, Q. and Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

Zhang, Q., Liu, W., Tsang, E., and Virginas, B. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2009.

Zhang, R. and Golovin, D. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 11096–11105, Virtual, 2020.

Zhang, X., Lin, X., Xue, B., Chen, Y., and Zhang, Q. Hypervolume maximization: A geometric view of Pareto set learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, New Orleans, LA, 2023.

Zhang, Y. and Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022.

Zhao, Y., Wang, L., Yang, K., Zhang, T., Guo, T., and Tian, Y. Multi-objective optimization by learning space partitions. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Virtual, 2022.

Zhou, Z.-H., Yu, Y., and Qian, C. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer, 2019.

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

Zitzler, E. and Thiele, L. Multiobjective optimization using evolutionary algorithms: A comparative case study. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 292–304, Amsterdam, The Netherlands, 1998.

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

A. Detailed settings

In this section, we provide detailed settings of Off-MOO-Bench regarding data collection, training set formulation, and default settings of offline MOO methods.

A.1. Dataset Collection

As discussed in Section 3.2, only using the expert MOEAs, i.e., NSGA-II, MOEA/D, and NSGA-III, may result in obtaining solutions with good quality, which generates a significant difference between data distribution and diverse reality distribution. Inspired by (Zhu et al., 2023), we propose an amateur survival operator for MOEAs. We first use the expert MOEAs to perform generic mating and generate offspring. Assume that the current population has μ individuals and the offspring population has k individuals. With a given probability p, we choose the μ among $(\mu + k)$ individuals according to non-dominated sorting of NSGA-II (Deb et al., 2002a) to form the next population. Otherwise, with probability $1 - p$, we survive the μ best-non-dominated individuals, as the Survival operator in NSGA-II. After a small amount of number of generations (e.g. 1 or 5), we collect the current population to form the final dataset. Typically, we use NSGA-II with the amateur survival operator as the amateur collection algorithm for all tasks of synthetic functions, MOCO, scientific design, and most tasks of RE, except for RE34, where we use NSGA-III due to its better performance in obtaining dataset with diversity.

For NAS-Bench 201 (Dong & Yang, 2020) and NARTS (Dong et al., 2021) in MO-NAS, since the size of their search space is limited (i.e., 15625 for NAS-Bench 201 and 32678 for NARTS), we directly iterate the whole search space and collect query-answers of MO-NAS-Bench (Lu et al., 2023). For MORL tasks, since our goal is to learn directly from policies to rewards, the algorithm for data collection proposed by D4MORL (Zhu et al., 2023) is not suitable for us. Thus, we run the SOTA MORL algorithm PGMORL (Xu et al., 2020) with 100 different seeds and collect the policies. For the tasks of scientific design, we use the Amateur-NSGA-II (as discussed above) to collect part of the dataset, and randomly sample over the whole search space to form the other part.

A.2. Training Set Construction

In realistic scientific and industrial scenarios, we usually hope to use offline collected dataset to obtain better designs than offline ones. Thus, similar to (Trabucco et al., 2022), we remove the top solutions sorted by NSGA-II ranking with a given percentile K, where K varies according to different tasks and is usually set 40%, except for Molecule with 1.2%, RFP and Regex with 20%, and MO-CVRP with 55%. Besides, we perform normalization within each objective of each problem because different objectives can have different scales, which may result in imbalanced model update.

A.3. Training Details

For NN-based model, similar to Design-Bench (Trabucco et al., 2022), the End-to-End network structure is:

$$\text{input} \to \textbf{MLP}(2048) \to \textbf{relu} \to \textbf{MLP}(2048) \to \textbf{relu} \to \textbf{MLP}(\text{number of objectives}).$$

The Multi-Head model is constructed by two parts of neural networks, feature extractor and task head. For feature extractor, the structure is:

$$\text{input} \to \textbf{MLP}(2048) \to \textbf{relu} \to \textbf{MLP}(2048).$$

For task head, the structure is:

$$\text{features with 2048 dimensions} \to \textbf{relu} \to \textbf{MLP}(1).$$

The network structure of multiple models is

$$\text{input} \to \textbf{MLP}(2048) \to \textbf{relu} \to \textbf{MLP}(2048) \to \textbf{relu} \to \textbf{MLP}(1).$$

We use MSE as loss function and optimize by Adam with learning rate $\eta = 0.001$ and learning-rate decay $\gamma = 0.98$. The DNN model is trained w.r.t. offline dataset for 200 epochs with a batch size of 32.

For the End-to-End + GradNorm method, we perform gradient normalization on the last MLP layer. For the Multi-Head + GradNorm method, we perform gradient normalization on the last MLP layer of the feature extractor, as in MTL (Chen et al., 2018).

For GP-based methods, we choose the top 100 solutions by NSGA-II ranking to initialize the GP model. Since the performance of a GP model is sensitive to the number of initialized points, we conduct ablation studies in Figure 4.

Among all tasks in Off-MOO-Bench, Molecule and MOCO have constraints. For Molecule, since we optimize in the latent space following Zhao et al. (2022), we cannot judge if a solution is feasible in the latent space. Thus, we first obtain a batch of 256 solutions generated by the algorithm and then filter out the infeasible ones during evaluation. For MOCO tasks, since we use the Start-From-Zero repair operator, the constraint is avoided.

A.4. MOO Settings

For DNN-based surrogate models, after the model is trained, we use multi-objective evolutionary algorithms (MOEAs) to optimize inside the trained model. To obtain K (approximately) Pareto-optimal solutions, we set the size of population to K and initialize the population with K non-dominated solutions in the offline dataset, and the algorithm searches for 50 generations. We use different genetic operators for different types of tasks. Specifically, for continuous tasks (i.e., synthetic functions, RE, NAS-Bench-201-Test, MO-Portfolio, MORL, and Molecule), we use the default genetic operators of NSGA-II implemented in PyMOO (Blank & Deb, 2020), i.e., Simulated Binary Crossover (SBX) and Polynomial Mutation (PM). For discrete tasks in MOCO (i.e., MO-TSP, MO-CVRP, and MO-KP), since the search space is combinatorial, where each solution in the three problems can be represented as a permutation, we use Order-Crossover as the crossover operator, Inversion-Mutation as the mutation operator, and for MO-TSP and MO-CVRP problems, we utilize the Start-From-Zero repair operator to make sure that the salesman starts from the deposit. For C-10/MOP and IN-1K/MOP test suites in MO-NAS, we use the suggested genetic operators from the source code of Lu et al. (2023), i.e., PM for integer with $\eta = 20$ and SBX for integer with $\eta = 30$. For Regex, ZINC, and RFP tasks, we use the local mutation operator implemented by LaMBO (Stanton et al., 2022), and SBX for integer as in Stanton et al. (2022).

For GP-based surrogate models, we use different methods to optimize the acquisition function for different types of tasks. Specifically, for continuous tasks (i.e., synthetic functions, RE, NAS-Bench-201-Test, MO-Portfolio, MORL, and Molecule), we use gradient-based methods (i.e., L-BFGS-B (Byrd et al., 1995)) to optimize the acquisition function, which is the default acquisition function optimization method implemented in BoTorch (Balandat et al., 2020). For discrete tasks, we use MOEAs to optimize the acquisition function. Our default MOBO employs NSGA-II (Deb et al., 2002a) to generate a batch of solutions that minimize the lower confidence bound of GP, where we set the size of population to K and initialize the population with K non-dominated solutions in the offline dataset, and the algorithm searches for 500 generations with SBX crossover and PM mutation to obtain the final solutions. For MOBO-qParEGO, we use single-objective evolutionary algorithms to optimize the acquisition function. Specifically, we first initialize the population with 50 randomly sampled points, and then search for 500 generations to obtain the best solution for each scalarized single-objective problem. The genetic operators for discrete spaces are as same as the ones in evolutionary search algorithms inside DNN-based surrogate models. Note that MOBO-JES cannot run in discrete tasks, since it requires a stationary kernel and thus Kendall kernel and transformed overlap kernels cannot be utilized.

The implementations of NSGA-II, MOEA/D, and NSGA-III are from the open-source repository PyMOO (Blank & Deb, 2020). The implementation of MOBO is inherited from BoTorch (Balandat et al., 2020).

B. Detailed Tasks

In this section, we provide details of different MOO tasks adopted in our experiments. Notably, certain maximization tasks undergo transformation into minimization problems through the multiplication of -1. The reference points $\boldsymbol{r}$ for majority of tasks are set in such a way that $(r_i - z^i_{\min})/(z^i_{\max} - z^i_{\min}) = 1.1$, except that for MORL tasks, $(r_i - z^i_{\min})/(z^i_{\max} - z^i_{\min}) = 2.0$, where r_i denotes the value on the i-th dimension of the reference point $\boldsymbol{r}$, and $z^i_{\max}$ and $z^i_{\min}$ are the maximum value and minimum value of the i-th objective in the collected data, respectively. It means that after normalization, the reference point becomes $(1.1, \ldots, 1.1)$ or $(2.0, \ldots, 2.0)$.

B.1. Synthetic Function

Various widely-used synthetic functions in MOO literature are employed to evaluate the algorithms. Specifically, the following benchmark problems are used: DTLZ1-7 (Deb et al., 2002b), ZDT1-4, ZDT6 (Zitzler et al., 2000) , Omni-test (Deb & Tiwari, 2008) and VLMOP1-3 (Van Veldhuizen & Lamont, 1999). The solution spaces for all synthetic problems are continuous. The detailed problem information, Pareto front shape and reference point can be found in Table 3. Note

that the concave (2d) Pareto front for DTLZ5 and DTLZ6 indicates that the Pareto front takes the form of a degenerated 2-dimensional curve within a 3-dimensional objective space.

Table 3. Problem information and reference point for synthetic functions.

Name	D	m	Type	Pareto Front Shape	Reference Point
DTLZ1	7	3	Continuous	Linear	(558.21, 552.30, 568.36)
DTLZ2	10	3	Continuous	Concave	(2.77, 2.78, 2.93)
DTLZ3	10	3	Continuous	Concave	(1703.72, 1605.54, 1670.48)
DTLZ4	10	3	Continuous	Concave	(3.03, 2.83, 2.78)
DTLZ5	10	3	Continuous	Concave (2d)	(2.65, 2.61, 2.70)
DTLZ6	10	3	Continuous	Concave (2d)	(9.80, 9.78, 9.78)
DTLZ7	10	3	Continuous	Disconnected	(1.10, 1.10, 33.43)
ZDT1	30	2	Continuous	Convex	(1.10, 8.58)
ZDT2	30	2	Continuous	Concave	(1.10, 9.59)
ZDT3	30	2	Continuous	Disconnected	(1.10, 8.74)
ZDT4	10	2	Continuous	Convex	(1.10, 300.42)
ZDT6	10	2	Continuous	Concave	(1.07, 10.27)
Omnitest	2	2	Continuous	Convex	(2.40, 2.40)
VLMOP1	1	2	Continuous	Concave	(4.0, 4.0)
VLMOP2	6	2	Continuous	Concave	(1.10, 1.10)
VLMOP3	2	3	Continuous	Disconnected	(9.07, 66.62, 0.23)

B.2. MO-NAS

MO-NAS (Lu et al., 2023) automates the exploration of optimal neural network architectures to enhance multiple model metrics for specific tasks. In our experiments, we conduct a toy example, named NAS-Bench-201-Test, to optimize three objectives: prediction error, number of parameters, and edge GPU latency, on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009). The prediction error metric primarily assesses the model's performance, the number of parameters gauges the model's scale, and the GPU latency is a hardware metric evaluating the efficiency of GPU during model execution. The search space is from Dong & Yang (2020). Given a macro skeleton of the neural network and a directed acyclic graph structure for each cell, our objective is to explore the operations (edges) within the cell. Each cell contains 6 edges, and there are 5 predefined operation options for each edge (zeroize, skip-connect, 1×1 convolution, 3×3 convolution, and 3×3 average pool). Consequently, the search space is a 6-dimensional discrete space, where each dimension can take 5 values, resulting in a total of $5^6 = 15625$ possible solutions. The data of NAS-Bench-201-Test, corresponding error and number of parameters are sourced from Dong & Yang (2020). Additionally, the edge GPU latency data is obtained from Li et al. (2021). The reference point is $\boldsymbol{r} = (98.48, 1.68, 12.81)$.

Furthermore, we consider two test suites from Lu et al. (2023), i.e., C-10/MOP and IN-1K/MOP, which contain 18 tasks. The search spaces of these test suites vary from *micro* search spaces to *macro* search spaces. Specifically, *micro* search spaces are used to create a basic building block, often called a *cell*, which is used repeatedly to build a full deep neural network (DNN) based on a set pattern; *macro* search spaces are used to design the overall structure of the network, while the individual layers are designed using well-established methods. *Micro* search spaces include NAS-Bench-101 (Ying et al., 2019), NAS-Bench-201 (Dong & Yang, 2020), and DARTS (Zela et al., 2020). *Macro* search spaces include NATS (Dong et al., 2021), ResNet50 (Cai et al., 2019), Transformer (Chen et al., 2021), and MNV3 (Cai et al., 2019). Detailed information of these search spaces $\mathcal{X}$ can be found in Table 4. The detailed problem information and reference point of C-10/MOP1-9 and IN-1K/MOP1-9 tasks can be found in Table 5. Note that we transform the discrete search space of NAS-Bench-201-Test into a continuous logit space, which is the strategy in Design-Bench (Trabucco et al., 2022) for handling discrete categorical tasks. However, it cannot be applied to all tasks in MO-NAS, since it requires that all dimensions have the same number of categories, while the number of categories in most tasks of MO-NAS differ from dimensions.

Table 4. An overview of the search spaces in MO-NAS tasks.

Search space $\mathcal{X}$	Type	D	$\lvert\mathcal{X}\rvert$
NAS-Bench-101	micro	26	423K
NAS-Bench-201	micro	6	15.6K
NATS	macro	5	32.8K
DARTS	micro	32	$\sim 10^{21}$
ResNet50	macro	25	$\sim 10^{14}$
Transformer	macro	34	$\sim 10^{14}$
MNV3	macro	21	$\sim 10^{20}$

Table 5. Problem information and reference point for C-10/MOP1-9 and IN-1K/MOP1-9 tasks.

Problem	Search space $\mathcal{X}$	D	m	Reference Point
C-10/MOP1	NAS-Bench-101	26	2	$(3.49 \times 10^{-1}, 3.14 \times 10^{7})$
C-10/MOP2	NAS-Bench-101	26	3	$(9.05 \times 10^{-1}, 3.05 \times 10^{7}, 8.97 \times 10^{9})$
C-10/MOP3	NATS	5	3	$(2.31 \times 10^{1}, 7.14 \times 10^{-1}, 2.74 \times 10^{2})$
C-10/MOP4	NATS	5	4	$(2.31 \times 10^{1}, 7.14 \times 10^{-1}, 2.74 \times 10^{2}, 2.12 \times 10^{-2})$
C-10/MOP5	NAS-Bench-201	6	5	$(9.03 \times 10^{1}, 1.53 \times 10^{0}, 2.20 \times 10^{2}, 1.17 \times 10^{1}, 4.88 \times 10^{1})$
C-10/MOP6	NAS-Bench-201	6	6	$(9.03 \times 10^{1}, 1.53 \times 10^{0}, 2.20 \times 10^{2}, 1.05 \times 10^{1}, 2.23 \times 10^{0}, 2.76 \times 10^{1})$
C-10/MOP7	NAS-Bench-201	6	8	$(9.03 \times 10^{1}, 1.53 \times 10^{0}, 2.20 \times 10^{2}, 1.17 \times 10^{1},$ $4.88 \times 10^{1}, 1.05 \times 10^{1}, 2.23 \times 10^{0}, 2.76 \times 10^{1})$
C-10/MOP8	DARTS	32	2	$(2.61 \times 10^{-1}, 1.55 \times 10^{6})$
C-10/MOP9	DARTS	32	3	$(4.85 \times 10^{-2}, 3.92 \times 10^{5})$
IN-1K/MOP1	ResNet50	25	2	$(2.81 \times 10^{-1}, 3.95 \times 10^{7})$
IN-1K/MOP2	ResNet50	25	2	$(2.80 \times 10^{-1}, 1.15 \times 10^{10})$
IN-1K/MOP3	ResNet50	25	3	$(2.81 \times 10^{-1}, 3.87 \times 10^{7}, 1.26 \times 10^{10})$
IN-1K/MOP4	Transformer	34	2	$(1.83 \times 10^{1}, 7.25 \times 10^{7})$
IN-1K/MOP5	Transformer	34	2	$(1.83 \times 10^{1}, 1.49 \times 10^{10})$
IN-1K/MOP6	Transformer	34	3	$(1.83 \times 10^{1}, 7.10 \times 10^{7}, 1.48 \times 10^{10})$
IN-1K/MOP7	MNV3	21	2	$(2.64 \times 10^{-1}, 9.98 \times 10^{6})$
IN-1K/MOP8	MNV3	21	3	$(2.65 \times 10^{-1}, 1.00 \times 10^{7}, 1.34 \times 10^{9})$
IN-1K/MOP9	MNV3	21	4	$(2.65 \times 10^{-1}, 1.03 \times 10^{7}, 1.31 \times 10^{9}, 6.30 \times 10^{1})$

B.3. MORL

MORL is an approach where the training of an agent focuses on simultaneously maximizing multiple cumulative rewards in some control environment. The primary purpose of proposing the MORL problem in our benchmark is to examine the performance of offline MOO in high-dimensional continuous spaces. Different from the D4MORL benchmark (Zhu et al., 2023), we focus on direct policy parameter search, ignoring some properties of MDP. Note that using numerous neural network parameters as a search space for black-box optimization presents a significant optimization challenge, which is profoundly significant for offline MOO itself. In our experiments, we consider two locomotion tasks namely MO-Swimmer and MO-Hopper, within the widely used MuJoCo benchmark (Todorov et al., 2012). The search space consists of the parameters of the policy network for each environment as defined in (Xu et al., 2020), whose dimension is much higher than other tasks.

MO-Swimmer. This is a two-objective task with an eight-dimensional state space and a two-dimensional action space. The two objectives are forward speed and energy efficiency, denoted as $\boldsymbol{R} = [R^s, R^e]$. The search space is the 9734-dimensional policy network for MO-Swimmer. At time t, the agent is at position (x_t, y_t) and takes an action a_t. Then, the instantaneous rewards at time t are defined as:

$$R_t^s = (x_t - x_{t-1})/0.05,$$
$$R_t^e = 0.3 - 0.15 \times \sum_k a_k^2.$$

The reference point $\boldsymbol{r} = (267.67, 99.05)$ after multiplying -1.

MO-Hopper. This is a two-objective task with an eleven-dimensional state space and a three-dimensional action space. The two objectives are forward speed and jumping height, denoted as $\boldsymbol{R} = [R^s, R^j]$. The search space is the 10184-dimensional policy network for MO-Hopper. At time t, the agent is at position (x_t, h_t) and takes an action a_t. Then, the instantaneous rewards at time t are defined as:

$$R_t^s = 1.5 \times (x_t - x_{t-1})/0.01 + 1 - 2 \times 10^{-4} \sum_k a_k^2,$$
$$R_t^j = 12 \times (h_t - h_0)/0.01 + 1 - 2 \times 10^{-4} \sum_k a_k^2,$$

where $h_0 = 1.25$ is the initial height. The reference points $\boldsymbol{r} = (1489.01, 4734.48)$ after multiplying -1.

B.4. MOCO

We evaluate the algorithms on three typical discrete MOCO problems, i.e., the multi-objective traveling salesman problem (MO-TSP) (Lust & Teghem, 2010), multi-objective capacitated vehicle routing problem (MO-CVRP) (Zajac & Huber, 2021) and multi-objective knapsack problem (MO-KP) (Ishibuchi et al., 2014), and one continuous MOCO problem, i.e., multi-objective portfolio problem (MO-Portfolio). The search spaces for the three discrete problems are formulated as permutation spaces, where the parameters of problem instance are randomly generated similar to (Chen et al., 2023b). Additional, for the MO-TSP problem, we also consider its tri-objective variant, as in (Chen et al., 2023b). The MO-Portfolio problem has a continuous search space, i.e., $[0, 1]^n$, to represent the weights of portfolio allocation. Historical stock prices data of each portfolio is provided by Blank & Deb (2020).

MO-TSP has $n = 500, 100, 50, 20$ nodes, and each node has two sets of two-dimensional coordinates, where the i-th objective value of the solution is calculated with respect to the i-th set of coordinates. The coordinates are generated uniformly from $[0, 1]^2$. Hence, this is a n-dimensional two-objective permutation optimization problem. The reference point is $\boldsymbol{r} = (255.18, 248.44)$.

MO-CVRP has $n = 100, 50, 20$ customer nodes and a depot node, with each node featured by a two-dimensional coordinate and each customer node associated with a demand. Following the common practice, we consider two objectives, i.e., the total tour length and the longest length of the route. The coordinates and demands are generated uniformly from $[0, 1]^2$ and $\{0, \ldots, 9\}$, respectively. The capacity of vehicle is set to 50. Each solution is represented as a n-dimensional permutation. For the evaluation of each solution (permutation), a vehicle departs from the depot and travels in the order specified by the permutation of customers. It accumulates customer capacity along the path, returning to the depot before reaching a

customer in the permutation whose capacity exceeds the limit. The vehicle then continues from the depot, following the point after the last visited customer in the permutation. This process continues until completion. This is a n-dimensional two-objective permutation optimization problem. The reference point is $\boldsymbol{r} = (49.19, 9.58)$.

MO-KP has $n = 200, 100, 50$ items, with each taking a weight and two separate values. The i-th objective is to maximize the sum of the i-th values under the constraint of not exceeding the knapsack capacity. The weight and value of each item are generated uniformly from $[0, 1]$. The capacity is set to 25. Each solution is represented as a 200-dimensional permutation. For the evaluation of each solution (permutation), we put the first k items in the permutation into the knapsack, such that including the $(k + 1)$-th item exceeds the knapsack capacity, while the first k items remain within the capacity limit. This is a n-dimensional two-objective permutation optimization problem. The reference point is $\boldsymbol{r} = (-7.85, -8.99)$ after multiplying -1.

MO-Portfolio has $n = 20$ types of portfolios, with each taking an input as its corresponding weight. Here we consider the portfolio allocation problem based on the Markowitz Mean-Variance Portfolio Theory (Fabozzi et al., 2008) with two objectives, where the overall performance of a portfolio can be assessed through the expected return and overall risk of its assets. Geometrically, the expected return of a portfolio is defined as the average return of its assets and the risk is defined as the standard deviation. Additionally, in order to ensure that portfolio allocations are valid, we provide a repair operator that modifies the portfolio weights to ensure that they sum to 1 (as a common constraint in portfolio optimization) and no weights are smaller than a threshold $\theta = 0.001$. The reference point is $\boldsymbol{r} = (0.29, -0.13)$.

B.5. Scientific Design

Molecule design. This is a two-objective molecular generation task (Zhao et al., 2022). The task is to optimize the activity against biological targets GSK3β and JNK3. The search space is a 32-dimensional continuous latent space. The solutions in the latent space will be decoded into molecular strings and evaluated by a pre-trained decoder from Jin et al. (2020). The reference point $\boldsymbol{r} = (0.09, 0.04)$.

Protein design. We have incorporated two protein sequence design challenges outlined in (Stanton et al., 2022). The sequence optimization task starts with a base sequence pool P of initial sequences, which are modified to produce new candidate sequences. The optimization problem is restructured into the following nested decisions: 1) Choose a base sequence from the pool; 2) Choose which positions on the sequence to change; 3) Choose the operations to change the token at those positions; 4) If the operation is substitution or insertion, then select the tokens to substitute or insert. Hence, the search space can be formalized as a four-dimensional space, encompassing choices for the base sequence, sequence positions, operations, and tokens.

For **Regex**, there are 16 base sequences, 73 sequence positions, 20 types of tokens, and 3 operations (substitution, deletion or insertion). So the search space has a size of $|\mathcal{X}| = 16 \times 72 \times 20 \times 3 = 69,120$. The goal is to maximize the counts of three predetermined bigrams. The reference point $\boldsymbol{r} = (1.11, 1.25, 1.21)$.

For **ZINC**, there are 16 base sequences, 257 sequence positions, 106 types of tokens, and 3 operations (substitution, deletion or insertion). So the search space has a size of $|\mathcal{X}| = 16 \times 257 \times 106 \times 3 = 1,307,616$. The goal is to maximize the the octanol-water partition coefficient (logP) and QED (quantitative estimate of druglikeness). The reference point $\boldsymbol{r} = (1.36, 2.25)$.

For **RFP**, there are 43 base sequences, 489 sequence positions, 20 types of tokens, and 1 operations (substitution). So the search space has a size of $|\mathcal{X}| = 43 \times 489 \times 20 \times 1 = 420,540$. The goal is to maximize the solvent-accessible surface area (SASA) and the stability of the RFP. The reference point $\boldsymbol{r} = (4.80, 4.54)$.

B.6. RE

We also conduct experiments on seven real-world multi-objective engineering design problems adopted from RE suite (Tanabe & Ishibuchi, 2020). These problems serve as practical application in various fields. The search spaces for the problems are continuous except for RE23, which has a mixed solution space (2 variables as integers and 2 as continuous values). The detailed problem information and reference points can be found in Table 6.

Table 6. Problem information and reference point for RE problems.

Name	D	m	Type	Pareto Front Shape	Reference Point
RE21 (Four bar truss design)	4	2	Continuous	Convex	(3144.44, 0.05)
RE22 (Reinforced concrete beam design)	3	2	Mixed	Mixed	(829.08, 2407217.25)
RE23 (Pressure vessel design)	4	2	Mixed	Mixed, Disconnected	(713710.88, 1288669.78)
RE24 (Hatch cover design)	2	2	Continuous	Convex	(5997.83, 43.67)
RE25 (Coil compression spring design)	3	2	Mixed	Mixed, Disconnected	(124.79, 10038735.00)
RE31 (Two bar truss design)	3	3	Continuous	Unknown	(808.85, 6893375.82, 6793450.00)
RE32 (Welded beam design)	4	3	Continuous	Unknown	(290.66, 16552.46, 388265024.00)
RE33 (Disc brake design)	4	3	Continuous	Unknown	(8.01, 8.84, 2343.30)
RE34 (Vehicle crashworthiness design)	5	3	Continuous	Unknown	(1702.52, 11.68, 0.26)
RE35 (Speed reducer design)	7	3	Mixed	Unknown	(7050.79, 1696.67, 397.83)
RE36 (Gear train design)	4	3	Integer	Concave, Disconnected	(10.21, 60.00 , 0.97)
RE37 (Rocket injector design)	4	3	Continuous	Unknown	(0.99, 0.96, 0.99)
RE41 (Car side impact design)	7	4	Continuous	Unknown	(42.65, 4.43, 13.08, 13.45)
RE42 (Conceptual marine design)	6	4	Continuous	Unknown	(-26.39, 19904.90, 28546.79, 14.98)
RE61 (Water resource planning)	3	6	Continuous	Unknown	(83060.03, 1350.00, 2853469.06, 16027067.60, 357719.74, 99660.36)

C. Detailed Experiments

C.1. Additional Results

Analysis on data pruning We first conduct ablation studies of data pruning on the Multi-Head model, as shown in Table 7. Although data pruning can alleviate the issue of model collapse in some problems, it does not consistently lead to improvements in all cases. Due to the severe impact of model collapse (sometimes resulting in only one solution), we default to using data pruning for all advanced methods. As mentioned in the main paper, exploring methods to mitigate model collapse is an important future direction in offline MOO.

Table 7. Average rank of Multi-Head and Multiple Models w/ and w/o data pruning on each type of task in Off-MOO-Bench.

Methods	Synthetic	MO-NAS	MORL	MOCO	Sci-Design	RE	Average Rank
$\mathcal{D}$(best)	4.34 ± 0.03	4.74 ± 0.21	2.75 ± 1.25	$\mathbf{1.21 \pm 0.14}$	3.12 ± 0.12	4.07 ± 0.13	3.67 ± 0.15
Multi-Head	2.66 ± 0.16	2.66 ± 0.24	$\mathbf{2.25 \pm 0.75}$	3.71 ± 0.00	3.38 ± 0.00	2.73 ± 0.00	2.90 ± 0.12
Multi-Head + Data Pruning	3.25 ± 0.12	$\mathbf{2.16 \pm 0.00}$	4.00 ± 1.00	3.31 ± 0.08	2.38 ± 0.25	3.12 ± 0.12	2.90 ± 0.02
Multiple Models	$\mathbf{2.12 \pm 0.06}$	2.61 ± 0.08	3.25 ± 0.25	3.07 ± 0.29	$\mathbf{2.00 \pm 0.25}$	$\mathbf{1.90 \pm 0.03}$	$\mathbf{2.41 \pm 0.09}$
Multiple Models + Data Pruning	2.62 ± 0.06	2.84 ± 0.11	2.75 ± 0.25	3.54 ± 0.23	3.83 ± 0.17	2.85 ± 0.00	2.94 ± 0.10

Analysis of the volume of data needed for MOBO. As we discussed before, the number of data points for GP is important due to the complexity of learning a GP. Here, we test the influence of the different number of data points, i.e., 50, 100, 200, and 400, on six randomly selected tasks. As shown in Figure 4 (a), 100 is a proper value. Thus, we use 100 for MOBO in our experiments on all the tasks.

Influence of the search algorithms. We compare four search algorithms on seven tasks, i.e., NSGA-II, MOEA/D, NSGA-III, and MOBO, as shown in Figure 4 (b). Their average ranks are 3.28, 2.07, 2.64, and 2.00, respectively. Although NSGA-II has the worst average ranking, we choose to use it as the default search algorithm due to its ease of use and popularity. These results also show that if a search algorithm specifically designed for offline MOO is implemented, the performance can be further improved, which is an interesting future work.

C.2. Detailed Results

Here, we provide the detailed results on different tasks. We provide results for each type of task with 256 solutions and 100th percentile evaluations. Additionally, we provide results for each type of task with 256 solutions and 50th percentile evaluations to demonstrate the robustness of the algorithms, and results with 32 solutions and 50th percentile evaluations to show the performance under the low-budget settings. Considering the three settings, Multiple Models + IOM, Multiple Models, and Multi-head Model are the top three performing algorithms, with average rankings of 4.91, 5.25, and 7.16, respectively. Note that $\mathcal{D}$(best) achieves the best average rank on MORL, MOCO, and Sci-Design tasks on the 256 solutions

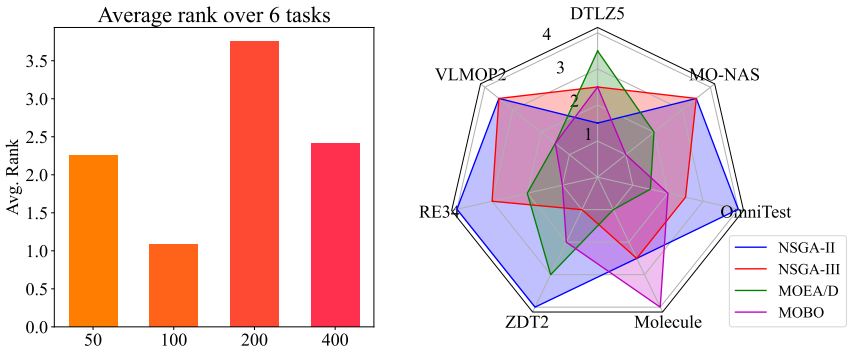

Figure 4. (a) The average rank of MOBO with different number of initial data points on six tasks. (b) The performance of four search algorithms on seven tasks.

with 50th percentile evaluations settings, underscoring the need for further enhancements in the robustness of offline MOO methods in these challenging tasks.

The average rank is calculated as follows: For each type of task (e.g., synthetic functions), we first determine the rankings for all methods across all sub-tasks (e.g., DTLZ1 and ZDT1) within it. After computing the six rankings for all methods, we average these values to report the average ranking of each method.

Table 8. Hypervolume results for synthetic functions with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	OmniTest	VLMOP1	VLMOP2	VLMOP3	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	10.43	9.43	9.71	**10.76**	9.06	8.20	8.32	3.87	0.08	1.64	45.14	4.04	4.70	5.05	**5.46**	4.76
End-to-End	10.12 ± 0.02	**10.65 ± 0.00**	10.65 ± 0.00	**10.70 ± 0.05**	**10.65 ± 0.00**	10.65 ± 0.00	10.70 ± 0.01	4.35 ± 0.00	2.57 ± 2.26	4.24 ± 0.01	46.93 ± 0.00	2.69 ± 0.00	3.21 ± 0.00	5.50 ± 0.04	3.12 ± 0.09	**4.92 ± 0.00**
End-to-End + GradNorm	10.65 ± 0.00	**10.65 ± 0.00**	10.65 ± 0.00	**10.76 ± 0.00**	10.54 ± 0.09	10.64 ± 0.00	10.71 ± 0.00	3.76 ± 0.03	2.33 ± 2.33	2.79 ± 1.34	42.23 ± 0.98	4.77 ± 0.01	5.63 ± 0.02	5.27 ± 0.03	3.23 ± 0.03	3.81 ± 1.02
End-to-End + PcGrad	10.65 ± 0.00	**10.65 ± 0.00**	10.65 ± 0.00	**10.70 ± 0.05**	9.02 ± 0.10	9.45 ± 0.15	10.52 ± 0.00	**4.35 ± 0.00**	2.57 ± 2.26	4.14 ± 0.07	46.79 ± 0.06	4.84 ± 0.01	5.70 ± 0.01	5.45 ± 0.00	3.12 ± 0.01	2.04 ± 0.22
Multi-Head	10.38 ± 0.25	**10.65 ± 0.00**	**10.65 ± 0.00**	**10.70 ± 0.05**	**10.65 ± 0.00**	10.65 ± 0.00	**10.63 ± 0.11**	4.30 ± 0.05	2.57 ± 2.26	4.26 ± 0.00	46.92 ± 0.02	2.69 ± 0.00	**4.48 ± 1.27**	5.50 ± 0.04	3.23 ± 0.16	4.91 ± 0.00
Multi-Head + GradNorm	10.65 ± 0.00	**10.65 ± 0.00**	10.65 ± 0.00	**10.76 ± 0.00**	9.29 ± 0.86	10.62 ± 0.02	10.61 ± 0.10	4.34 ± 0.00	0.00 ± 0.00	4.13 ± 0.03	46.64 ± 0.22	4.83 ± 0.00	5.68 ± 0.05	5.26 ± 0.04	3.39 ± 0.00	4.87 ± 0.00
Multi-Head + PcGrad	10.64 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	**10.76 ± 0.00**	9.08 ± 0.35	10.59 ± 0.01	10.49 ± 0.01	4.35 ± 0.00	2.55 ± 2.24	4.01 ± 0.02	46.91 ± 0.00	2.73 ± 0.03	5.69 ± 0.03	5.45 ± 0.00	3.64 ± 0.17	2.17 ± 0.05
Multiple Models	10.65 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	**10.76 ± 0.00**	10.65 ± 0.00	10.65 ± 0.00	**10.73 ± 0.00**	**4.35 ± 0.00**	2.57 ± 2.26	**4.28 ± 0.00**	**46.94 ± 0.00**	4.75 ± 0.00	5.58 ± 0.00	**5.80 ± 0.01**	4.14 ± 0.20	4.91 ± 0.00
Multiple Models + COMs	10.64 ± 0.01	10.39 ± 0.18	10.59 ± 0.05	**10.70 ± 0.05**	10.57 ± 0.06	10.26 ± 0.25	9.64 ± 0.22	4.29 ± 0.03	2.54 ± 2.25	1.90 ± 0.05	46.78 ± 0.07	4.24 ± 0.01	4.89 ± 0.07	5.54 ± 0.02	4.56 ± 0.04	4.57 ± 0.00
Multiple Models + RoMA	10.64 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	**10.76 ± 0.00**	10.18 ± 0.45	10.65 ± 0.00	10.63 ± 0.03	3.03 ± 0.03	2.54 ± 2.24	1.46 ± 0.00	44.15 ± 2.36	**4.87 ± 0.00**	5.65 ± 0.00	5.78 ± 0.02	3.18 ± 0.05	1.77 ± 0.02
Multiple Models + IOM	10.65 ± 0.00	10.61 ± 0.02	10.62 ± 0.02	**10.76 ± 0.00**	10.63 ± 0.01	10.50 ± 0.11	**10.74 ± 0.08**	4.34 ± 0.00	2.55 ± 2.24	3.77 ± 0.01	46.92 ± 0.00	4.66 ± 0.01	**5.74 ± 0.01**	5.61 ± 0.01	4.65 ± 0.19	4.89 ± 0.02
Multiple Models + ICT	10.64 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	**10.76 ± 0.00**	10.63 ± 0.01	**10.65 ± 0.00**	**10.75 ± 0.02**	4.30 ± 0.00	0.26 ± 0.06	1.46 ± 0.00	46.74 ± 0.09	4.39 ± 0.01	5.53 ± 0.00	4.37 ± 0.03	3.44 ± 0.16	2.33 ± 0.11
Multiple Models + Tri-Mentoring	10.64 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	**10.76 ± 0.00**	10.59 ± 0.04	**10.65 ± 0.00**	10.67 ± 0.01	3.97 ± 0.00	**4.83 ± 0.00**	1.46 ± 0.00	46.82 ± 0.02	4.52 ± 0.02	5.55 ± 0.01	5.62 ± 0.09	3.47 ± 0.04	2.36 ± 0.28
MOBO	**10.65 ± 0.00**	10.27 ± 0.07	10.36 ± 0.11	10.66 ± 0.01	9.28 ± 0.14	9.38 ± 0.01	10.51 ± 0.05	4.35 ± 0.00	0.32 ± 0.00	2.18 ± 0.69	46.91 ± 0.03	4.44 ± 0.09	5.18 ± 0.09	5.41 ± 0.12	4.60 ± 0.13	3.96 ± 0.73
MOBO-qParEGO	10.63 ± 0.00	9.73 ± 0.20	9.80 ± 0.19	**10.76 ± 0.00**	9.03 ± 0.24	9.16 ± 0.10	10.25 ± 0.05	4.33 ± 0.00	0.29 ± 0.01	2.93 ± 0.06	46.93 ± 0.00	4.32 ± 0.02	5.12 ± 0.17	5.20 ± 0.01	4.81 ± 0.10	3.31 ± 0.03
MOBO-JES	10.61 ± 0.00	10.22 ± 0.08	10.23 ± 0.18	8.56 ± 0.07	9.67 ± 0.01	9.62 ± 0.04	9.36 ± 0.08	3.87 ± 0.00	N/A	1.46 ± 0.00	46.88 ± 0.00	3.97 ± 0.09	4.44 ± 0.07	5.17 ± 0.02	4.43 ± 0.08	3.09 ± 0.02

Table 9. Hypervolume results for MO-NAS with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9	NasBench201-Test
$\mathcal{D}$(best)	4.78	10.48	9.72	21.15	40.51	92.43	358.27	4.55	10.59	4.97	5.00	11.21	16.63	17.26	44.43	4.69	11.42	21.50	10.07
End-to-End	4.84 ± 0.00	10.49 ± 0.02	10.84 ± 0.01	26.25 ± 0.07	50.20 ± 0.03	112.47 ± 0.01	523.26 ± 0.11	4.56 ± 0.07	10.50 ± 0.15	4.85 ± 0.07	4.70 ± 0.02	10.88 ± 0.10	17.07 ± 0.03	**17.93 ± 0.13**	**45.89 ± 0.16**	5.17 ± 0.02	11.15 ± 0.18	23.35 ± 0.30	10.30 ± 0.03
End-to-End + GradNorm	4.84 ± 0.00	10.54 ± 0.01	**10.86 ± 0.01**	26.46 ± 0.04	**50.64 ± 0.00**	112.27 ± 0.03	525.63 ± 0.10	4.56 ± 0.08	10.48 ± 0.23	5.21 ± 0.00	5.04 ± 0.02	10.87 ± 0.05	16.93 ± 0.07	17.80 ± 0.09	45.40 ± 0.12	4.96 ± 0.05	10.73 ± 0.08	22.55 ± 0.13	9.94 ± 0.01
End-to-End + PcGrad	4.84 ± 0.00	10.50 ± 0.01	10.81 ± 0.00	26.30 ± 0.04	50.54 ± 0.08	112.15 ± 0.01	526.46 ± 0.12	4.58 ± 0.04	11.10 ± 0.04	5.27 ± 0.04	5.20 ± 0.03	11.11 ± 0.03	16.82 ± 0.01	17.63 ± 0.01	44.47 ± 0.04	5.03 ± 0.07	11.20 ± 0.20	23.43 ± 0.24	10.45 ± 0.09
Multi-Head	4.83 ± 0.01	10.55 ± 0.00	10.60 ± 0.00	26.59 ± 0.01	50.45 ± 0.03	**112.72 ± 0.00**	**529.51 ± 0.36**	4.52 ± 0.09	9.88 ± 0.15	**5.37 ± 0.00**	5.05 ± 0.01	11.55 ± 0.01	17.03 ± 0.04	**18.00 ± 0.06**	**46.01 ± 0.10**	5.33 ± 0.00	11.73 ± 0.10	25.31 ± 0.14	10.36 ± 0.01
Multi-Head + GradNorm	**4.85 ± 0.00**	**10.57 ± 0.02**	10.23 ± 0.00	24.49 ± 0.06	50.24 ± 0.00	111.16 ± 0.09	525.72 ± 0.21	3.90 ± 0.05	9.43 ± 0.13	5.05 ± 0.01	4.41 ± 0.09	10.07 ± 0.07	**17.03 ± 0.19**	17.74 ± 0.08	45.25 ± 0.18	4.97 ± 0.01	8.61 ± 0.05	20.78 ± 0.20	10.35 ± 0.00
Multi-Head + PcGrad	**4.85 ± 0.00**	**10.58 ± 0.03**	**10.86 ± 0.01**	24.44 ± 0.11	50.53 ± 0.01	112.13 ± 0.19	520.43 ± 0.27	4.51 ± 0.09	10.51 ± 0.18	4.99 ± 0.10	5.18 ± 0.02	11.40 ± 0.13	17.01 ± 0.06	17.76 ± 0.15	45.14 ± 0.13	5.53 ± 0.05	10.95 ± 0.10	23.69 ± 0.29	10.35 ± 0.01
Multiple Models	4.82 ± 0.00	10.54 ± 0.00	10.83 ± 0.01	**26.62 ± 0.02**	50.20 ± 0.02	111.91 ± 0.03	525.80 ± 0.28	4.75 ± 0.04	11.17 ± 0.06	5.34 ± 0.00	**5.22 ± 0.01**	11.57 ± 0.01	**17.19 ± 0.04**	17.82 ± 0.00	**46.02 ± 0.04**	**5.64 ± 0.11**	11.77 ± 0.03	24.92 ± 0.24	10.34 ± 0.02
Multiple Models + COMs	4.84 ± 0.01	**10.57 ± 0.03**	10.66 ± 0.07	25.66 ± 0.12	50.42 ± 0.01	111.31 ± 0.50	525.66 ± 1.11	**4.90 ± 0.00**	10.93 ± 0.04	5.26 ± 0.00	5.17 ± 0.00	11.58 ± 0.01	16.79 ± 0.01	17.76 ± 0.00	**46.02 ± 0.23**	5.57 ± 0.05	12.16 ± 0.06	26.29 ± 0.05	10.40 ± 0.12
Multiple Models + RoMA	4.83 ± 0.00	10.54 ± 0.01	10.63 ± 0.12	26.23 ± 0.04	50.51 ± 0.00	112.44 ± 0.22	527.32 ± 0.45	4.49 ± 0.01	10.77 ± 0.02	5.25 ± 0.04	**5.23 ± 0.01**	11.55 ± 0.04	17.09 ± 0.02	17.84 ± 0.00	**45.90 ± 0.26**	**5.65 ± 0.02**	12.37 ± 0.01	26.55 ± 0.05	10.33 ± 0.03
Multiple Models + IOM	4.83 ± 0.01	10.43 ± 0.02	10.73 ± 0.00	26.04 ± 0.20	50.52 ± 0.01	112.47 ± 0.04	527.83 ± 0.01	**4.88 ± 0.05**	**11.50 ± 0.02**	5.28 ± 0.01	5.21 ± 0.02	**11.62 ± 0.01**	16.97 ± 0.01	17.81 ± 0.07	45.72 ± 0.13	5.27 ± 0.01	**12.42 ± 0.02**	**26.61 ± 0.02**	**10.54 ± 0.00**
Multiple Models + ICT	4.42 ± 0.36	10.49 ± 0.01	10.01 ± 0.03	25.82 ± 0.03	50.36 ± 0.09	108.86 ± 3.66	493.14 ± 0.99	3.62 ± 0.04	8.21 ± 0.14	4.75 ± 0.01	4.99 ± 0.02	9.60 ± 0.23	16.72 ± 0.02	16.85 ± 0.23	45.61 ± 0.01	4.79 ± 0.40	11.41 ± 0.24	21.48 ± 0.31	9.62 ± 0.29
Multiple Models + Tri-Mentoring	4.39 ± 0.33	7.77 ± 0.01	9.87 ± 0.55	25.34 ± 0.16	**50.62 ± 0.04**	111.60 ± 0.07	526.05 ± 0.78	3.61 ± 0.07	8.04 ± 0.25	4.86 ± 0.10	4.66 ± 0.07	10.19 ± 0.06	16.75 ± 0.10	17.12 ± 0.15	44.75 ± 0.70	5.23 ± 0.02	10.60 ± 0.63	22.54 ± 0.94	10.17 ± 0.15
MOBO	4.82 ± 0.02	**10.58 ± 0.01**	10.70 ± 0.00	26.35 ± 0.27	50.32 ± 0.01	111.28 ± 0.58	488.97 ± 4.01	4.71 ± 0.00	11.11 ± 0.03	5.25 ± 0.04	5.16 ± 0.02	11.24 ± 0.03	16.74 ± 0.05	17.56 ± 0.10	44.44 ± 0.06	5.21 ± 0.02	**12.33 ± 0.14**	25.65 ± 0.03	10.49 ± 0.01
MOBO-qParEGO	4.78 ± 0.01	10.44 ± 0.00	8.94 ± 0.05	20.01 ± 0.05	37.21 ± 0.00	94.72 ± 5.91	350.55 ± 0.13	4.50 ± 0.00	10.83 ± 0.13	4.93 ± 0.08	4.70 ± 0.01	10.34 ± 0.16	16.72 ± 0.14	17.62 ± 0.06	44.63 ± 0.27	5.11 ± 0.08	11.39 ± 0.27	23.75 ± 0.25	10.43 ± 0.04
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	9.16 ± 0.07

C.3. Additional Visualization Results

In this section, we visualize all the tasks with number of objectives less than 3, for better understanding the tasks. We first show the dataset of tasks in Off-MOO-Bench in Figure 5, and then show the Pareto fronts found by Multi-Head + GradNorm in Figure 6.

Table 10. Hypervolume results for MORL with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	4.21	2.85
End-to-End	4.76 ± 0.25	2.77 ± 0.03
End-to-End + GradNorm	$\mathbf{5.02 \pm 0.04}$	2.90 ± 0.07
End-to-End + PcGrad	4.60 ± 0.27	2.49 ± 0.05
Multi-Head	4.57 ± 0.28	2.91 ± 0.04
Multi-Head + GradNorm	3.78 ± 0.05	2.69 ± 0.24
Multi-Head + PcGrad	4.27 ± 0.61	2.49 ± 0.25
Multiple Models	4.58 ± 0.19	2.60 ± 0.15
Multiple Models + COMs	4.84 ± 0.17	2.71 ± 0.04
Multiple Models + RoMA	$\mathbf{5.23 \pm 0.23}$	2.78 ± 0.20
Multiple Models + IOM	$\mathbf{5.32 \pm 0.49}$	2.94 ± 0.11
Multiple Models + ICT	4.67 ± 0.12	$\mathbf{3.11 \pm 0.08}$
Multiple Models + Tri-Mentoring	4.93 ± 0.11	2.82 ± 0.10
MOBO	4.43 ± 0.08	2.61 ± 0.02
MOBO-qParEGO	N/A	N/A
MOBO-JES	N/A	N/A

Table 11. Hypervolume results for MOCO with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Bi-CVRP-20	Bi-CVRP-50	Bi-CVRP-100	Bi-KP-50	Bi-KP-100	Bi-KP-200	Bi-TSP-20	Bi-TSP-50	Bi-TSP-100	Bi-TSP-500	Tri-TSP-20	Tri-TSP-50	Tri-TSP-100	MO-Portfolio
$\mathcal{D}$(best)	**5.37**	**5.11**	**4.93**	3.00	**3.45**	**4.68**	**5.05**	**4.89**	4.55	**4.52**	**11.88**	9.82	9.36	**3.78**
End-to-End	4.85 ± 0.18	4.70 ± 0.17	4.91 ± 0.02	2.98 ± 0.00	3.02 ± 0.00	3.71 ± 0.47	3.64 ± 0.08	4.57 ± 0.08	$\mathbf{4.60 \pm 0.01}$	4.20 ± 0.29	6.96 ± 0.52	4.67 ± 0.17	9.26 ± 0.23	3.07 ± 0.16
End-to-End + GradNorm	4.77 ± 0.06	3.68 ± 0.20	3.84 ± 0.05	3.06 ± 0.00	2.84 ± 0.14	3.25 ± 0.01	3.12 ± 0.13	4.33 ± 0.00	4.04 ± 0.06	4.45 ± 0.02	7.84 ± 0.17	8.96 ± 0.07	9.41 ± 0.04	3.28 ± 0.15
End-to-End + PcGrad	4.56 ± 0.06	4.32 ± 0.09	4.15 ± 0.34	$\mathbf{3.16 \pm 0.02}$	3.08 ± 0.03	3.52 ± 0.27	3.08 ± 0.03	4.68 ± 0.03	4.55 ± 0.02	4.17 ± 0.12	10.83 ± 0.03	8.15 ± 0.39	9.68 ± 0.15	3.08 ± 0.05
Multi-Head	4.00 ± 0.06	4.03 ± 0.26	$\mathbf{4.78 \pm 0.16}$	$\mathbf{3.11 \pm 0.06}$	2.88 ± 0.04	4.03 ± 0.41	3.36 ± 0.14	4.60 ± 0.09	4.04 ± 0.00	3.70 ± 0.27	6.31 ± 0.14	6.43 ± 0.42	8.60 ± 0.13	3.18 ± 0.04
Multi-Head + GradNorm	4.50 ± 0.51	3.69 ± 0.12	4.35 ± 0.57	2.88 ± 0.07	2.47 ± 0.01	2.98 ± 0.41	4.39 ± 0.17	3.40 ± 0.02	3.49 ± 0.39	3.83 ± 0.02	7.84 ± 0.20	6.48 ± 0.65	8.48 ± 0.23	3.11 ± 0.11
Multi-Head + PcGrad	4.48 ± 0.48	4.24 ± 0.12	3.45 ± 0.11	$\mathbf{3.13 \pm 0.06}$	2.60 ± 0.37	4.34 ± 0.10	2.85 ± 0.10	3.89 ± 0.12	3.59 ± 0.19	2.62 ± 0.88	10.39 ± 0.14	7.46 ± 1.00	$\mathbf{10.10 \pm 0.07}$	3.09 ± 0.13
Multiple Models	4.95 ± 0.06	4.88 ± 0.03	4.92 ± 0.00	3.03 ± 0.02	3.19 ± 0.07	3.84 ± 0.54	3.55 ± 0.05	3.27 ± 0.04	4.22 ± 0.18	$\mathbf{4.51 \pm 0.01}$	7.26 ± 0.09	6.90 ± 0.08	7.68 ± 0.38	3.69 ± 0.03
Multiple Models + COMs	5.28 ± 0.00	4.27 ± 0.14	4.23 ± 0.10	2.97 ± 0.00	3.11 ± 0.05	4.05 ± 0.46	4.64 ± 0.05	4.54 ± 0.06	4.30 ± 0.07	4.02 ± 0.10	10.94 ± 0.46	8.55 ± 0.40	8.84 ± 0.26	2.20 ± 0.02
Multiple Models + RoMA	4.56 ± 0.01	4.22 ± 0.23	3.97 ± 0.15	2.80 ± 0.06	3.11 ± 0.09	3.74 ± 0.26	4.25 ± 0.36	4.48 ± 0.04	4.31 ± 0.01	2.51 ± 0.69	9.37 ± 0.51	7.77 ± 0.03	9.10 ± 0.09	2.92 ± 0.02
Multiple Models + IOM	5.28 ± 0.01	5.11 ± 0.00	$\mathbf{4.93 \pm 0.00}$	2.98 ± 0.02	2.86 ± 0.05	4.12 ± 0.21	4.85 ± 0.04	4.79 ± 0.01	4.53 ± 0.00	4.39 ± 0.09	11.65 ± 0.03	$\mathbf{10.19 \pm 0.05}$	9.89 ± 0.02	2.93 ± 0.00
Multiple Models + ICT	4.15 ± 0.21	4.29 ± 0.17	3.96 ± 0.08	2.75 ± 0.11	2.68 ± 0.14	3.56 ± 0.09	4.40 ± 0.34	3.88 ± 0.24	3.75 ± 0.04	4.20 ± 0.11	7.09 ± 0.03	8.37 ± 0.02	7.50 ± 0.12	2.05 ± 0.10
Multiple Models + Tri-Mentoring	4.12 ± 0.16	4.06 ± 0.20	4.09 ± 0.13	2.90 ± 0.10	2.88 ± 0.04	3.32 ± 0.05	3.96 ± 0.18	3.75 ± 0.11	4.17 ± 0.14	3.69 ± 0.39	7.50 ± 0.47	8.16 ± 0.23	8.23 ± 0.27	2.63 ± 0.12
MOBO	3.38 ± 0.23	2.40 ± 0.09	1.58 ± 0.02	2.73 ± 0.14	2.33 ± 0.05	1.89 ± 0.06	2.42 ± 0.16	1.77 ± 0.01	1.56 ± 0.06	N/A	5.64 ± 0.04	3.36 ± 0.12	2.54 ± 0.03	3.29 ± 0.02
MOBO-qParEGO	3.69 ± 0.13	2.77 ± 0.10	1.66 ± 0.02	2.72 ± 0.06	2.33 ± 0.01	N/A	2.70 ± 0.09	2.10 ± 0.01	1.75 ± 0.01	N/A	4.40 ± 0.14	3.54 ± 0.00	2.38 ± 0.01	3.15 ± 0.04
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3.53 ± 0.07

Table 12. Hypervolume results for scientific design with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Molecule	Regex	RFP	ZINC
$\mathcal{D}$(best)	2.26	2.82	3.36	4.01
End-to-End	$\mathbf{2.30 \pm 0.48}$	2.80 ± 0.00	3.80 ± 0.04	4.17 ± 0.00
End-to-End + GradNorm	1.10 ± 0.03	2.80 ± 0.00	4.11 ± 0.30	4.17 ± 0.00
End-to-End + PcGrad	1.54 ± 0.53	2.80 ± 0.00	3.84 ± 0.05	4.16 ± 0.08
Multi-Head	2.08 ± 0.00	2.80 ± 0.00	3.75 ± 0.00	4.16 ± 0.00
Multi-Head + GradNorm	1.62 ± 0.61	2.38 ± 0.00	4.08 ± 0.32	4.21 ± 0.05
Multi-Head + PcGrad	1.22 ± 0.10	2.80 ± 0.00	4.19 ± 0.22	4.12 ± 0.02
Multiple Models	$\mathbf{2.78 \pm 0.00}$	2.80 ± 0.00	$\mathbf{4.40 \pm 0.02}$	4.16 ± 0.00
Multiple Models + COMs	2.30 ± 0.00	2.21 ± 0.17	$\mathbf{4.14 \pm 0.35}$	4.12 ± 0.05
Multiple Models + RoMA	1.65 ± 0.02	2.80 ± 0.00	$\mathbf{4.13 \pm 0.29}$	4.16 ± 0.01
Multiple Models + IOM	1.75 ± 0.33	2.80 ± 0.00	$\mathbf{4.13 \pm 0.28}$	4.17 ± 0.00
Multiple Models + ICT	1.37 ± 0.17	2.80 ± 0.00	$\mathbf{4.41 \pm 0.00}$	4.10 ± 0.07
Multiple Models + Tri-Mentoring	2.03 ± 0.00	2.80 ± 0.00	$\mathbf{4.12 \pm 0.29}$	4.06 ± 0.01
MOBO	2.22 ± 0.08	$\mathbf{5.12 \pm 0.17}$	3.74 ± 0.00	$\mathbf{4.26 \pm 0.00}$
MOBO-qParEGO	2.12 ± 0.04	4.26 ± 0.25	3.33 ± 0.00	4.05 ± 0.02
MOBO-JES	$\mathbf{2.10 \pm 1.04}$	N/A	N/A	N/A

Table 13. Hypervolume results for RE with 256 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	4.23	4.78	4.75	4.59	4.79	10.23	10.53	10.59	48.06	10.96	7.57	4.72	36.17	12.53	135.87
End-to-End	4.42 ± 0.00	4.84 ± 0.00	**4.84 ± 0.00**	4.38 ± 0.00	4.84 ± 0.00	10.56 ± 0.00	10.64 ± 0.00	10.69 ± 0.00	52.86 ± 0.05	11.69 ± 0.00	9.25 ± 1.00	6.21 ± 0.00	44.13 ± 0.02	20.04 ± 0.01	**144.30 ± 0.02**
End-to-End + GradNorm	4.81 ± 0.01	4.84 ± 0.00	2.64 ± 0.00	4.38 ± 0.00	**4.84 ± 0.00**	10.65 ± 0.00	10.63 ± 0.00	9.90 ± 0.00	51.38 ± 0.03	11.52 ± 0.00	9.16 ± 0.02	6.22 ± 0.01	41.26 ± 0.81	13.46 ± 0.00	141.37 ± 0.03
End-to-End + PcGrad	4.90 ± 0.05	4.84 ± 0.00	4.84 ± 0.00	4.38 ± 0.00	4.60 ± 0.24	**10.65 ± 0.00**	10.65 ± 0.00	10.41 ± 0.07	52.83 ± 0.08	11.68 ± 0.02	10.02 ± 0.00	5.52 ± 0.00	43.53 ± 0.37	14.27 ± 0.12	142.98 ± 0.38
Multi-Head	4.57 ± 0.15	4.84 ± 0.00	4.74 ± 0.00	4.78 ± 0.00	4.60 ± 0.24	10.65 ± 0.00	10.64 ± 0.00	10.69 ± 0.00	52.93 ± 0.01	11.74 ± 0.00	6.76 ± 0.00	5.78 ± 0.05	44.06 ± 0.02	**20.71 ± 0.29**	141.28 ± 1.13
Multi-Head + GradNorm	4.91 ± 0.00	4.83 ± 0.01	4.49 ± 0.09	2.64 ± 0.00	3.95 ± 0.00	10.65 ± 0.00	10.63 ± 0.00	5.85 ± 0.00	52.84 ± 0.00	11.52 ± 0.00	0.02 ± 0.00	6.36 ± 0.01	43.77 ± 0.09	19.01 ± 0.04	143.82 ± 0.20
Multi-Head + PcGrad	4.91 ± 0.01	4.84 ± 0.00	4.27 ± 0.12	4.83 ± 0.00	4.35 ± 0.00	7.66 ± 0.00	10.08 ± 0.00	10.61 ± 0.00	52.84 ± 0.02	11.60 ± 0.08	9.95 ± 0.14	6.42 ± 0.00	43.80 ± 0.08	20.39 ± 0.45	142.22 ± 0.38
Multiple Models	4.94 ± 0.00	4.84 ± 0.00	4.84 ± 0.00	4.82 ± 0.00	4.84 ± 0.00	10.65 ± 0.00	10.63 ± 0.00	10.67 ± 0.00	**54.16 ± 0.01**	11.70 ± 0.00	**10.52 ± 0.00**	6.49 ± 0.00	44.09 ± 0.12	**20.89 ± 0.07**	144.20 ± 0.02
Multiple Models + COMs	4.20 ± 0.38	4.84 ± 0.00	4.79 ± 0.01	4.59 ± 0.00	4.84 ± 0.00	5.28 ± 5.28	10.64 ± 0.00	10.56 ± 0.03	50.57 ± 1.05	11.55 ± 0.02	8.96 ± 0.02	5.99 ± 0.03	40.84 ± 0.30	15.12 ± 0.32	141.00 ± 0.81
Multiple Models + RoMA	4.92 ± 0.00	4.84 ± 0.00	**4.84 ± 0.00**	4.79 ± 0.02	4.69 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	10.66 ± 0.00	52.73 ± 0.02	**11.78 ± 0.00**	8.08 ± 0.40	6.49 ± 0.01	43.80 ± 0.08	19.53 ± 0.21	143.61 ± 0.20
Multiple Models + IOM	**4.94 ± 0.00**	4.84 ± 0.00	4.84 ± 0.00	**4.84 ± 0.00**	**4.84 ± 0.00**	**10.65 ± 0.00**	**10.65 ± 0.00**	10.68 ± 0.00	54.12 ± 0.02	11.75 ± 0.01	10.02 ± 0.01	6.54 ± 0.00	43.92 ± 0.01	20.78 ± 0.02	143.30 ± 0.14
Multiple Models + ICT	4.75 ± 0.17	4.84 ± 0.00	2.77 ± 0.00	4.67 ± 0.00	4.84 ± 0.00	10.65 ± 0.00	2.77 ± 0.00	10.51 ± 0.00	53.53 ± 0.01	11.70 ± 0.02	8.94 ± 0.01	6.25 ± 0.07	43.96 ± 0.11	20.59 ± 0.20	143.11 ± 0.23
Multiple Models + Tri-Mentoring	4.91 ± 0.00	4.84 ± 0.00	2.76 ± 0.00	4.83 ± 0.00	4.70 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	10.54 ± 0.00	53.39 ± 0.02	11.75 ± 0.00	9.72 ± 0.06	6.38 ± 0.06	43.81 ± 0.13	20.30 ± 0.11	143.82 ± 0.07
MOBO	4.70 ± 0.08	**4.84 ± 0.00**	4.84 ± 0.00	4.83 ± 0.00	4.84 ± 0.00	10.19 ± 0.00	10.64 ± 0.01	**10.69 ± 0.00**	52.06 ± 0.00	11.75 ± 0.00	0.00 ± 0.00	**6.60 ± 0.00**	**53.86 ± 0.06**	15.82 ± 0.64	N/A
MOBO-qParEGO	4.65 ± 0.04	4.61 ± 0.00	4.84 ± 0.00	3.74 ± 0.00	4.71 ± 0.00	10.64 ± 0.01	9.77 ± 0.02	10.61 ± 0.03	49.27 ± 1.14	0.00 ± 0.00	0.00 ± 0.00	5.87 ± 0.05	N/A	N/A	N/A
MOBO-JES	4.85 ± 0.03	4.84 ± 0.00	4.83 ± 0.00	4.82 ± 0.00	4.84 ± 0.00	10.28 ± 0.00	10.65 ± 0.00	10.61 ± 0.03	50.30 ± 0.00	11.59 ± 0.02	9.43 ± 0.10	6.20 ± 0.03	N/A	N/A	N/A

Table 14. Average rank of different offline MOO methods on each type of task with 256 solutions and 50th percentile evaluations, where the best and runner-up results are **bolded** and underlined, respectively.

Methods	Synthetic	MO-NAS	MORL	MOCO	Sci-Design	RE	Average Rank
$\mathcal{D}$(best)	9.20 ± 0.42	9.58 ± 0.11	**1.75 ± 0.25**	**1.07 ± 0.00**	**1.88 ± 0.62**	10.50 ± 0.30	7.22 ± 0.14
End-to-End	8.50 ± 0.38	7.53 ± 0.21	9.50 ± 2.00	7.38 ± 0.38	9.25 ± 1.00	6.33 ± 0.13	7.55 ± 0.19
End-to-End + GradNorm	9.34 ± 0.34	8.34 ± 0.29	5.75 ± 1.75	8.08 ± 0.92	8.00 ± 0.25	9.67 ± 0.00	8.76 ± 0.12
End-to-End + PcGrad	8.61 ± 0.17	8.08 ± 0.34	10.00 ± 1.00	8.50 ± 0.43	7.62 ± 0.12	7.77 ± 0.57	8.29 ± 0.18
Multi-Head	7.47 ± 1.09	5.16 ± 0.05	5.75 ± 1.75	9.46 ± 0.39	7.25 ± 0.25	7.90 ± 0.50	7.19 ± 0.48
Multi-Head + GradNorm	8.48 ± 0.52	10.46 ± 0.20	10.75 ± 2.75	8.64 ± 0.43	12.25 ± 0.25	10.07 ± 0.40	9.67 ± 0.23
Multi-Head + PcGrad	7.95 ± 0.52	8.97 ± 0.66	7.25 ± 2.75	8.49 ± 0.28	11.79 ± 0.54	8.43 ± 0.10	8.55 ± 0.33
Multiple Models	**3.92 ± 0.58**	**3.99 ± 0.38**	8.75 ± 0.25	6.72 ± 0.43	9.56 ± 0.81	**4.40 ± 0.27**	**5.01 ± 0.01**
Multiple Models + COMs	8.81 ± 0.06	5.66 ± 0.08	8.00 ± 0.00	6.63 ± 0.94	7.62 ± 0.62	9.75 ± 0.22	7.63 ± 0.11
Multiple Models + RoMA	9.23 ± 0.36	4.97 ± 0.29	5.25 ± 1.25	7.04 ± 0.54	7.25 ± 0.50	8.13 ± 0.00	7.26 ± 0.12
Multiple Models + IOM	5.98 ± 0.45	4.16 ± 0.79	6.25 ± 0.75	4.21 ± 0.00	7.00 ± 0.12	5.03 ± 0.43	5.00 ± 0.36
Multiple Models + ICT	8.88 ± 0.47	12.92 ± 0.03	6.50 ± 0.00	9.14 ± 0.14	9.25 ± 0.00	8.53 ± 0.20	9.89 ± 0.09
Multiple Models + Tri-Mentoring	7.95 ± 0.02	12.13 ± 0.13	6.75 ± 0.25	8.71 ± 0.14	8.50 ± 0.25	5.30 ± 0.23	8.64 ± 0.06
MOBO	9.73 ± 0.40	6.42 ± 0.47	12.75 ± 0.25	12.35 ± 0.12	6.81 ± 0.19	9.32 ± 0.50	9.13 ± 0.36
MOBO-qParEGO	9.78 ± 0.09	11.24 ± 0.34	N/A	13.12 ± 0.21	6.12 ± 0.12	12.52 ± 0.31	11.15 ± 0.12
MOBO-JES	11.50 ± 0.57	15.00 ± 1.00	N/A	5.50 ± 2.50	14.00 ± 0.00	10.42 ± 0.08	11.06 ± 0.20

Table 15. Hypervolume results for synthetic functions with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	OmniTest	VLMOP1	VLMOP2	VLMOP3	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	10.43	9.43	9.71	**10.76**	9.06	8.20	8.32	3.87	0.08	1.64	45.14	4.04	4.70	5.05	**5.46**	4.76
End-to-End	10.06 ± 0.00	**10.65 ± 0.00**	**10.65 ± 0.00**	9.98 ± 0.35	**8.63 ± 2.01**	9.42 ± 0.00	6.37 ± 0.07	**4.35 ± 0.00**	0.00 ± 0.00	4.18 ± 0.02	46.76 ± 0.09	2.69 ± 0.00	3.21 ± 0.00	5.46 ± 0.00	3.04 ± 0.02	**4.87 ± 0.02**
End-to-End + GradNorm	**10.65 ± 0.00**	10.36 ± 0.28	10.63 ± 0.02	**10.28 ± 0.48**	**8.50 ± 2.07**	8.69 ± 0.68	**8.62 ± 2.08**	2.32 ± 0.04	0.00 ± 0.00	2.67 ± 1.21	38.20 ± 0.17	4.76 ± 0.00	4.01 ± 0.17	5.27 ± 0.03	3.02 ± 0.02	2.55 ± 0.23
End-to-End + PcGrad	10.62 ± 0.02	8.22 ± 1.99	10.65 ± 0.00	**10.76 ± 0.00**	8.01 ± 0.07	7.80 ± 0.80	10.52 ± 0.00	4.32 ± 0.03	1.36 ± 1.36	4.08 ± 0.10	34.65 ± 0.06	4.37 ± 0.29	5.70 ± 0.01	4.45 ± 0.94	2.99 ± 0.02	1.87 ± 0.10
Multi-Head	10.37 ± 0.24	10.64 ± 0.01	10.63 ± 0.01	10.14 ± 0.19	6.62 ± 0.00	9.39 ± 0.03	**10.61 ± 0.09**	4.29 ± 0.05	0.95 ± 0.95	4.18 ± 0.01	**46.78 ± 0.16**	2.69 ± 0.00	**4.48 ± 1.27**	5.50 ± 0.03	2.94 ± 0.08	**4.90 ± 0.00**
Multi-Head + GradNorm	**10.64 ± 0.00**	10.37 ± 0.13	10.50 ± 0.12	**10.76 ± 0.00**	7.97 ± 0.95	8.67 ± 0.78	9.58 ± 0.93	3.43 ± 0.90	0.00 ± 0.00	4.06 ± 0.01	29.52 ± 0.54	4.82 ± 0.01	4.32 ± 0.24	4.14 ± 1.07	3.16 ± 0.06	4.83 ± 0.03
Multi-Head + PcGrad	10.61 ± 0.01	10.64 ± 0.00	10.58 ± 0.04	**10.76 ± 0.00**	6.73 ± 0.51	9.22 ± 0.13	10.36 ± 0.02	4.34 ± 0.00	1.47 ± 1.47	2.66 ± 1.21	45.33 ± 1.58	2.69 ± 0.01	5.68 ± 0.04	5.38 ± 0.02	3.49 ± 0.18	2.06 ± 0.15
Multiple Models	10.64 ± 0.00	10.63 ± 0.02	10.64 ± 0.00	**10.76 ± 0.00**	**8.52 ± 1.90**	10.19 ± 0.37	**10.56 ± 0.03**	**4.35 ± 0.00**	0.56 ± 0.56	**4.22 ± 0.00**	**46.93 ± 0.00**	4.75 ± 0.00	5.56 ± 0.00	**5.71 ± 0.01**	3.70 ± 0.38	4.87 ± 0.01
Multiple Models + COMs	10.55 ± 0.04	9.83 ± 0.37	9.76 ± 0.10	10.72 ± 0.03	**9.93 ± 0.22**	9.10 ± 0.65	8.73 ± 0.01	3.85 ± 0.21	0.00 ± 0.00	1.68 ± 0.01	46.03 ± 0.26	3.82 ± 0.16	4.66 ± 0.11	5.44 ± 0.07	4.31 ± 0.05	4.33 ± 0.01
Multiple Models + RoMA	10.53 ± 0.06	10.47 ± 0.05	10.62 ± 0.01	**10.76 ± 0.00**	6.63 ± 0.01	**10.57 ± 0.02**	10.01 ± 0.08	2.60 ± 0.01	0.00 ± 0.00	1.46 ± 0.00	40.48 ± 0.34	**4.86 ± 0.01**	5.62 ± 0.01	5.40 ± 0.18	2.87 ± 0.09	1.76 ± 0.02
Multiple Models + IOM	10.61 ± 0.00	8.93 ± 0.23	9.63 ± 0.07	**10.76 ± 0.00**	**9.66 ± 0.36**	9.25 ± 0.27	**10.55 ± 0.15**	4.34 ± 0.00	0.58 ± 0.58	3.73 ± 0.03	46.92 ± 0.00	4.62 ± 0.03	**5.72 ± 0.00**	5.50 ± 0.01	4.39 ± 0.44	4.86 ± 0.00
Multiple Models + ICT	10.63 ± 0.00	10.52 ± 0.01	10.63 ± 0.01	**10.76 ± 0.00**	**9.61 ± 0.50**	9.42 ± 0.00	9.94 ± 0.05	3.93 ± 0.00	0.06 ± 0.06	1.46 ± 0.00	43.55 ± 2.98	3.45 ± 0.07	5.50 ± 0.01	4.14 ± 0.12	3.27 ± 0.09	1.88 ± 0.01
Multiple Models + Tri-Mentoring	10.61 ± 0.01	**10.65 ± 0.00**	10.48 ± 0.07	**10.76 ± 0.00**	8.26 ± 0.02	9.42 ± 0.00	9.76 ± 0.01	3.39 ± 0.00	**3.73 ± 0.07**	1.46 ± 0.00	46.56 ± 0.08	4.33 ± 0.05	5.53 ± 0.01	5.45 ± 0.04	3.21 ± 0.22	1.90 ± 0.00
MOBO	10.64 ± 0.00	9.81 ± 0.14	8.85 ± 0.25	10.22 ± 0.00	8.73 ± 0.26	8.89 ± 0.35	8.00 ± 0.03	4.25 ± 0.01	0.00 ± 0.00	1.46 ± 0.00	46.91 ± 0.00	4.30 ± 0.01	4.34 ± 0.01	4.99 ± 0.04	3.88 ± 0.00	2.63 ± 0.11
MOBO-qParEGO	10.55 ± 0.08	8.62 ± 0.15	8.84 ± 0.48	**10.76 ± 0.00**	8.10 ± 0.10	7.55 ± 0.83	9.85 ± 0.14	4.19 ± 0.09	0.00 ± 0.00	1.46 ± 0.00	46.82 ± 0.03	4.11 ± 0.08	4.66 ± 0.06	4.96 ± 0.11	4.31 ± 0.07	2.51 ± 0.65
MOBO-JES	10.26 ± 0.10	10.13 ± 0.11	9.89 ± 0.48	8.31 ± 0.12	9.20 ± 0.10	9.39 ± 0.06	8.75 ± 0.07	2.98 ± 0.00	N/A	1.46 ± 0.00	45.77 ± 0.64	3.87 ± 0.04	3.90 ± 0.02	4.72 ± 0.10	3.97 ± 0.24	1.87 ± 0.10

Table 16. Hypervolume results for MO-NAS with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9	NasBench201-Test
$\mathcal{D}$(best)	4.78	10.48	9.72	21.15	40.51	92.43	358.27	4.55	10.59	4.97	5.00	11.21	16.63	17.26	44.43	4.69	11.42	21.50	10.07
End-to-End	**4.80 ± 0.00**	10.38 ± 0.10	10.66 ± 0.02	24.71 ± 0.03	48.81 ± 0.31	109.63 ± 0.09	513.62 ± 1.06	4.41 ± 0.07	9.29 ± 0.16	4.75 ± 0.01	4.04 ± 0.00	10.27 ± 0.13	16.81 ± 0.01	**17.68 ± 0.06**	**45.13 ± 0.21**	4.66 ± 0.22	10.33 ± 0.25	22.45 ± 0.31	10.22 ± 0.00
End-to-End + GradNorm	4.78 ± 0.00	10.51 ± 0.00	**10.72 ± 0.00**	24.11 ± 0.34	49.40 ± 0.08	110.13 ± 0.06	517.76 ± 1.15	4.25 ± 0.23	9.52 ± 0.34	5.11 ± 0.05	4.96 ± 0.03	10.29 ± 0.03	15.29 ± 0.00	17.40 ± 0.05	44.18 ± 0.00	4.60 ± 0.15	10.56 ± 0.24	22.07 ± 0.17	9.17 ± 0.03
End-to-End + PcGrad	**4.80 ± 0.00**	10.49 ± 0.01	10.65 ± 0.01	23.78 ± 0.15	50.02 ± 0.08	107.50 ± 0.05	512.26 ± 0.83	3.97 ± 0.01	10.72 ± 0.08	4.72 ± 0.01	**5.16 ± 0.04**	10.06 ± 0.57	15.42 ± 0.04	17.35 ± 0.02	43.51 ± 0.02	4.84 ± 0.01	10.15 ± 0.78	22.68 ± 0.10	10.01 ± 0.22
Multi-Head	4.80 ± 0.00	**10.52 ± 0.01**	10.36 ± 0.06	**24.79 ± 0.47**	49.91 ± 0.07	**111.62 ± 0.00**	**519.26 ± 1.09**	4.17 ± 0.04	9.03 ± 0.00	**5.31 ± 0.01**	4.93 ± 0.05	11.49 ± 0.02	16.72 ± 0.03	**17.69 ± 0.02**	**45.25 ± 0.07**	5.07 ± 0.01	11.33 ± 0.07	23.48 ± 0.53	9.99 ± 0.31
Multi-Head + GradNorm	4.79 ± 0.00	10.44 ± 0.05	9.05 ± 0.00	22.13 ± 0.41	48.51 ± 0.28	106.59 ± 0.06	515.87 ± 1.16	3.45 ± 0.06	7.92 ± 0.36	5.00 ± 0.03	4.30 ± 0.08	9.90 ± 0.05	**16.76 ± 0.12**	17.55 ± 0.08	41.23 ± 2.98	4.80 ± 0.02	7.93 ± 0.10	19.15 ± 0.26	10.24 ± 0.00
Multi-Head + PcGrad	4.80 ± 0.00	10.34 ± 0.18	9.71 ± 0.02	19.72 ± 0.19	49.20 ± 0.67	107.25 ± 0.06	504.65 ± 1.50	4.23 ± 0.06	9.36 ± 0.05	4.91 ± 0.05	5.12 ± 0.01	11.30 ± 0.11	16.66 ± 0.08	16.87 ± 0.44	44.17 ± 0.05	5.35 ± 0.07	10.53 ± 0.11	22.51 ± 0.48	9.69 ± 0.07
Multiple Models	4.78 ± 0.01	**10.52 ± 0.01**	10.72 ± 0.01	**24.85 ± 0.08**	49.34 ± 0.09	109.68 ± 0.35	**519.10 ± 0.05**	4.56 ± 0.05	10.04 ± 0.19	**5.30 ± 0.02**	**5.17 ± 0.01**	11.54 ± 0.02	16.82 ± 0.02	17.67 ± 0.02	**45.30 ± 0.13**	**5.47 ± 0.08**	11.27 ± 0.36	22.29 ± 1.00	10.24 ± 0.00
Multiple Models + COMs	**4.78 ± 0.03**	10.51 ± 0.00	10.42 ± 0.05	24.42 ± 0.08	48.71 ± 0.56	108.69 ± 0.12	508.52 ± 1.36	**4.75 ± 0.01**	10.40 ± 0.00	5.14 ± 0.03	5.10 ± 0.01	11.43 ± 0.05	16.53 ± 0.00	17.56 ± 0.01	44.51 ± 0.04	5.24 ± 0.09	11.96 ± 0.17	25.68 ± 0.19	10.21 ± 0.02
Multiple Models + RoMA	4.80 ± 0.00	**10.52 ± 0.00**	10.44 ± 0.12	24.52 ± 0.04	47.08 ± 0.11	108.04 ± 0.10	**517.85 ± 1.26**	4.26 ± 0.10	10.21 ± 0.02	5.02 ± 0.05	**5.18 ± 0.03**	11.48 ± 0.05	**16.86 ± 0.03**	17.51 ± 0.01	45.05 ± 0.11	**5.43 ± 0.06**	**12.27 ± 0.01**	**26.22 ± 0.04**	10.23 ± 0.00
Multiple Models + IOM	**4.80 ± 0.01**	10.38 ± 0.00	10.53 ± 0.00	**24.63 ± 0.24**	48.43 ± 0.72	110.39 ± 1.00	518.27 ± 0.04	**4.64 ± 0.13**	**11.05 ± 0.06**	5.20 ± 0.03	5.14 ± 0.02	**11.57 ± 0.01**	16.76 ± 0.01	**17.64 ± 0.08**	**45.23 ± 0.16**	4.94 ± 0.01	**12.31 ± 0.09**	26.08 ± 0.01	**10.43 ± 0.08**
Multiple Models + ICT	2.31 ± 2.31	10.22 ± 0.24	8.34 ± 0.03	21.97 ± 0.06	48.94 ± 0.05	93.35 ± 2.86	447.32 ± 2.89	3.24 ± 0.07	7.31 ± 0.18	4.65 ± 0.02	4.91 ± 0.01	9.25 ± 0.11	15.14 ± 0.01	14.82 ± 0.26	43.85 ± 0.28	4.57 ± 0.39	10.44 ± 0.07	20.49 ± 0.17	9.14 ± 0.03
Multiple Models + Tri-Mentoring	4.38 ± 0.32	3.89 ± 3.89	8.65 ± 0.96	23.69 ± 0.01	48.86 ± 0.02	107.89 ± 0.47	506.44 ± 1.45	3.27 ± 0.02	7.32 ± 0.15	4.69 ± 0.01	4.52 ± 0.08	9.93 ± 0.10	15.24 ± 0.01	15.35 ± 0.37	41.14 ± 0.99	4.89 ± 0.00	9.65 ± 0.73	21.11 ± 0.99	9.74 ± 0.56
MOBO	4.79 ± 0.00	10.46 ± 0.02	10.56 ± 0.00	**24.46 ± 0.48**	**50.27 ± 0.02**	107.95 ± 2.80	482.04 ± 6.29	4.66 ± 0.00	10.97 ± 0.01	5.21 ± 0.01	5.12 ± 0.05	10.96 ± 0.06	16.44 ± 0.06	17.40 ± 0.06	44.05 ± 0.19	5.17 ± 0.03	**12.12 ± 0.27**	25.10 ± 0.18	9.53 ± 0.04
MOBO-qParEGO	4.69 ± 0.07	10.39 ± 0.04	8.45 ± 0.19	19.62 ± 0.44	37.17 ± 0.02	94.55 ± 5.93	337.57 ± 8.91	4.13 ± 0.06	10.23 ± 0.15	4.60 ± 0.00	4.30 ± 0.12	9.93 ± 0.53	16.47 ± 0.03	17.37 ± 0.01	44.19 ± 0.06	4.74 ± 0.11	11.33 ± 0.27	23.52 ± 0.35	10.01 ± 0.09
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8.96 ± 0.16

Table 17. Hypervolume results for MORL with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	4.21	**2.85**
End-to-End	3.68 ± 0.00	2.04 ± 0.10
End-to-End + GradNorm	3.94 ± 0.23	2.08 ± 0.02
End-to-End + PcGrad	3.72 ± 0.01	1.90 ± 0.05
Multi-Head	3.74 ± 0.07	2.66 ± 0.04
Multi-Head + GradNorm	3.67 ± 0.00	1.98 ± 0.12
Multi-Head + PcGrad	3.86 ± 0.18	2.08 ± 0.02
Multiple Models	3.76 ± 0.01	1.91 ± 0.02
Multiple Models + COMs	3.72 ± 0.02	1.98 ± 0.01
Multiple Models + RoMA	$\mathbf{4.74 \pm 0.00}$	1.95 ± 0.06
Multiple Models + IOM	4.17 ± 0.18	1.96 ± 0.06
Multiple Models + ICT	3.70 ± 0.01	2.38 ± 0.11
Multiple Models + Tri-Mentoring	3.82 ± 0.03	1.98 ± 0.01
MOBO	3.68 ± 0.00	1.49 ± 0.02
MOBO-qParEGO	N/A	N/A
MOBO-JES	N/A	N/A

Table 18. Hypervolume results for MOCO with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Bi-CVRP-20	Bi-CVRP-50	Bi-CVRP-100	Bi-KP-50	Bi-KP-100	Bi-KP-200	Bi-TSP-20	Bi-TSP-50	Bi-TSP-100	Bi-TSP-500	Tri-TSP-20	Tri-TSP-50	Tri-TSP-100	MO-Portfolio
$\mathcal{D}$(best)	**5.37**	**5.11**	**4.93**	**3.00**	**3.45**	**4.68**	**5.05**	**4.89**	**4.55**	**4.52**	**11.88**	**9.82**	9.36	**3.78**
End-to-End	3.48 ± 0.08	3.30 ± 0.22	4.14 ± 0.02	2.40 ± 0.03	1.87 ± 0.00	3.69 ± 0.00	2.54 ± 0.22	4.12 ± 0.01	3.83 ± 0.58	3.35 ± 0.01	4.43 ± 0.35	3.69 ± 0.04	6.27 ± 0.56	2.97 ± 0.14
End-to-End + GradNorm	3.26 ± 0.05	2.93 ± 0.15	2.95 ± 0.02	2.23 ± 0.30	2.35 ± 0.22	1.90 ± 0.00	2.26 ± 0.10	3.21 ± 0.27	3.03 ± 0.18	4.19 ± 0.06	4.84 ± 0.36	4.66 ± 0.30	8.02 ± 0.11	3.14 ± 0.14
End-to-End + PcGrad	3.15 ± 0.07	3.39 ± 0.05	3.14 ± 0.24	2.02 ± 0.05	1.95 ± 0.04	2.35 ± 0.00	2.29 ± 0.03	3.86 ± 0.24	3.32 ± 0.02	3.65 ± 0.09	7.04 ± 1.00	5.06 ± 0.93	8.50 ± 0.48	1.99 ± 0.27
Multi-Head	3.09 ± 0.11	3.21 ± 0.01	3.84 ± 0.26	2.45 ± 0.01	1.75 ± 0.04	2.49 ± 0.00	2.45 ± 0.07	4.00 ± 0.08	3.25 ± 0.05	3.04 ± 0.09	3.90 ± 0.14	4.22 ± 0.14	6.29 ± 0.94	2.02 ± 0.22
Multi-Head + GradNorm	3.10 ± 0.14	3.12 ± 0.22	2.64 ± 0.16	2.21 ± 0.02	2.18 ± 0.17	2.20 ± 0.17	3.23 ± 0.05	2.80 ± 0.04	2.92 ± 0.11	3.18 ± 0.03	4.50 ± 0.09	5.43 ± 0.47	6.52 ± 0.51	3.06 ± 0.09
Multi-Head + PcGrad	2.97 ± 0.15	3.11 ± 0.03	2.97 ± 0.03	2.33 ± 0.03	2.06 ± 0.14	2.35 ± 0.00	2.24 ± 0.02	3.20 ± 0.05	3.02 ± 0.19	2.27 ± 0.78	6.97 ± 0.02	4.75 ± 0.27	$\mathbf{9.39 \pm 0.00}$	3.00 ± 0.05
Multiple Models	3.37 ± 0.03	3.38 ± 0.13	4.24 ± 0.03	2.42 ± 0.05	2.07 ± 0.01	2.47 ± 0.00	2.45 ± 0.09	2.48 ± 0.20	3.40 ± 0.19	4.34 ± 0.03	4.32 ± 0.13	4.63 ± 0.20	6.34 ± 0.42	3.66 ± 0.01
Multiple Models + COMs	3.89 ± 0.00	3.49 ± 0.14	3.27 ± 0.22	2.14 ± 0.03	2.10 ± 0.06	2.30 ± 0.00	3.06 ± 0.05	3.93 ± 0.05	3.35 ± 0.20	3.71 ± 0.13	7.18 ± 0.37	5.88 ± 0.92	7.31 ± 0.23	2.10 ± 0.08
Multiple Models + RoMA	3.23 ± 0.03	3.41 ± 0.08	2.95 ± 0.05	2.17 ± 0.13	2.01 ± 0.09	2.49 ± 0.03	2.90 ± 0.14	3.60 ± 0.20	3.84 ± 0.03	1.53 ± 0.03	6.08 ± 0.11	6.04 ± 0.07	7.67 ± 0.11	2.88 ± 0.03
Multiple Models + IOM	3.50 ± 0.22	3.54 ± 0.02	4.55 ± 0.14	2.15 ± 0.03	2.02 ± 0.02	2.45 ± 0.10	3.62 ± 0.30	4.54 ± 0.03	4.33 ± 0.03	4.01 ± 0.26	9.25 ± 0.08	9.07 ± 0.21	9.17 ± 0.05	2.88 ± 0.02
Multiple Models + ICT	3.16 ± 0.02	3.23 ± 0.20	3.19 ± 0.04	2.01 ± 0.15	2.03 ± 0.22	2.46 ± 0.11	2.67 ± 0.01	2.89 ± 0.11	2.97 ± 0.08	3.38 ± 0.10	4.72 ± 0.16	5.71 ± 0.04	5.96 ± 0.20	1.75 ± 0.30
Multiple Models + Tri-Mentoring	3.10 ± 0.06	3.03 ± 0.01	3.26 ± 0.20	2.25 ± 0.02	1.97 ± 0.00	2.35 ± 0.31	2.69 ± 0.03	2.98 ± 0.26	3.57 ± 0.16	3.31 ± 0.14	4.53 ± 0.30	5.74 ± 0.35	5.65 ± 0.18	2.50 ± 0.08
MOBO	2.77 ± 0.05	2.26 ± 0.14	1.42 ± 0.07	2.38 ± 0.05	1.97 ± 0.12	2.04 ± 0.07	2.14 ± 0.24	1.61 ± 0.05	1.51 ± 0.05	N/A	3.47 ± 0.16	2.69 ± 0.03	2.20 ± 0.07	2.89 ± 0.01
MOBO-qParEGO	2.79 ± 0.01	2.21 ± 0.01	1.20 ± 0.01	2.15 ± 0.01	2.05 ± 0.00	N/A	2.18 ± 0.07	1.71 ± 0.01	1.52 ± 0.00	N/A	3.52 ± 0.09	2.70 ± 0.02	2.11 ± 0.02	2.90 ± 0.06
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3.15 ± 0.21

Table 19. Hypervolume results for scientific design with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Molecule	Regex	RFP	ZINC
$\mathcal{D}$(best)	**2.26**	3.05	3.75	**4.06**
End-to-End	1.07 ± 0.07	2.05 ± 0.00	3.64 ± 0.05	3.95 ± 0.04
End-to-End + GradNorm	1.07 ± 0.07	2.05 ± 0.00	3.73 ± 0.04	3.92 ± 0.00
End-to-End + PcGrad	2.12 ± 0.04	2.05 ± 0.00	3.70 ± 0.05	3.89 ± 0.06
Multi-Head	2.08 ± 0.00	2.05 ± 0.00	3.74 ± 0.00	3.86 ± 0.02
Multi-Head + GradNorm	1.00 ± 0.00	2.05 ± 0.00	3.69 ± 0.01	3.82 ± 0.01
Multi-Head + PcGrad	1.00 ± 0.00	2.05 ± 0.00	3.68 ± 0.02	3.86 ± 0.01
Multiple Models	1.10 ± 0.09	2.05 ± 0.00	3.70 ± 0.01	3.84 ± 0.00
Multiple Models + COMs	1.76 ± 0.14	2.38 ± 0.33	3.70 ± 0.00	3.86 ± 0.02
Multiple Models + RoMA	1.03 ± 0.00	2.05 ± 0.00	$\mathbf{3.79 \pm 0.04}$	3.91 ± 0.02
Multiple Models + IOM	1.02 ± 0.01	2.05 ± 0.00	$\mathbf{3.76 \pm 0.03}$	3.91 ± 0.02
Multiple Models + ICT	1.02 ± 0.02	2.05 ± 0.00	3.67 ± 0.00	3.96 ± 0.07
Multiple Models + Tri-Mentoring	1.41 ± 0.17	2.05 ± 0.00	3.75 ± 0.03	3.75 ± 0.00
MOBO	1.02 ± 0.02	$\mathbf{3.42 \pm 0.25}$	3.70 ± 0.01	3.90 ± 0.01
MOBO-qParEGO	1.96 ± 0.12	$\mathbf{3.17 \pm 0.11}$	3.33 ± 0.00	4.00 ± 0.03
MOBO-JES	1.00 ± 0.00	N/A	N/A	N/A

Table 20. Hypervolume results for RE with 256 solutions and 50th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	4.23	4.78	4.75	4.59	4.79	10.23	10.53	10.59	48.06	10.96	7.57	4.72	36.17	12.53	135.87
End-to-End	4.42 ± 0.00	4.84 ± 0.00	4.84 ± 0.00	4.38 ± 0.00	4.73 ± 0.04	10.56 ± 0.00	10.64 ± 0.00	**10.68 ± 0.00**	50.27 ± 0.40	11.69 ± 0.00	8.86 ± 0.79	4.67 ± 0.35	43.75 ± 0.07	19.89 ± 0.05	**143.86 ± 0.14**
End-to-End + GradNorm	4.39 ± 0.02	4.84 ± 0.00	2.64 ± 0.00	4.29 ± 0.00	**4.84 ± 0.00**	10.65 ± 0.00	10.61 ± 0.00	9.72 ± 0.03	47.25 ± 3.23	11.47 ± 0.01	5.74 ± 0.00	6.02 ± 0.07	40.63 ± 1.01	13.46 ± 0.00	137.59 ± 3.59
End-to-End + PcGrad	**4.61 ± 0.32**	4.52 ± 0.32	**4.84 ± 0.00**	4.22 ± 0.02	4.35 ± 0.00	10.65 ± 0.00	10.64 ± 0.00	9.86 ± 0.36	52.17 ± 0.55	11.67 ± 0.01	9.46 ± 0.03	4.00 ± 0.18	41.87 ± 0.01	15.81 ± 1.68	140.21 ± 2.41
Multi-Head	4.56 ± 0.14	4.83 ± 0.01	4.59 ± 0.10	4.11 ± 0.01	3.82 ± 0.30	10.64 ± 0.00	10.64 ± 0.00	**10.47 ± 0.22**	51.24 ± 0.12	11.73 ± 0.00	6.76 ± 0.00	4.43 ± 0.01	43.62 ± 0.02	20.20 ± 0.15	140.38 ± 0.56
Multi-Head + GradNorm	4.84 ± 0.01	3.75 ± 0.06	3.70 ± 0.09	2.64 ± 0.00	3.14 ± 0.01	10.65 ± 0.00	10.62 ± 0.01	6.12 ± 0.49	52.50 ± 0.14	11.52 ± 0.00	0.02 ± 0.00	5.90 ± 0.44	43.41 ± 0.16	15.25 ± 1.79	142.41 ± 0.63
Multi-Head + PcGrad	4.79 ± 0.03	4.84 ± 0.00	3.42 ± 0.57	3.77 ± 0.00	4.35 ± 0.00	7.64 ± 0.00	10.08 ± 0.00	10.11 ± 0.35	52.48 ± 0.05	11.50 ± 0.00	9.64 ± 0.00	6.32 ± 0.05	43.60 ± 0.02	**19.69 ± 0.96**	139.81 ± 2.17
Multiple Models	**4.93 ± 0.00**	4.84 ± 0.00	4.84 ± 0.00	4.79 ± 0.01	4.83 ± 0.01	10.63 ± 0.00	10.63 ± 0.00	9.62 ± 0.62	**54.02 ± 0.00**	11.65 ± 0.01	**10.31 ± 0.03**	**6.45 ± 0.01**	42.97 ± 0.42	**20.55 ± 0.01**	143.59 ± 0.17
Multiple Models + COMs	3.90 ± 0.10	4.83 ± 0.00	4.76 ± 0.02	4.59 ± 0.00	4.84 ± 0.00	5.28 ± 5.28	10.62 ± 0.00	10.26 ± 0.31	48.14 ± 2.03	11.41 ± 0.02	8.18 ± 0.15	5.68 ± 0.20	39.96 ± 0.64	13.00 ± 0.46	138.74 ± 0.89
Multiple Models + RoMA	4.88 ± 0.00	4.84 ± 0.00	4.83 ± 0.00	3.66 ± 0.01	3.40 ± 0.01	10.60 ± 0.00	10.64 ± 0.00	10.11 ± 0.05	50.37 ± 0.46	**11.76 ± 0.01**	3.76 ± 0.05	6.37 ± 0.04	43.33 ± 0.04	17.14 ± 0.23	131.51 ± 4.58
Multiple Models + IOM	**4.93 ± 0.01**	4.84 ± 0.00	4.81 ± 0.02	4.28 ± 0.01	4.14 ± 0.01	**10.65 ± 0.00**	**10.65 ± 0.00**	10.64 ± 0.03	53.83 ± 0.06	11.68 ± 0.05	9.33 ± 0.09	6.33 ± 0.08	42.93 ± 0.11	**20.54 ± 0.02**	141.76 ± 0.13
Multiple Models + ICT	4.70 ± 0.15	4.84 ± 0.00	2.76 ± 0.00	3.23 ± 0.00	4.74 ± 0.00	10.62 ± 0.01	2.77 ± 0.00	9.80 ± 0.50	53.26 ± 0.07	11.68 ± 0.04	8.00 ± 0.16	6.14 ± 0.09	43.33 ± 0.15	16.04 ± 0.41	142.65 ± 0.50
Multiple Models + Tri-Mentoring	4.89 ± 0.01	4.84 ± 0.00	2.76 ± 0.00	**4.81 ± 0.01**	4.70 ± 0.00	10.65 ± 0.00	**10.65 ± 0.00**	10.54 ± 0.00	50.66 ± 0.02	11.75 ± 0.00	8.47 ± 0.15	6.35 ± 0.07	43.25 ± 0.23	18.95 ± 0.91	143.33 ± 0.31
MOBO	4.31 ± 0.05	**4.84 ± 0.00**	4.18 ± 0.01	3.32 ± 0.02	4.83 ± 0.00	10.03 ± 0.00	10.53 ± 0.12	10.48 ± 0.02	43.92 ± 2.75	11.42 ± 0.07	0.00 ± 0.00	**6.40 ± 0.08**	**45.03 ± 0.60**	12.08 ± 0.00	N/A
MOBO-qParEGO	4.44 ± 0.15	4.21 ± 0.40	4.75 ± 0.01	0.00 ± 0.00	4.12 ± 0.29	5.31 ± 5.31	8.82 ± 0.37	10.46 ± 0.09	43.07 ± 0.74	0.00 ± 0.00	0.00 ± 0.00	5.52 ± 0.04	N/A	N/A	N/A
MOBO-JES	3.89 ± 0.03	4.57 ± 0.03	4.66 ± 0.05	4.54 ± 0.00	4.80 ± 0.00	10.01 ± 0.01	10.63 ± 0.01	10.52 ± 0.03	48.52 ± 0.00	11.12 ± 0.03	6.46 ± 0.34	5.24 ± 0.17	N/A	N/A	N/A

Table 21. Average rank of different offline MOO methods on each type of task with 32 solutions and 100th percentile evaluations, where the best and runner-up results are **bolded** and underlined, respectively.

Methods	Synthetic	MO-NAS	MORL	MOCO	Sci-Design	RE	Average Rank
$\mathcal{D}$(best)	12.03 ± 0.16	10.58 ± 0.32	4.00 ± 0.00	**1.07 ± 0.00**	4.25 ± 0.25	13.20 ± 0.07	8.95 ± 0.12
End-to-End	7.03 ± 0.09	7.35 ± 0.13	7.00 ± 1.00	5.68 ± 0.68	**3.81 ± 0.31**	8.47 ± 0.13	6.95 ± 0.13
End-to-End + GradNorm	7.80 ± 0.02	8.29 ± 0.24	6.00 ± 2.00	7.82 ± 0.18	12.25 ± 0.25	10.43 ± 0.50	8.71 ± 0.00
End-to-End + PcGrad	8.05 ± 0.67	7.69 ± 0.52	4.00 ± 0.00	6.99 ± 0.15	6.52 ± 0.35	8.30 ± 0.03	7.63 ± 0.34
Multi-Head	6.48 ± 0.33	6.55 ± 0.76	11.25 ± 0.75	7.88 ± 0.05	6.23 ± 0.40	8.40 ± 0.00	7.30 ± 0.13
Multi-Head + GradNorm	8.14 ± 0.27	9.42 ± 0.32	10.00 ± 0.50	10.50 ± 0.43	13.25 ± 0.50	10.10 ± 0.70	9.71 ± 0.42
Multi-Head + PcGrad	8.52 ± 0.30	7.76 ± 0.24	9.50 ± 2.00	8.39 ± 0.61	10.75 ± 0.12	7.93 ± 0.13	8.31 ± 0.32
Multiple Models	**5.23 ± 0.14**	4.89 ± 0.53	11.25 ± 0.75	5.42 ± 0.73	10.75 ± 0.25	5.65 ± 0.08	5.74 ± 0.02
Multiple Models + COMs	8.88 ± 0.06	**4.95 ± 0.27**	**1.25 ± 0.25**	6.11 ± 0.25	10.31 ± 0.94	9.95 ± 0.02	7.29 ± 0.07
Multiple Models + RoMA	9.20 ± 0.27	5.24 ± 0.29	5.50 ± 1.00	8.82 ± 0.89	8.88 ± 0.00	7.40 ± 0.40	7.55 ± 0.36
Multiple Models + IOM	6.56 ± 0.09	5.39 ± 0.23	7.50 ± 1.00	3.88 ± 0.05	6.06 ± 0.69	**4.10 ± 0.10**	**5.12 ± 0.05**
Multiple Models + ICT	8.08 ± 0.36	12.35 ± 0.26	6.75 ± 0.25	10.18 ± 0.18	5.75 ± 1.38	7.90 ± 0.10	9.28 ± 0.01
Multiple Models + Tri-Mentoring	6.27 ± 0.61	11.35 ± 1.31	11.50 ± 0.00	10.46 ± 0.11	7.88 ± 0.00	6.63 ± 0.30	8.74 ± 0.11
MOBO	10.41 ± 0.09	5.37 ± 0.37	9.50 ± 0.50	13.11 ± 0.68	5.75 ± 0.00	7.89 ± 0.04	8.76 ± 0.02
MOBO-qParEGO	10.52 ± 0.39	10.32 ± 0.11	N/A	12.00 ± 0.25	6.94 ± 0.19	8.54 ± 0.04	10.14 ± 0.08
MOBO-JES	12.27 ± 0.06	14.50 ± 0.50	N/A	2.00 ± 0.00	7.50 ± 5.50	7.75 ± 0.17	9.99 ± 0.31

Table 22. Hypervolume results for synthetic functions with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	OmniTest	VLMOP1	VLMOP2	VLMOP3	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	10.43	9.43	9.71	**10.76**	9.06	8.20	8.32	3.87	0.08	1.64	45.14	4.04	4.70	5.05	**5.46**	4.76
End-to-End	10.64 ± 0.00	**10.65 ± 0.00**	10.64 ± 0.00	**10.76 ± 0.00**	**10.64 ± 0.01**	**10.62 ± 0.03**	10.60 ± 0.09	4.35 ± 0.00	2.54 ± 2.25	4.05 ± 0.05	46.90 ± 0.01	2.70 ± 0.00	3.19 ± 0.01	5.33 ± 0.03	3.72 ± 0.36	**4.90 ± 0.00**
End-to-End + GradNorm	10.63 ± 0.01	10.64 ± 0.01	**10.65 ± 0.00**	**10.76 ± 0.00**	**10.54 ± 0.09**	10.64 ± 0.00	**10.71 ± 0.00**	3.76 ± 0.03	1.26 ± 1.08	2.79 ± 1.34	42.23 ± 0.98	4.77 ± 0.01	5.63 ± 0.02	5.22 ± 0.02	3.23 ± 0.03	4.73 ± 0.11
End-to-End + PcGrad	10.63 ± 0.01	10.57 ± 0.04	**10.65 ± 0.00**	**10.76 ± 0.00**	9.02 ± 0.10	9.45 ± 0.15	10.52 ± 0.00	4.35 ± 0.00	4.80 ± 0.02	4.06 ± 0.01	46.79 ± 0.06	**4.84 ± 0.01**	5.70 ± 0.01	5.45 ± 0.00	3.12 ± 0.01	2.04 ± 0.22
Multi-Head	10.37 ± 0.26	**10.65 ± 0.00**	10.64 ± 0.00	**10.76 ± 0.00**	**10.61 ± 0.03**	10.65 ± 0.00	10.52 ± 0.00	4.24 ± 0.00	**2.57 ± 2.25**	4.08 ± 0.01	46.93 ± 0.00	2.72 ± 0.00	**5.72 ± 0.00**	5.41 ± 0.10	3.70 ± 0.23	4.87 ± 0.01
Multi-Head + GradNorm	10.62 ± 0.01	10.63 ± 0.01	10.64 ± 0.00	**10.76 ± 0.00**	9.29 ± 0.86	10.62 ± 0.02	**10.61 ± 0.10**	4.33 ± 0.00	0.00 ± 0.00	4.13 ± 0.03	46.64 ± 0.22	4.83 ± 0.00	**5.68 ± 0.05**	5.26 ± 0.04	3.39 ± 0.00	4.87 ± 0.00
Multi-Head + PcGrad	10.63 ± 0.00	10.63 ± 0.00	10.61 ± 0.03	**10.76 ± 0.00**	9.08 ± 0.35	10.59 ± 0.01	10.49 ± 0.01	4.35 ± 0.00	4.80 ± 0.01	3.99 ± 0.01	46.91 ± 0.00	2.72 ± 0.04	**5.69 ± 0.03**	5.45 ± 0.00	3.64 ± 0.17	2.05 ± 0.18
Multiple Models	10.61 ± 0.01	10.64 ± 0.00	**10.65 ± 0.00**	**10.76 ± 0.00**	**10.63 ± 0.01**	**10.65 ± 0.00**	10.67 ± 0.03	4.35 ± 0.00	2.52 ± 2.21	4.09 ± 0.02	**46.97 ± 0.00**	4.80 ± 0.03	5.51 ± 0.01	5.51 ± 0.14	4.26 ± 0.07	4.88 ± 0.00
Multiple Models + COMs	10.63 ± 0.00	10.48 ± 0.08	10.55 ± 0.04	**10.76 ± 0.00**	9.14 ± 0.44	9.50 ± 0.15	9.14 ± 0.10	**4.41 ± 0.00**	4.78 ± 0.01	**4.87 ± 0.00**	46.81 ± 0.12	4.21 ± 0.07	4.85 ± 0.02	5.25 ± 0.24	3.99 ± 0.10	4.59 ± 0.02
Multiple Models + RoMA	10.63 ± 0.01	10.57 ± 0.03	10.64 ± 0.00	**10.76 ± 0.00**	9.10 ± 0.48	10.60 ± 0.00	10.12 ± 0.05	3.97 ± 0.06	4.78 ± 0.01	1.46 ± 0.00	44.15 ± 2.36	**4.83 ± 0.01**	5.65 ± 0.00	**5.80 ± 0.00**	3.18 ± 0.05	1.84 ± 0.07
Multiple Models + IOM	10.62 ± 0.01	10.56 ± 0.08	10.46 ± 0.02	**10.76 ± 0.00**	10.08 ± 0.04	10.15 ± 0.45	10.48 ± 0.00	**4.41 ± 0.00**	4.79 ± 0.00	**3.29 ± 1.58**	**46.97 ± 0.00**	4.73 ± 0.00	5.45 ± 0.02	5.52 ± 0.05	4.84 ± 0.00	4.87 ± 0.02
Multiple Models + ICT	10.63 ± 0.00	10.62 ± 0.01	**10.56 ± 0.09**	**10.76 ± 0.00**	7.98 ± 0.53	10.61 ± 0.01	10.54 ± 0.01	**4.41 ± 0.00**	0.31 ± 0.00	1.46 ± 0.00	**46.97 ± 0.00**	4.25 ± 0.15	5.54 ± 0.01	4.35 ± 0.08	4.84 ± 0.00	2.79 ± 0.56
Multiple Models + Tri-Mentoring	10.63 ± 0.01	10.65 ± 0.00	10.63 ± 0.01	**10.76 ± 0.00**	9.48 ± 0.20	10.62 ± 0.02	10.58 ± 0.08	**4.41 ± 0.00**	**4.82 ± 0.00**	1.46 ± 0.00	**46.97 ± 0.00**	4.51 ± 0.01	5.55 ± 0.01	5.59 ± 0.06	4.84 ± 0.00	2.31 ± 0.01
MOBO	**10.64 ± 0.00**	9.82 ± 0.23	9.68 ± 0.19	10.24 ± 0.00	8.83 ± 0.19	8.42 ± 0.16	10.34 ± 0.01	4.34 ± 0.00	0.30 ± 0.00	2.73 ± 0.02	46.92 ± 0.00	4.30 ± 0.00	5.08 ± 0.00	5.26 ± 0.01	4.52 ± 0.10	3.20 ± 0.00
MOBO-qParEGO	10.63 ± 0.00	9.56 ± 0.02	9.43 ± 0.00	**10.76 ± 0.00**	8.48 ± 0.03	8.43 ± 0.00	10.17 ± 0.00	4.33 ± 0.00	0.26 ± 0.02	3.34 ± 0.33	46.92 ± 0.01	4.25 ± 0.02	5.22 ± 0.06	5.20 ± 0.01	4.85 ± 0.02	3.36 ± 0.00
MOBO-JES	10.61 ± 0.00	10.22 ± 0.08	10.23 ± 0.18	8.56 ± 0.07	9.67 ± 0.01	9.62 ± 0.04	9.36 ± 0.08	3.87 ± 0.00	N/A	1.46 ± 0.00	46.88 ± 0.00	3.97 ± 0.09	4.44 ± 0.07	5.17 ± 0.02	4.43 ± 0.08	3.09 ± 0.02

Table 23. Hypervolume results for MO-NAS with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9	NasBench201-Test
$\mathcal{D}$(best)	4.78	10.48	9.72	21.15	40.51	92.43	358.27	4.55	10.59	4.97	5.00	11.21	16.63	17.26	44.43	4.69	11.42	21.50	10.07
End-to-End	4.82 ± 0.00	10.22 ± 0.26	10.63 ± 0.00	24.21 ± 0.12	49.48 ± 0.00	109.05 ± 0.00	**512.31 ± 1.40**	4.61 ± 0.07	10.68 ± 0.00	4.80 ± 0.02	5.11 ± 0.00	10.93 ± 0.06	16.88 ± 0.04	17.68 ± 0.03	45.39 ± 0.33	5.07 ± 0.07	11.07 ± 0.04	22.38 ± 0.37	10.28 ± 0.01
End-to-End + GradNorm	4.83 ± 0.00	10.49 ± 0.03	10.45 ± 0.06	24.96 ± 0.30	**50.35 ± 0.02**	108.57 ± 0.41	508.72 ± 0.60	4.55 ± 0.01	10.26 ± 0.17	5.14 ± 0.00	4.88 ± 0.03	10.62 ± 0.07	**16.87 ± 0.16**	17.49 ± 0.25	44.38 ± 0.63	4.97 ± 0.00	10.44 ± 0.06	22.71 ± 0.19	9.39 ± 0.32
End-to-End + PcGrad	4.83 ± 0.00	10.50 ± 0.00	10.53 ± 0.06	25.30 ± 0.15	50.17 ± 0.01	108.22 ± 0.52	501.11 ± 0.00	4.34 ± 0.18	**10.93 ± 0.27**	5.01 ± 0.17	5.05 ± 0.01	11.14 ± 0.13	16.16 ± 0.61	17.44 ± 0.04	44.07 ± 0.26	4.93 ± 0.00	10.95 ± 0.01	23.39 ± 0.00	**10.32 ± 0.02**
Multi-Head	4.83 ± 0.01	10.53 ± 0.01	9.59 ± 0.00	**25.78 ± 0.15**	49.77 ± 0.07	109.47 ± 0.83	493.13 ± 4.07	4.31 ± 0.04	9.73 ± 0.40	5.15 ± 0.03	4.84 ± 0.04	11.35 ± 0.04	16.79 ± 0.10	**17.80 ± 0.02**	44.91 ± 0.45	5.15 ± 0.07	11.26 ± 0.17	24.00 ± 0.06	**10.39 ± 0.09**
Multi-Head + GradNorm	**4.84 ± 0.00**	10.54 ± 0.00	9.59 ± 0.01	23.47 ± 0.12	49.49 ± 0.62	107.76 ± 0.09	499.69 ± 3.16	4.41 ± 0.21	10.19 ± 0.61	4.66 ± 0.02	4.37 ± 0.02	10.40 ± 0.24	16.80 ± 0.09	17.74 ± 0.01	45.10 ± 0.00	4.90 ± 0.07	8.11 ± 0.27	17.78 ± 0.01	9.99 ± 0.09
Multi-Head + PcGrad	**4.84 ± 0.00**	**10.55 ± 0.00**	10.64 ± 0.06	23.50 ± 1.02	49.62 ± 0.09	108.33 ± 1.09	480.47 ± 0.00	4.28 ± 0.08	10.46 ± 0.00	4.90 ± 0.00	**5.16 ± 0.01**	11.32 ± 0.05	**16.91 ± 0.14**	17.45 ± 0.18	44.78 ± 0.05	5.25 ± 0.07	10.32 ± 0.35	21.32 ± 0.14	10.18 ± 0.15
Multiple Models	4.83 ± 0.01	10.53 ± 0.02	10.67 ± 0.03	25.56 ± 0.01	49.96 ± 0.00	110.94 ± 0.29	509.96 ± 1.58	**4.69 ± 0.17**	10.38 ± 0.32	5.13 ± 0.04	5.07 ± 0.03	11.26 ± 0.02	16.84 ± 0.11	17.70 ± 0.04	**45.82 ± 0.14**	**5.30 ± 0.09**	11.40 ± 0.22	23.48 ± 0.51	10.28 ± 0.00
Multiple Models + COMs	4.82 ± 0.00	10.53 ± 0.00	10.59 ± 0.00	24.66 ± 0.12	49.96 ± 0.14	110.40 ± 0.19	507.19 ± 2.53	**4.77 ± 0.01**	10.77 ± 0.03	5.19 ± 0.04	5.11 ± 0.00	11.33 ± 0.01	16.84 ± 0.04	17.59 ± 0.07	45.20 ± 0.20	**5.40 ± 0.07**	11.93 ± 0.06	25.18 ± 0.21	10.27 ± 0.06
Multiple Models + RoMA	4.81 ± 0.01	10.51 ± 0.00	10.56 ± 0.00	24.89 ± 0.09	50.04 ± 0.07	108.37 ± 0.28	506.10 ± 2.55	4.37 ± 0.04	10.20 ± 0.00	5.23 ± 0.02	**5.17 ± 0.01**	**11.39 ± 0.02**	**16.97 ± 0.06**	**17.77 ± 0.05**	45.35 ± 0.03	**5.37 ± 0.03**	**12.11 ± 0.07**	**25.48 ± 0.05**	10.03 ± 0.03
Multiple Models + IOM	4.82 ± 0.00	10.44 ± 0.01	10.65 ± 0.04	24.55 ± 0.00	50.23 ± 0.06	111.25 ± 0.08	493.81 ± 15.32	**4.78 ± 0.02**	10.80 ± 0.07	5.15 ± 0.01	5.07 ± 0.00	**11.40 ± 0.00**	16.57 ± 0.08	17.71 ± 0.02	**45.73 ± 0.11**	5.03 ± 0.04	12.01 ± 0.05	24.94 ± 0.33	10.13 ± 0.16
Multiple Models + ICT	4.06 ± 0.00	10.51 ± 0.01	10.01 ± 0.03	24.16 ± 0.00	48.33 ± 0.35	104.07 ± 1.13	477.25 ± 1.44	3.65 ± 0.00	7.93 ± 0.22	4.71 ± 0.06	4.65 ± 0.02	9.96 ± 0.05	16.23 ± 0.00	16.03 ± 0.00	45.34 ± 0.14	4.59 ± 0.39	9.53 ± 0.07	20.68 ± 0.03	9.09 ± 0.19
Multiple Models + Tri-Mentoring	4.06 ± 0.00	7.76 ± 0.00	9.48 ± 0.17	23.83 ± 0.10	49.02 ± 0.00	110.23 ± 0.25	481.97 ± 1.83	3.27 ± 0.13	8.22 ± 0.00	4.97 ± 0.19	4.90 ± 0.17	9.82 ± 0.00	15.61 ± 0.00	15.88 ± 0.00	44.90 ± 0.00	4.63 ± 0.51	10.66 ± 0.04	21.85 ± 1.10	8.09 ± 2.16
MOBO	4.61 ± 0.08	10.48 ± 0.03	**10.79 ± 0.00**	**25.87 ± 0.14**	50.24 ± 0.02	**111.83 ± 0.40**	**513.76 ± 2.91**	4.75 ± 0.00	**11.03 ± 0.08**	**5.35 ± 0.02**	**5.16 ± 0.01**	11.18 ± 0.02	16.70 ± 0.09	17.51 ± 0.07	44.01 ± 0.41	5.24 ± 0.05	11.53 ± 0.13	22.64 ± 0.89	10.30 ± 0.06
MOBO-qParEGO	4.78 ± 0.01	10.44 ± 0.00	8.94 ± 0.05	20.01 ± 0.05	37.21 ± 0.00	94.72 ± 5.91	350.55 ± 0.13	4.50 ± 0.00	10.83 ± 0.13	4.93 ± 0.08	4.70 ± 0.01	10.34 ± 0.16	16.72 ± 0.14	17.61 ± 0.07	45.27 ± 0.31	5.11 ± 0.08	11.39 ± 0.27	23.25 ± 0.25	9.43 ± 0.02
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	9.16 ± 0.07

Table 24. Hypervolume results for MORL with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	4.21	**2.85**
End-to-End	4.04 ± 0.24	2.56 ± 0.13
End-to-End + GradNorm	4.42 ± 0.36	2.61 ± 0.03
End-to-End + PcGrad	4.51 ± 0.27	2.71 ± 0.05
Multi-Head	3.81 ± 0.04	2.36 ± 0.13
Multi-Head + GradNorm	3.71 ± 0.03	2.59 ± 0.04
Multi-Head + PcGrad	4.40 ± 0.24	2.18 ± 0.19
Multiple Models	3.87 ± 0.00	2.28 ± 0.11
Multiple Models + COMs	$\mathbf{4.90 \pm 0.06}$	$\mathbf{2.85 \pm 0.04}$
Multiple Models + RoMA	4.18 ± 0.03	$\mathbf{2.67 \pm 0.18}$
Multiple Models + IOM	$\mathbf{4.63 \pm 0.33}$	2.23 ± 0.02
Multiple Models + ICT	3.95 ± 0.03	2.69 ± 0.02
Multiple Models + Tri-Mentoring	3.70 ± 0.01	2.41 ± 0.02
MOBO	4.20 ± 0.01	2.27 ± 0.10
MOBO-qParEGO	N/A	N/A
MOBO-JES	N/A	N/A

Table 25. Hypervolume results for MOCO with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Bi-CVRP-20	Bi-CVRP-50	Bi-CVRP-100	Bi-KP-50	Bi-KP-100	Bi-KP-200	Bi-TSP-20	Bi-TSP-50	Bi-TSP-100	Bi-TSP-500	Tri-TSP-20	Tri-TSP-50	Tri-TSP-100	Portfolio
$\mathcal{D}$(best)	**5.37**	**5.11**	**4.93**	**3.00**	**3.45**	**4.68**	**5.05**	**4.89**	**4.55**	**4.52**	**11.88**	**9.82**	**9.36**	**3.78**
End-to-End	5.05 ± 0.17	4.30 ± 0.12	4.53 ± 0.36	2.70 ± 0.04	2.33 ± 0.30	3.37 ± 0.00	3.44 ± 0.03	4.68 ± 0.12	4.48 ± 0.02	$\mathbf{4.48 \pm 0.05}$	6.87 ± 1.32	5.56 ± 0.70	9.07 ± 0.04	2.95 ± 0.01
End-to-End + GradNorm	4.02 ± 0.27	3.75 ± 0.21	3.63 ± 0.00	2.67 ± 0.01	2.67 ± 0.02	2.69 ± 0.00	3.06 ± 0.30	3.87 ± 0.13	3.53 ± 0.00	4.11 ± 0.16	5.61 ± 0.01	6.29 ± 0.00	8.47 ± 0.12	3.13 ± 0.00
End-to-End + PcGrad	3.59 ± 0.17	3.52 ± 0.40	3.63 ± 0.12	2.75 ± 0.00	2.58 ± 0.22	2.84 ± 0.15	3.14 ± 0.24	4.49 ± 0.02	4.18 ± 0.17	3.98 ± 0.02	9.31 ± 0.08	8.11 ± 0.00	8.54 ± 0.18	3.02 ± 0.00
Multi-Head	4.65 ± 0.00	3.75 ± 0.12	4.88 ± 0.01	2.73 ± 0.03	2.45 ± 0.14	2.99 ± 0.24	3.10 ± 0.14	4.50 ± 0.02	3.48 ± 0.32	3.19 ± 0.04	5.14 ± 0.05	6.65 ± 1.16	8.34 ± 0.32	2.98 ± 0.16
Multi-Head + GradNorm	3.77 ± 0.33	3.21 ± 0.07	3.04 ± 0.17	2.29 ± 0.16	2.24 ± 0.22	1.64 ± 0.00	3.69 ± 0.08	3.23 ± 0.01	3.27 ± 0.17	3.18 ± 0.08	5.96 ± 0.47	5.53 ± 0.30	6.73 ± 0.00	3.11 ± 0.11
Multi-Head + PcGrad	4.22 ± 0.23	3.53 ± 0.73	3.09 ± 0.05	2.81 ± 0.00	2.35 ± 0.12	3.54 ± 0.00	3.78 ± 1.21	2.91 ± 0.15	3.26 ± 0.08	3.03 ± 0.20	6.93 ± 1.64	6.73 ± 0.27	8.67 ± 0.20	3.09 ± 0.13
Multiple Models	5.01 ± 0.27	4.64 ± 0.13	4.62 ± 0.27	2.84 ± 0.10	2.73 ± 0.27	3.63 ± 0.58	2.71 ± 0.03	4.12 ± 0.54	4.23 ± 0.00	4.45 ± 0.00	6.36 ± 1.17	7.16 ± 0.68	8.65 ± 0.68	3.16 ± 0.06
Multiple Models + COMs	5.28 ± 0.00	4.21 ± 0.23	3.52 ± 0.24	2.66 ± 0.02	2.73 ± 0.07	2.79 ± 0.38	4.60 ± 0.15	4.63 ± 0.07	3.96 ± 0.24	3.76 ± 0.15	10.28 ± 0.86	8.07 ± 0.15	8.11 ± 0.02	2.20 ± 0.02
Multiple Models + RoMA	4.63 ± 0.11	3.55 ± 0.03	3.03 ± 0.20	2.57 ± 0.08	2.51 ± 0.33	2.38 ± 0.13	4.14 ± 0.11	3.72 ± 0.42	3.85 ± 0.23	1.76 ± 0.07	8.40 ± 0.76	5.89 ± 0.72	7.85 ± 0.69	2.92 ± 0.02
Multiple Models + IOM	5.28 ± 0.00	5.11 ± 0.00	4.92 ± 0.00	2.64 ± 0.12	2.68 ± 0.23	3.39 ± 0.22	4.86 ± 0.01	4.70 ± 0.01	4.46 ± 0.06	4.00 ± 0.36	9.79 ± 0.09	8.91 ± 0.24	$\mathbf{9.01 \pm 0.37}$	2.93 ± 0.00
Multiple Models + ICT	3.55 ± 0.17	4.01 ± 0.01	4.01 ± 0.37	2.36 ± 0.15	2.12 ± 0.00	2.31 ± 0.15	3.37 ± 0.21	3.09 ± 0.18	3.13 ± 0.07	4.11 ± 0.00	5.60 ± 0.15	6.12 ± 0.25	5.85 ± 0.20	2.05 ± 0.10
Multiple Models + Tri-Mentoring	3.48 ± 0.20	3.82 ± 0.05	3.65 ± 0.00	2.04 ± 0.04	2.44 ± 0.11	2.45 ± 0.31	3.19 ± 0.20	2.99 ± 0.33	3.31 ± 0.10	3.34 ± 0.05	5.85 ± 0.97	6.15 ± 0.44	5.70 ± 0.19	2.63 ± 0.12
MOBO	3.19 ± 0.38	2.50 ± 0.04	1.34 ± 0.02	2.54 ± 0.21	2.17 ± 0.14	2.08 ± 0.05	2.33 ± 0.08	1.81 ± 0.09	1.62 ± 0.01	1.26 ± 0.00	4.18 ± 0.37	3.20 ± 0.04	2.20 ± 0.09	3.03 ± 0.06
MOBO-qParEGO	3.45 ± 0.03	2.63 ± 0.00	1.46 ± 0.01	2.65 ± 0.11	2.43 ± 0.11	N/A	2.53 ± 0.01	1.96 ± 0.01	1.73 ± 0.04	N/A	4.69 ± 0.10	3.14 ± 0.01	2.52 ± 0.08	3.15 ± 0.04
MOBO-JES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3.53 ± 0.07

Table 26. Hypervolume results for scientific design with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	Molecule	Regex	RFP	ZINC
$\mathcal{D}$(best)	2.26	3.05	3.75	4.06
End-to-End	1.95 ± 0.13	2.80 ± 0.00	$\mathbf{4.34 \pm 0.03}$	$\mathbf{4.12 \pm 0.01}$
End-to-End + GradNorm	1.05 ± 0.00	2.38 ± 0.00	3.46 ± 0.08	4.00 ± 0.01
End-to-End + PcGrad	$\mathbf{3.33 \pm 0.00}$	2.47 ± 0.00	3.62 ± 0.00	4.04 ± 0.04
Multi-Head	2.20 ± 0.38	2.80 ± 0.00	3.67 ± 0.00	$\mathbf{4.12 \pm 0.00}$
Multi-Head + GradNorm	1.06 ± 0.05	2.38 ± 0.00	3.67 ± 0.00	3.82 ± 0.00
Multi-Head + PcGrad	1.04 ± 0.00	2.47 ± 0.00	3.71 ± 0.00	3.92 ± 0.00
Multiple Models	1.19 ± 0.14	2.38 ± 0.00	3.70 ± 0.00	4.01 ± 0.01
Multiple Models + COMs	2.16 ± 0.09	2.05 ± 0.00	3.65 ± 0.00	3.97 ± 0.10
Multiple Models + RoMA	1.84 ± 0.10	2.47 ± 0.00	3.71 ± 0.00	3.95 ± 0.00
Multiple Models + IOM	1.72 ± 0.36	2.47 ± 0.00	$\mathbf{4.06 \pm 0.35}$	4.08 ± 0.03
Multiple Models + ICT	$\mathbf{2.65 \pm 1.31}$	2.47 ± 0.00	3.76 ± 0.00	4.05 ± 0.02
Multiple Models + Tri-Mentoring	2.03 ± 0.00	2.47 ± 0.00	3.75 ± 0.00	3.94 ± 0.00
MOBO	1.15 ± 0.02	$\mathbf{4.52 \pm 0.18}$	3.69 ± 0.01	$\mathbf{4.18 \pm 0.08}$
MOBO-qParEGO	2.12 ± 0.04	4.26 ± 0.25	3.33 ± 0.00	4.05 ± 0.02
MOBO-JES	2.10 ± 1.04	N/A	N/A	N/A

Table 27. Hypervolume results for RE with 32 solutions and 100th percentile evaluations. For each task, algorithms within one standard deviation of having the highest performance are **bolded**.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	4.23	4.78	4.75	4.59	4.79	10.23	10.53	10.59	48.06	10.96	7.57	4.72	36.17	12.53	135.87
End-to-End	4.42 ± 0.00	4.84 ± 0.00	**4.84 ± 0.00**	4.38 ± 0.00	4.82 ± 0.02	10.58 ± 0.03	10.56 ± 0.08	10.68 ± 0.00	52.82 ± 0.05	11.61 ± 0.01	8.24 ± 0.00	6.20 ± 0.00	42.53 ± 0.05	17.09 ± 2.14	140.13 ± 1.42
End-to-End + GradNorm	4.81 ± 0.01	4.84 ± 0.00	2.64 ± 0.00	4.38 ± 0.00	**4.84 ± 0.00**	10.65 ± 0.00	10.63 ± 0.00	9.90 ± 0.00	51.98 ± 0.63	11.51 ± 0.00	8.63 ± 0.71	6.11 ± 0.10	39.21 ± 0.12	13.46 ± 0.00	140.24 ± 0.25
End-to-End + PcGrad	4.86 ± 0.01	4.84 ± 0.00	4.84 ± 0.00	4.40 ± 0.00	4.35 ± 0.00	**10.65 ± 0.00**	10.65 ± 0.00	10.41 ± 0.07	52.78 ± 0.04	11.66 ± 0.01	9.76 ± 0.11	5.52 ± 0.00	41.64 ± 0.31	15.92 ± 1.75	139.60 ± 0.28
Multi-Head	4.61 ± 0.13	4.84 ± 0.00	4.70 ± 0.04	4.78 ± 0.00	4.35 ± 0.00	10.65 ± 0.00	10.64 ± 0.00	10.65 ± 0.01	52.63 ± 0.22	11.70 ± 0.00	6.76 ± 0.00	5.72 ± 0.00	42.45 ± 0.40	19.30 ± 0.32	141.50 ± 0.03
Multi-Head + GradNorm	4.91 ± 0.00	4.65 ± 0.12	4.83 ± 0.00	3.51 ± 0.87	4.31 ± 0.53	10.65 ± 0.00	10.63 ± 0.00	5.82 ± 0.02	52.81 ± 0.00	11.51 ± 0.00	0.02 ± 0.00	6.26 ± 0.01	42.56 ± 0.11	15.82 ± 2.36	140.23 ± 0.66
Multi-Head + PcGrad	4.91 ± 0.01	4.84 ± 0.00	4.39 ± 0.01	**4.83 ± 0.00**	4.35 ± 0.00	7.66 ± 0.00	10.08 ± 0.00	10.52 ± 0.04	52.77 ± 0.01	11.71 ± 0.02	9.76 ± 0.18	6.34 ± 0.03	42.23 ± 0.17	20.19 ± 0.14	140.00 ± 0.12
Multiple Models	4.89 ± 0.01	4.84 ± 0.00	4.84 ± 0.00	4.82 ± 0.00	4.82 ± 0.00	10.65 ± 0.00	10.63 ± 0.00	10.61 ± 0.00	**53.58 ± 0.08**	11.61 ± 0.07	**10.38 ± 0.04**	6.30 ± 0.01	42.04 ± 0.28	**20.70 ± 0.98**	141.04 ± 0.60
Multiple Models + COMs	**5.19 ± 0.00**	4.82 ± 0.00	4.80 ± 0.00	4.59 ± 0.00	4.84 ± 0.00	10.56 ± 0.00	10.64 ± 0.00	10.39 ± 0.07	51.15 ± 0.13	11.57 ± 0.01	8.57 ± 0.20	5.85 ± 0.07	39.82 ± 0.10	21.13 ± 0.55	134.26 ± 1.43
Multiple Models + RoMA	4.81 ± 0.04	4.84 ± 0.00	4.84 ± 0.00	4.77 ± 0.05	4.69 ± 0.00	10.65 ± 0.00	10.65 ± 0.00	10.46 ± 0.04	52.07 ± 0.16	**11.74 ± 0.00**	8.95 ± 0.01	6.31 ± 0.01	41.87 ± 0.20	16.32 ± 0.06	139.91 ± 0.72
Multiple Models + IOM	**5.19 ± 0.00**	4.84 ± 0.00	4.84 ± 0.00	**4.83 ± 0.01**	4.84 ± 0.00	**10.65 ± 0.00**	10.65 ± 0.00	10.67 ± 0.00	**53.55 ± 0.08**	11.72 ± 0.02	9.79 ± 0.07	6.34 ± 0.01	42.46 ± 0.20	**21.68 ± 0.00**	139.47 ± 1.09
Multiple Models + ICT	**5.19 ± 0.00**	4.84 ± 0.00	2.87 ± 0.03	4.67 ± 0.00	4.82 ± 0.00	10.65 ± 0.00	2.77 ± 0.00	10.51 ± 0.00	52.81 ± 0.18	11.67 ± 0.00	9.75 ± 0.19	6.08 ± 0.05	42.06 ± 0.03	**21.68 ± 0.00**	139.24 ± 1.09
Multiple Models + Tri-Mentoring	**5.19 ± 0.00**	4.84 ± 0.00	2.65 ± 0.00	4.78 ± 0.03	4.74 ± 0.00	10.65 ± 0.00	**10.65 ± 0.00**	10.49 ± 0.00	52.91 ± 0.40	**11.73 ± 0.01**	8.92 ± 0.24	6.06 ± 0.12	42.18 ± 0.26	**21.68 ± 0.00**	**141.87 ± 0.26**
MOBO	4.49 ± 0.07	4.84 ± 0.00	4.84 ± 0.00	4.81 ± 0.01	4.84 ± 0.00	9.50 ± 0.00	10.63 ± 0.00	**10.69 ± 0.00**	51.79 ± 0.00	11.61 ± 0.02	0.00 ± 0.00	**6.50 ± 0.00**	**51.83 ± 0.12**	13.77 ± 0.13	N/A
MOBO-qParEGO	4.67 ± 0.06	4.84 ± 0.00	4.84 ± 0.00	4.82 ± 0.00	4.84 ± 0.00	10.64 ± 0.01	10.18 ± 0.05	10.61 ± 0.00	47.81 ± 1.93	11.60 ± 0.00	**10.19 ± 0.23**	5.81 ± 0.11	N/A	N/A	N/A
MOBO-JES	4.85 ± 0.03	**4.84 ± 0.00**	4.83 ± 0.00	4.82 ± 0.00	4.84 ± 0.00	10.28 ± 0.00	10.65 ± 0.00	10.61 ± 0.03	50.30 ± 0.00	11.59 ± 0.02	9.43 ± 0.10	6.20 ± 0.03	N/A	N/A	N/A

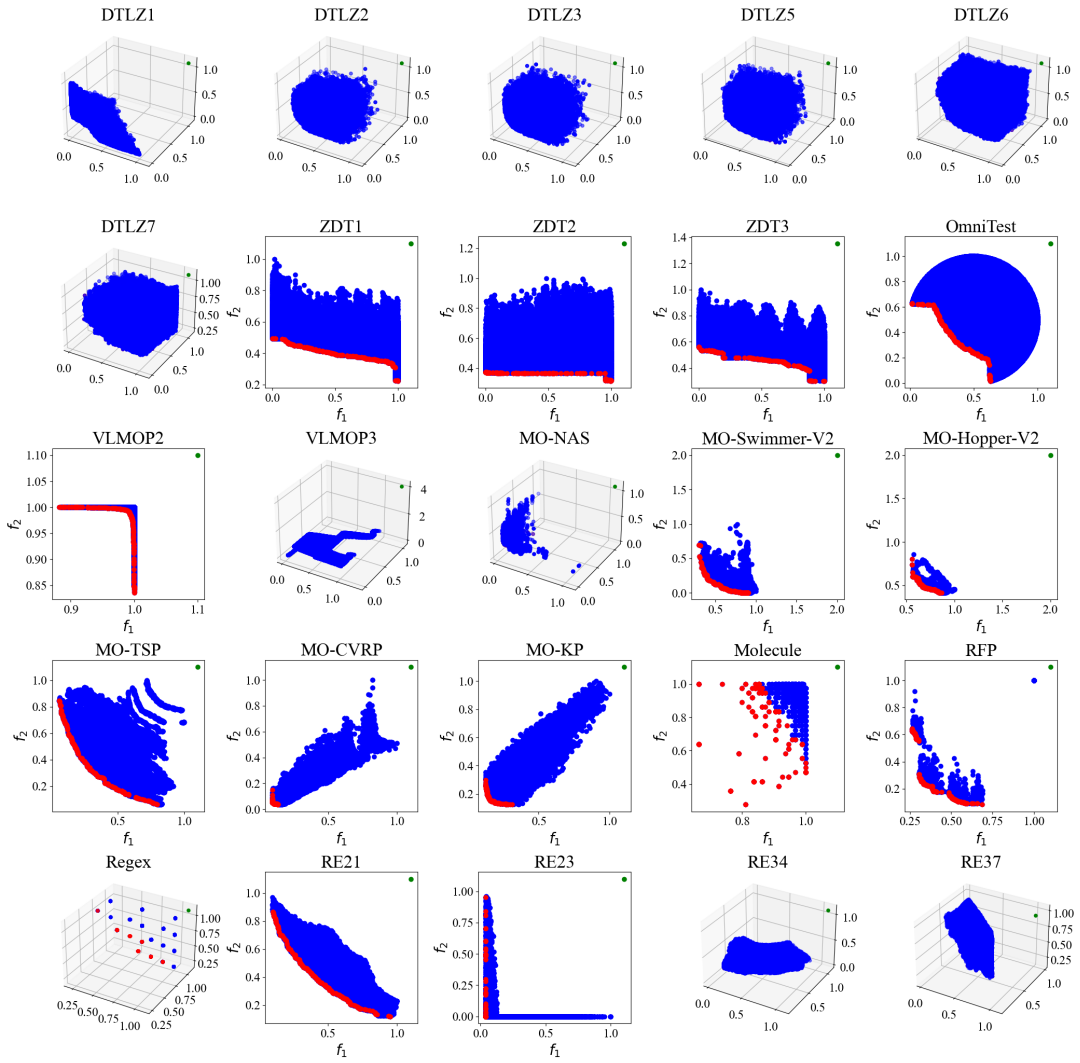

Figure 5. Visualization of datasets in Off-MOO-Bench. Blue points represent the offline dataset, and red points represent the 256 best-non-dominated solutions over the dataset. Note that some red dots are not visible in the graph due to the plot perspective.

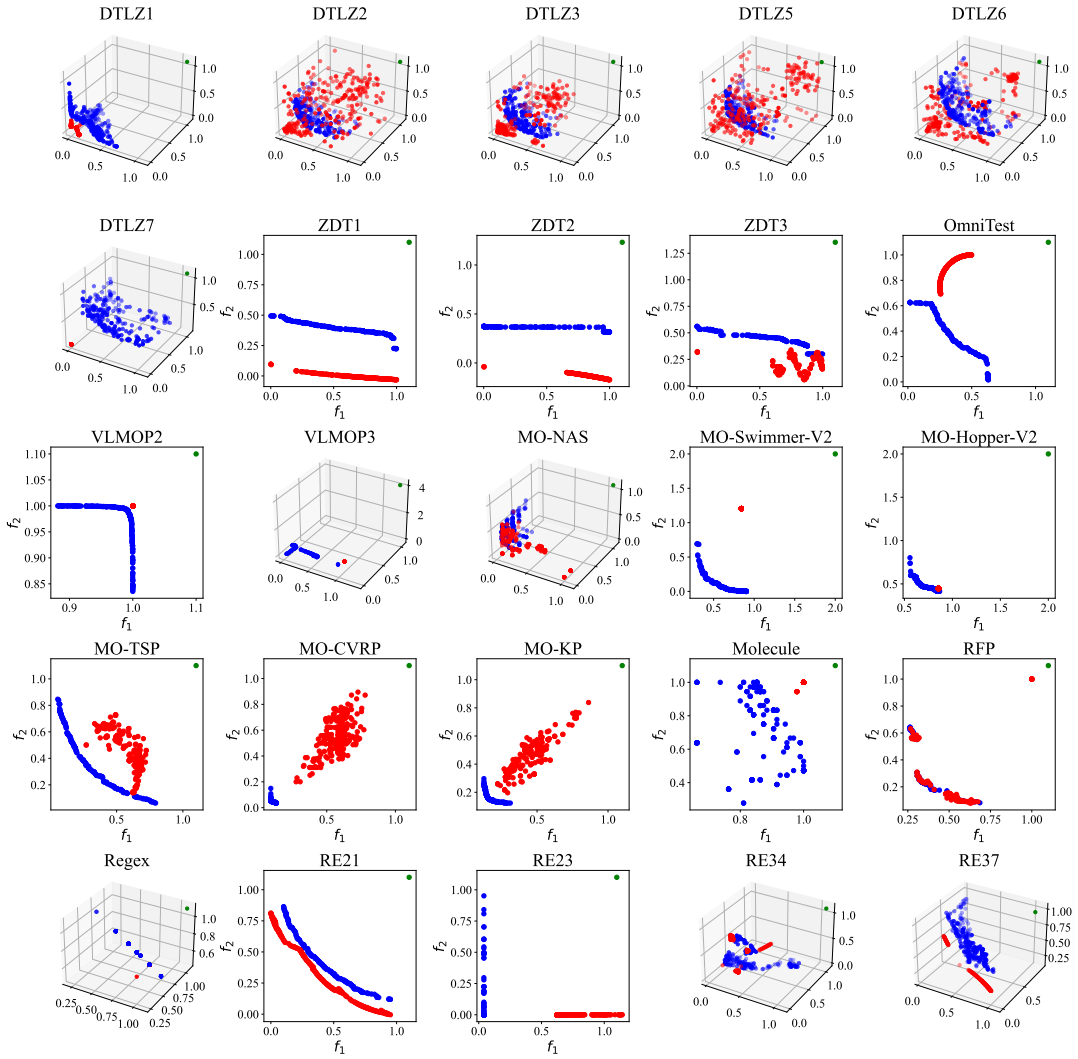

Figure 6. Visualization of Pareto fronts found by Multi-Head + GradNorm. Blue points represent the initial population, which are the 256 best-non-dominated solutions over the offline dataset. Red points represent the Pareto fronts found by algorithm, and green points represent the reference points (i.e., nadir points) that are set by us manually.