FoMEMO: Towards Foundation Models for Expensive Multi-objective Optimization

Yiming Yao[1], **Fei Liu**[1], **Liang Zhao**[1], **Xi Lin**[1], **Qingfu Zhang**[1]

[1]Department of Computer Science, City University of Hong Kong

{yimingyao3-c, fliu36-c, liazhao5-c, xi.lin}@my.cityu.edu.hk, qingfu.zhang@cityu.edu.hk

Abstract

Expensive multi-objective optimization is a prevalent and crucial concern in many real-world scenarios, where sample-efficiency is vital due to the limited evaluations to recover the true Pareto front for decision making. Existing works either involve rebuilding Gaussian process surrogates from scratch for each objective in each new problem encountered, or rely on extensive past domain experiments for pre-training deep learning models, making them hard to generalize and impractical to cope with various emerging applications in the real world. To address this issue, we propose a new paradigm named **FoMEMO** (**Fo**undation **M**odels for **E**xpensive **M**ulti-objective **O**ptimization), which enables the establishment of a foundation model conditioned on any domain trajectory and user preference, and facilitates fast in-context optimization based on the predicted preference-wise aggregation posteriors. Rather than accessing extensive domain experiments in the real world, we demonstrate that pre-training the foundation model with a diverse set of hundreds of millions of synthetic data can lead to superior adaptability to unknown problems, without necessitating any subsequent model training or updates in the optimization process. We evaluate our method across a variety of synthetic benchmarks and real-word applications, and demonstrate its superior generality and competitive performance compared to existing methods.

1 Introduction

Optimizing multiple competitive objectives simultaneously is ubiquitous in many real-world applications, such as neural architecture search (Ying et al. 2019), engineering design (Tanabe and Ishibuchi 2020) and scientific discovery (Dara et al. 2022). The primary difficulty in these scenarios stems from the black-box nature of real-world evaluations, which are often costly and time-consuming. When decision-makers have access to only a limited number of evaluations, it becomes challenging to accurately recover the unknown true Pareto fronts.

To achieve sample-efficiency, traditional approaches, such as multi-objective Bayesian optimization, typically build Gaussian process (GP) models (Williams and Rasmussen 1995) as the surrogates to approximate the objective functions, and then optimize well-designed acquisition functions to generate candidate solutions continuously to update the surrogates in an online manner (Daulton, Balandat, and Bakshy 2020, 2021). However, these methods typically need to rebuild surrogates from scratch for each objective when encountering a new problem, and the efficiency is greatly limited by the cost of GP training and inference (Williams and Rasmussen 2006). To enhance the generalization and scalability in downstream optimization, some research focuses on pre-training deep learning models in offline scenarios (Xue et al. 2024; Yuan et al. 2024). These approaches typically assume accessing to a large amount of real-world experimental data is available for training. However, acquiring large datasets in practical applications, especially in unknown or emerging fields, is often challenging or even impossible. Moreover, these methods also require training dedicated models for specific problems, and hardly be generalized to unseen scenarios.

In real-world contexts, a wide range of multi-objective optimization problems arise daily across various fields, each exhibiting unique characteristics. Given the limitations of existing methodologies, a compelling research question arises: *Can we devise a universal method that can efficiently address a wide range of emerging real-world multi-objective optimization problems?*

In this paper, we attempt to answer this question by exploring a new paradigm named FoMEMO. It enables us to establish a foundation model conditioned on arbitrary domain trajectory and user preference, and facilitates fast in-context optimization based on the predicted preference-wise aggregation posteriors. What distinguishes our approach from existing methods lies in two aspects. Firstly, unlike traditional online methods that require rebuilding a surrogate from scratch for each objective in newly encountered problems and necessitate frequent, costly model updates during optimization, our approach involves pre-training a foundation model only once, which can be directly adapted to arbitrary scenarios throughout the entire optimization process in an in-context manner. This means that by simply providing domain trajectories and user preferences as contextual information, the pre-trained model can achieve few-shot generalization in unknown scenarios to initiate fast optimization without the need for any subsequent model training or updates. Secondly, rather than accessing expensive real-world data from extensive past experiments, we continuously sample a large amount of diverse synthetic data from a computationally efficient sampling mechanism during pre-training. By simulating a large set of potential scenarios that may exist in the real world, the foundation model emerges with the ability to learn to predict the aggregated posteriors conditioned on any contexts.

To the best of our knowledge, this is the first attempt to establish a foundation model for addressing expensive multi-objective optimization problems in an in-context manner. The main contributions of this paper are as follows:

- We pre-train a foundation model using hundreds of millions of sampled synthetic data, enabling it to learn to predict aggregated posteriors conditioned on arbitrary domain trajectories and user preferences. This approach eliminates the need for access to expensive experimental data from the real world, allowing for efficient and adaptable optimization across diverse scenarios.
- Based on the predicted aggregated posteriors, we develop preference-based and preference-free acquisition functions that can be quickly optimized to generate candidate solutions in an in-context manner, without requiring any subsequent model updates.
- We evaluate the proposed method on both synthetic problems and real-world applications. The experimental results show that the proposed method demonstrates superior adaptability and generality across a wide range of unseen problems. In comparison to existing methods, our approach exhibits competitive optimization performance and high efficiency in the majority of cases.

2 Related Work

2.1 Multi-objective Bayesian Optimization

Multi-objective Bayesian optimization (MOBO) aims to identify a finite set of Pareto optimal solutions within a constrained evaluation budget in an online manner. A popular class of MOBO algorithms is based on decomposition, wherein the original MOP is converted into several single-objective sub-problems that are easier to solve. For example, ParEGO (Knowles 2006) and TS-TCH (Paria, Kandasamy, and Póczos 2020) randomly select a preference in each iteration and optimize the corresponding sub-problem using single-objective BO. MOEA/D-EGO (Zhang et al. 2009) pre-defines several sub-problems with uniformly distributed preferences and then employs the population-based MOEA/D algorithm (Zhang and Li 2007) to jointly optimize the utilities of all sub-problems. Another representative type of MOBO algorithm is based on indicators. For instance, (Emmerich, Giannakoglou, and Naujoks 2006) developed expected hypervolume improvement (EHVI), which extends from the widely used expected improvement (EI) acquisition function (Jones, Schonlau, and Welch 1998), providing an overall performance across all objectives with uncertainty. To extend EHVI to the parallel evaluation setting, qEHVI (Wada and Hino 2019; Daulton, Balandat, and Bakshy 2020) was developed and further adapted to noisy scenarios (Daulton, Balandat, and Bakshy 2021). In addition to the application in designing acquisition functions, multi-objective indicators have also been utilized in batch sample selection before expensive evaluations (Bradford, Schweidtmann, and Lapkin 2018; Konakovic Lukovic, Tian, and Matusik 2020; Lin et al. 2022). While existing methods have demonstrated their effectiveness, it is important to note that they require the modeling of each objective for every new problem from the ground up. Additionally, the costs associated with GP training and inference further constrain their scalability.

2.2 Offline Multi-objective Optimization

Beyond online optimization, there also exit scenarios where only pre-collected offline data is accessible and iterative online evaluations are not permitted. Although our work mainly focuses on online methods, we still present a briefly introduction here to highlight the superiority of the synthetic training paradigm in our method. In the offline context, a common approach involves training a deep neural network (DNN) on a static dataset to serve as an oracle (Trabucco et al. 2021; Yu et al. 2021; Chen et al. 2023). This is followed by applying search methods, such as gradient descent or evolutionary algorithms, to generate a set of candidate solutions that are potentially superior to those present in the dataset. (Xue et al. 2024) introduced the first benchmark for offline multi-objective optimization and proposed DNN-based and GP-based methods. (Yuan et al. 2024) developed ParetoFlow, specifically designed to guide flow sampling to approximate the Pareto front. However, existing offline methods typically require a large amount of training data from expensive experiments, which is often impractical for various emerging applications in the real-world. Moreover, these methods often necessitate the training of dedicated models for specific problems, which limits their ability to generalize to unseen scenarios.

2.3 Universal Regression Based on Transformers

Recently, Transformers (Vaswani et al. 2017) have become the preferred architecture for deep learning and foundation models. As model sizes and training data scaling, the models exhibit remarkable in-context learning (ICL) capabilities (Brown et al. 2020): they can make accurate predictions quickly in new tasks when prompted with only a few training examples, without any parameter update. Several research has demonstrated that Transformers can learn in-context simple logistic regression algorithms (Garg et al. 2022; Akyürek et al. 2022). Beyond predicting labels, Transformer Neural Processes (Nguyen and Grover 2022) enhance the prediction with uncertainty as an alternative framework for uncertainty-aware meta learning. Prior-data Fitted Networks (PFNs) (Müller et al. 2021) further show that Transformers can even learn to perform Bayesian inference with ICL. The developed universal regression models not only provide new insights into various model-based prediction topics (Hollmann et al. 2022, 2025; Dooley et al. 2023; Song et al. 2024a), but also promote the regress-then-optimize framework towards a generalized optimization paradigm in both numerical (Müller et al. 2023; Nguyen, Agrawal, and Grover 2023; Rakotoarison et al. 2024) and even linguistic space (Nguyen et al. 2024; Tan et al. 2025). Building on this potential, we aim to develop a general optimization method for multi-objective scenarios, aligning more closely with real-world challenges that involve multi-criteria outputs.

3 Preliminaries

3.1 Expensive Multi-objective Optimization

In this paper, we address the following multi-objective optimization problem (MOP):

$$\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$ is the feature vector in the decision space $\mathcal{X} \subset \mathbb{R}^d$, and $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^m$ is the objective vector containing m continuous objective functions. We consider the scenario in which all the objective functions are expensive to evaluate, with no known analytical expressions or gradient information. Typically, there is no single solution that can simultaneously achieve the optimal for all objectives. Instead, the goal is to find a set of solutions with Pareto optimal objective trade-offs. We say an objective vector $\boldsymbol{f}(\boldsymbol{x})$ Pareto dominates another $\boldsymbol{f}(\boldsymbol{x}')$, denoted as $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}')$ if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}'), \forall i \in \{1, \ldots, m\}$ and there exists $j \in \{1, \ldots, m\}$ such that $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}')$. A solution $\boldsymbol{x}^* \in \mathcal{X}$ is Pareto optimal if there is no $\boldsymbol{x}$ such that $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$. The set of all Pareto optimal solutions is called the Pareto set (PS), and the image of PS in the objective space is called the Pareto front (PF).

3.2 Scalarization for Multiple Objectives

Scalarization is a straightforward and effective technique for transforming multiple objectives into a single metric. A classic approach is based on decomposition, in which multiple objectives are aggregated using a family of scalarization functions $s_{\boldsymbol{\lambda}}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ parameterized by a set of preferences $\Delta = \{\boldsymbol{\lambda} \in \mathbb{R}^m_+ \mid \|\boldsymbol{\lambda}\|_1 = 1\}$. We consider the popular Tchebycheff scalarization function:

$$s^{\text{tch}}_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \max_{1\leq i\leq m} \{\lambda_i (y_i(\boldsymbol{x}) - z^*_i)\}, \tag{2}$$

where $y_i(\boldsymbol{x})$ is the i-th observation value of the corresponding true function $f_i(\boldsymbol{x})$ at the input $\boldsymbol{x}$, $\boldsymbol{z}^* = (z^*_1, \cdots, z^*_m)$ is the ideal point (lower bound) in the objective space. A promising property behind Tchebycheff scalarization function is that we can identify each Pareto optimal solution by optimizing the scalarization function with a specific but unknown preference (Choo and Atkins 1983):

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} s^{\text{tch}}_{\boldsymbol{\lambda}}(\boldsymbol{x})). \tag{3}$$

In out work, we use the negative of the scalarization function as the aggregation target $g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = -s_{\boldsymbol{\lambda}}(\boldsymbol{x})$ for the convenience of model training and inference. Thus, we wish to maximize $g_{\boldsymbol{\lambda}}(\boldsymbol{x})$ given each preference vector $\boldsymbol{\lambda}$.

Another representative scalarization method is the hypervolume (HV) indicator (Zitzler and Thiele 2002), which is monotonic to the Pareto dominance relation. The hypervolume of a finite approximate Pareto front $\mathcal{P}$ is defined as:

$$\text{HV}(\mathcal{P}) = \boldsymbol{\Lambda}_m(\{\boldsymbol{v} \in \mathbb{R}^m | \exists \boldsymbol{p} \in \mathcal{P} : \boldsymbol{p} \prec \boldsymbol{v} \prec \boldsymbol{r}\}), \tag{4}$$

where $\boldsymbol{\Lambda}_m$ denotes the m-dimensional Lebesgue measure, and $\boldsymbol{r} \in \mathbb{R}^m$ is a pre-defined reference point.

3.3 Bayesian Optimization

Bayesian optimization (BO) (Shahriari et al. 2015; Frazier 2018; Garnett 2023) is a sample-efficient technique for solving expensive black-box optimization problems. BO utilizes probabilistic surrogate models, typically Gaussian processes (GPs) (Williams and Rasmussen 2006), to provide an prediction with calibrated uncertainty for the latent functions $\boldsymbol{f}$ based on the true observed data $D_n = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}^n_{i=1}$. To suggest the candidate solutions for next evaluations, BO defines and optimizes an acquisition function $\alpha : \mathcal{X} \mapsto \mathbb{R}$, which quantifies the utility value of evaluating a new set of solutions based on the predictive distribution of the surrogate models, effectively balancing exploration and exploitation. BO continuously generates candidate solutions for evaluation and updates the surrogate models in an online manner until the total evaluation budget is exhausted.

4 Methodology

4.1 Framework

Instead of building a separate surrogate model from scratch for each objective function in every new problem encountered, the core concept of our method is to develop a universal foundation model. This model is pre-trained once only using a vast corpus of synthetic samples and preferences as context information, and outputs a large set of preference-wise aggregation posterior distributions for fast in-context optimization. We present the details of our framework in Figure 1, which includes synthetic pre-training stage on the left and in-context optimization stage on the right.

Specifically, in the synthetic pre-training stage, rather than accessing expensive real-world experiments, we implement a data sampling mechanism that allows us to heavily sample from it to generate a large set of synthetic data as the context inputs, including the trajectory pairs $D_n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}^n_{i=1}$, the query input $\boldsymbol{x}$ with the corresponding aggregation label $g = -s_{\boldsymbol{\lambda}}(\boldsymbol{x})$ masked, and the artificial preference $\boldsymbol{\lambda}$ sampled on the simplex, constructing a supervised learning problem for training the foundation model efficiently. We repeatedly synthesize training data as above and optimize the model's parameters $\boldsymbol{\theta}$ to predict the aggregated posterior distributions $q_{\boldsymbol{\theta}}(g|\boldsymbol{x}, D_n; \boldsymbol{\lambda})$ conditioned on the contextual inputs. We will present the details of data generation in Section 4.2.

Once the pre-training is completed, during the in-context optimization stage, users can tackle new problems by simply providing the evaluated trajectory from arbitrary unseen real-world applications, as well as the the user's potential preferences (if any) as prompt inputs for the foundation model. Based on the aggregated posteriors predicted by the foundation model in a single neural network forward pass, various acquisition functions can be easily derived to initiate fast in-context optimization to generate the next candidate for expensive evaluation for the utility of interest, without the need for any additional model training or updates during the optimization process.

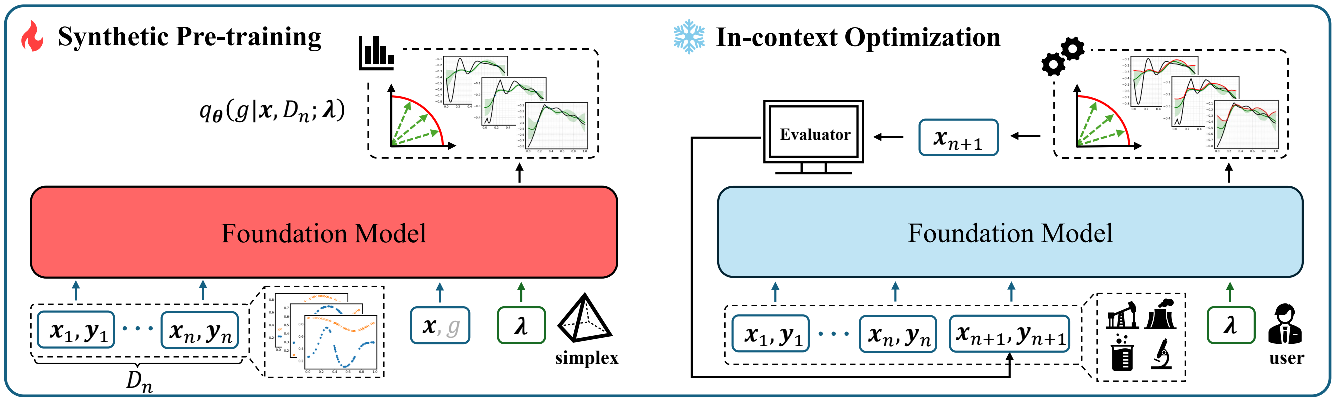

Figure 1: Overview of FoMEMO framework, including synthetic pre-training stage (left) and in-context optimization stage (right). During synthetic pre-training, the foundation model is fed with a large set of synthetic data as the context inputs, which include the trajectory pairs $D_n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, the query input $\boldsymbol{x}$ with its aggregation label $g = -s_{\boldsymbol{\lambda}}(\boldsymbol{x})$ masked, and the corresponding artificial preference $\boldsymbol{\lambda}$ sampled on the simplex. The model parameterized by $\boldsymbol{\theta}$ is trained once only to predict the aggregated posterior distributions $q_{\boldsymbol{\theta}}(g|\boldsymbol{x}, D_n; \boldsymbol{\lambda})$ as the output conditioned on the contextual information. During the in-context optimization stage, users can readily address new problems by simply providing evaluated trajectories from arbitrary unseen real-world applications, along with any potential user preferences (if applicable) as prompt inputs for the foundation model. Using the predicted aggregated posteriors as the basis, various acquisition functions can be naturally derived to enable fast in-context optimization. These acquisition functions can be efficiently optimized to suggest the next candidate within the utility of interest, free from any additional model training or updates throughout the entire optimization process.

4.2 Data Generation

Given that expensive data of real-world experiments are typically scarce and users' potential preferences can be difficult to obtain, we implement a data generation mechanism designed to simulate a substantial amount of potential scenarios that the model may encounter. This process enables us to create a diverse and lightweight dataset that enhances the model's ability to generalize and respond rapidly to various situations, ultimately improving its optimization performance in emerging real-world tasks.

Our data generation mechanism consists of three steps. Firstly, we sample general parameters for each set of the data, including the number of features d, the number of objectives m and the length of trajectory n. Secondly, in order to simulate the multitude of implicit mappings that may exist in the real world, we sample a variety of synthetic functions from Gaussian processes (GPs). The choice of GPs is motivated by their cost-effectiveness and flexibility. Sampling from a GP prior is a easy and cost-effective method that enables the generation of a substantial number of latent mappings at a low cost. Besides, we can easily control the diversity of the generated functions by adjusting the kernel parameters. In the third step, we sample the observed trajectory $D_n = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$ together with an arbitrary query pair $(\boldsymbol{x}, \boldsymbol{y})$ with d-dimensional features and m-dimensional objectives from each sampled function. Since the aggregation target g is determined only by $\boldsymbol{x}$ given $\boldsymbol{\lambda}$, after sampling artificial preferences from the simplex $\Delta = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m \mid \|\boldsymbol{\lambda}\|_1 = 1\}$, we can calculate the aggregation targets efficiently for each query input given arbitrary preferences. We synthesize hundreds of millions of training data using the aforementioned method, and mask the query aggregation targets to formulate a supervised prediction problem. This processing allows the foundation model to learn to effectively infer the preference-wise posterior distribution $q_{\boldsymbol{\theta}}(g|\boldsymbol{x}, D_n; \boldsymbol{\lambda})$.

4.3 Model Architecture

Among the model architectures capable of learning uncertainty-calibrated predictions directly from data, we utilize the Prior-Data Fitted Networks (PFNs) (Müller et al. 2021) as the base framework, as it provides a flexible paradigm for approximating the posterior distribution with arbitrary shape in a principled manner (Nagler 2023). PFNs are based on a Transformer encoder that does not incorporate positional encodings, allowing the model to process data regardless of the order in which the input elements are presented. We use a variable-dimensional encoder to encode each input $\boldsymbol{x}$ and observation $\boldsymbol{y}$ into a token, allowing the model to adapt to cross-dimensional feature and objective spaces. An attention mask is implemented to enable each trajectory point can only attend to each other and the query point can only attend to the trajectory points. We also encode the sampled preference to inform the model to learn to predict the posterior distribution conditioned on which preference. To approximate the complex posterior distribution in high-dimensional feature and objective spaces, PFNs employ a piecewise constant distribution in the regression head. This approach converts the continuous distribution into a discrete distribution by utilizing multiple intervals, thereby transforming the regression task into a classification task to enhance the robustness during training. The detailed implementation can be referred to (Müller et al. 2021).

4.4 Training Details

For any set of synthetic data generated from the sampling mechanism $p(D)$ presented in Section 4.2, we optimize the following loss function, which is the cross-entropy between the held-out aggregation target and the model prediction:

$$\mathbb{E}_{(\boldsymbol{x},y)\cup D_n\sim p(D),\boldsymbol{\lambda}\sim\Delta,g=-s_{\boldsymbol{\lambda}}(\boldsymbol{x})}[-\log q_\theta(g|\boldsymbol{x},D_n;\boldsymbol{\lambda})]. \tag{5}$$

We continuously synthesize data through extensive sampling and optimize the model parameters to minimize the loss function, thereby progressively approaching the true posterior distribution $p(g|\boldsymbol{x},D_n;\boldsymbol{\lambda})$.

Our final foundation model uses the PFNs architecture with 12 layers of Transformer, an embedding size of 512, a hidden size 1024 in feed-forward layers, and 4-head attention, resulting a total of 26.81M parameters. During training, we randomly sample each dataset from a large data space with feature dimensions ranging from 1 to 30, objective dimensions from 1 to 6, and trajectory lengths from 1 to 128. We sample a large number of functions from a standard RBF-kernel GP with zero mean. We randomly sample the length scale kernel parameter $l\sim\Gamma(\alpha=3.0,\beta=6.0)$ for each feature and add a random observation noise $\epsilon\sim\mathcal{N}(0,10^{-4})$ to ensure function diversity. We do not sample the output scale parameter but instead set it to 1.0, as the predicted aggregation target will be normalized during inference. The final model is trained using 500 epochs and 2048 steps in each epoch with a batch size of 256 datasets, resulting more than 200 million datasets in total. The loss function is optimized using Adam optimizer (Kingma and Ba 2014) with linear warmup and cosine annealing (Loshchilov and Hutter 2016) using a learning rate of 10^{-4}. The entire training process is conducted on a single NVIDIA A100 GPU and takes approximately 140 hours. We performed the pre-training only once and used the same model in all experiments in this paper.

4.5 Generating Candidates

Benefiting from the preference-wise aggregation posterior as the output basis, modeling the distribution of infinite sub-problems becomes possible. Acquisition functions such as random scalarization and hypervolume aggregation can be well adapted directly. We have developed two types of acquisition functions including preference-based and preference-free in our work.

We firstly derive the preference-based expected improvement (EI) and upper confidence bound (UCB) acquisition functions based on the single-objective scenario (Müller et al. 2023):

$$\alpha_{\text{EI}}(\boldsymbol{x};\boldsymbol{\lambda})=\mathbb{E}_{q_\theta(g|\boldsymbol{x},D_n;\boldsymbol{\lambda})}[[g_{\boldsymbol{\lambda}}(\boldsymbol{x})-g^*_{\boldsymbol{\lambda}}]_+], \tag{6}$$

$$\alpha_{\text{UCB}}(\boldsymbol{x};\boldsymbol{\lambda})=\mu_{\boldsymbol{\lambda}}(\boldsymbol{x})+\beta\sigma_{\boldsymbol{\lambda}}(\boldsymbol{x}), \tag{7}$$

where $g^*_{\boldsymbol{\lambda}}$ is the best value of the aggregation $g_{\boldsymbol{\lambda}}(\boldsymbol{x})$ given an arbitrary preference $\boldsymbol{\lambda}$. $\mu_{\boldsymbol{\lambda}}(\boldsymbol{x})$ and $\sigma_{\boldsymbol{\lambda}}(\boldsymbol{x})$ are the mean and standard deviation of the aggregated posterior $q_\theta(g|\boldsymbol{x},D_n;\boldsymbol{\lambda})$, respectively. β is a constant, which we set to 1.0 in this paper. As inference the acquisition functions is conditioned on the preference, during the optimization process, we randomly sample the preferences in each iteration and optimize the acquisition function to generate the next candidate solutions for evaluation.

To marginalize the preferences, we draw inspiration from a key observation: using infinite reference vectors uniformly sampled from the unit sphere $S=\{\boldsymbol{w}\in\mathbb{R}^m_+\mid\|\boldsymbol{w}\|_2=1\}$, the hypervolume metric can be represented as the mean of a scalarization function (Deng and Zhang 2019; Zhang and Golovin 2020). Based on this insight, we develop a preference-free acquisition function that uses the UCB utility in Equality (7) to approximate the hypervolume improvment with uncertainty (UHVI) for any $\boldsymbol{x}$ given the evaluated inputs $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$:

$$\begin{aligned}\alpha_{\text{UHVI}}(\boldsymbol{x})=c_m\mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m\max\{0,\\ [\min_{\boldsymbol{x}\in\boldsymbol{X}}\{-g_{\boldsymbol{\lambda}}(\boldsymbol{x})\}]^m-[-\alpha_{\text{UCB}}(\boldsymbol{x};\boldsymbol{\lambda})]^m\}],\end{aligned} \tag{8}$$

where $c_m=\frac{\pi^{m/2}}{2^m\Gamma(m/2+1)}$ is a constant that depends only on m. $c_{\boldsymbol{\lambda}}=\sqrt{\sum_{j=1}^m\frac{1}{\lambda_j^2}}$ is a transformation constant that depends only on $\boldsymbol{\lambda}$. It enables us to convert sampling reference vectors from the unit sphere to sampling preferences from the simplex, and can be adapted to our model by simply using the transformation $w_j=1/(\lambda_jc_{\boldsymbol{\lambda}})$. The detailed derivation can be found in the appendix.

5 Experiments

Baseline Algorithms We compare FoMEMO against existing state-of-the-art model-based methods including qParEGO (Knowles 2006; Daulton, Balandat, and Bakshy 2020), qEVHI (Daulton, Balandat, and Bakshy 2020) and qNEHVI (Daulton, Balandat, and Bakshy 2021), all of which demonstrate promising scalability in scenarios involving noise interruptions and parallel optimization. Additionally, we include a Sobol sequence-based random sampling method as a trivial baseline for comparison.

Benchmark Problems We empirically evaluate the algorithms on various synthetic benchmarks and real-world applications, encompassing a diverse set of feature and objective spaces, as well as various Pareto fronts characteristics with convex, concave and disconnected shapes. For synthetic benchmarks, we consider the popular ZDT (Zitzler, Deb, and Thiele 2000) and Omnitest (Deb and Tiwari 2008) test suits. For real-world applications, we use the RE test suite (Tanabe and Ishibuchi 2020) consisting various multi-objective engineering design problems, such as four bar truss design, reinforced concrete beam design, rocket injector design, etc. The detailed descriptions can be referred to appendix.

Experiment Setting For model-based algorithms, we run each experiment using $2(d+1)$ initial samples and a total of 100 evaluations beyond the initial points. The implementations of qEVHI, qNEHVI and qParEGO are based on BoTorch (Balandat et al. 2020), we use the default setting that optimizes the acquisition functions via multi-start L-BFGS-B optimizer with 20 restarts seeded from 1024 pseudo-random samples through BoTorch's initialization heuristic, we keep the same parameter setting in FoMEMO.

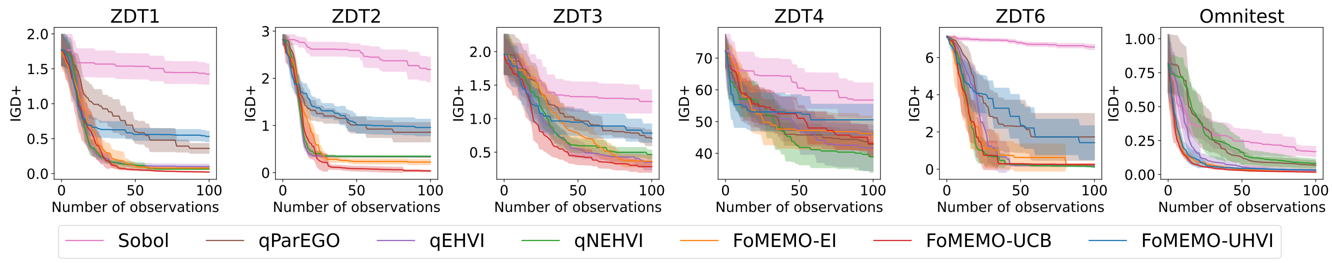

Figure 2: Sequential optimization performance of IGD+ metrics (↓) w.r.t. the number of observations (beyond initial points) on synthetic problems. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

Table 1: Means (stds) of best hypervolume metrics (↑) in sequential optimization on real-world problems over 10 independent runs. The best and second best mean results for each row are highlighted with shaded background and underlines respectively.

Test Instances	Sobol	qParEGO	qEHVI	qNEHVI	FoMEMO-EI	FoMEMO-UCB	FoMEMO-UHVI
RE21	7.366e-01(1.22e-02)	8.586e-01(6.61e-03)	8.825e-01(1.11e-04)	8.824e-01(2.66e-04)	8.683e-01(1.62e-03)	8.720e-01(1.37e-03)	8.629e-01(1.06e-03)
RE22	5.376e-01(2.88e-02)	5.473e-01(3.81e-02)	5.333e-01(4.46e-02)	5.732e-01(3.80e-02)	4.982e-01(4.31e-02)	5.043e-01(7.61e-02)	6.242e-01(3.42e-02)
RE23	4.107e-01(1.18e-01)	1.111e+00(9.70e-03)	1.119e+00(8.77e-03)	1.120e+00(9.26e-03)	1.154e+00(3.32e-03)	1.156e+00(1.04e-03)	1.011e+00(5.45e-02)
RE24	9.695e-01(1.19e-01)	1.164e+00(1.29e-03)	1.138e+00(4.64e-03)	1.153e+00(3.83e-03)	1.164e+00(1.74e-03)	1.161e+00(1.20e-03)	1.123e+00(8.61e-03)
RE25	1.059e-01(9.92e-02)	9.064e-01(5.54e-02)	9.932e-01(2.13e-02)	9.531e-01(2.01e-02)	1.053e+00(1.32e-02)	1.047e+00(1.02e-02)	6.843e-01(7.65e-02)
RE31	1.233e+00(3.56e-02)	1.318e+00(4.60e-03)	1.270e+00(1.33e-02)	1.300e+00(5.35e-03)	1.314e+00(3.42e-03)	1.310e+00(4.70e-03)	1.298e+00(1.97e-02)
RE32	1.273e+00(1.30e-02)	1.316e+00(6.28e-03)	1.321e+00(3.85e-03)	1.318e+00(4.94e-03)	1.326e+00(1.31e-03)	1.327e+00(1.83e-03)	1.294e+00(8.23e-03)
RE33	1.130e+00(3.50e-02)	1.293e+00(7.83e-03)	1.305e+00(5.02e-03)	1.310e+00(1.01e-03)	1.310e+00(6.72e-04)	1.312e+00(5.31e-04)	1.272e+00(6.93e-03)
RE34	5.788e-01(2.85e-02)	9.183e-01(1.40e-02)	1.039e+00(4.80e-04)	1.038e+00(8.30e-04)	9.890e-01(6.40e-03)	1.005e+00(8.92e-03)	9.367e-01(1.00e-02)
RE35	1.186e+00(9.18e-03)	1.266e+00(8.01e-03)	1.294e+00(1.73e-03)	1.294e+00(2.16e-03)	1.294e+00(3.86e-03)	1.297e+00(9.41e-04)	1.260e+00(6.02e-03)
RE36	3.135e-01(4.38e-02)	8.161e-01(2.77e-02)	8.877e-01(8.70e-03)	8.958e-01(7.11e-03)	9.175e-01(6.87e-03)	9.192e-01(4.87e-03)	8.566e-01(1.13e-02)
RE37	6.046e-01(2.31e-02)	8.068e-01(1.97e-02)	8.862e-01(8.58e-04)	8.864e-01(1.07e-03)	8.515e-01(3.69e-03)	8.642e-01(2.99e-03)	8.325e-01(4.08e-03)

Performance Metric To compare the performance of different algorithms on synthetic problems with known true Pareto fronts, we utilize the inverted generational distance plus (IGD+) indicator (Ishibuchi et al. 2015) as the metric to evaluate the quality of the final solutions obtained by each algorithm. For real-world applications, we use the hypervolume indicator (Zitzler and Thiele 2002) as the metric in the normalized objective space for fair comparison. Each experiment is conducted with 10 independent runs. See appendix for more detailed experimental settings.

5.1 Optimization Performance

Figure 2 shows the sequential optimization performance of IGD+ metrics w.r.t. the number of observations on synthetic problems. The experimental results indicate that the three FoMEMO methods are highly competitive when compared to existing approaches. Specifically, FoMEMO-UCB has demonstrated superior performance compared to other algorithms in most of the tested problems.

Table 1 represents the results of sequential optimization on real-world applications in terms of the hypervolume metric. Experimental results indicate that our method together with the three proposed acquisition functions achieve superior performance in most cases. This demonstrates that by pre-training the foundation model with a large set of diverse synthetic data, the model becomes highly adaptable and generalizable to complex problems that it has not encountered during training, without having to access to large amounts of expensive real-world experimental data.

5.2 Parallel Scalability

In addition to sequential optimization, we further investigate the scalability of our proposed method in the context of parallel optimization. We demonstrate the parallel optimization performance of all algorithms when generating $q = 5$ candidate solutions in each iteration as an example, as shown in Figure 3. More results can be found in the appendix. Excluding the initial sample point, a total of 20 iterations are conducted. For preference-based acquisition functions, we iterate over q preferences to generate q candidate solutions in each iteration. For preference-free methods, using a limited number of preferences to approximate the hypervolume improvement introduces a degree of deviation. Therefore, we repeatedly perform the UHVI calculation q times, resulting in q candidates for parallel evaluation. Our findings indicate that, despite employing a straightforward parallelization strategy, our algorithms remain competitive when compared to existing state-of-art parallelization methods. More principled parallel acquisition functions and sample selection strategies deserve further exploration in the future.

5.3 Runtime Advantage

We compare the query time of different algorithms in sequential optimization across all real-world applications, as

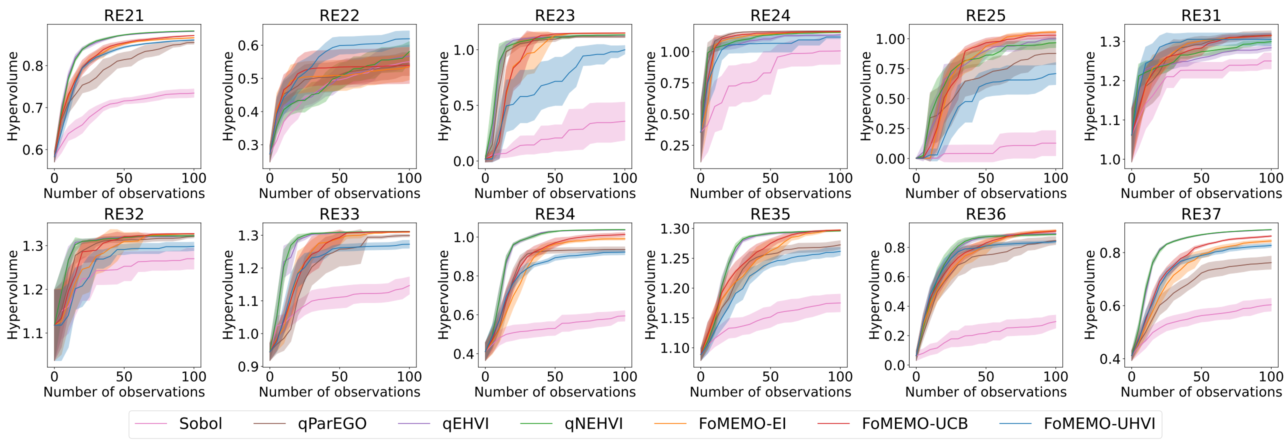

Figure 3: Parallel optimization performance (q=5) of hypervolume metrics (↑) w.r.t. the number of observations (beyond initial points) on real-world applications. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

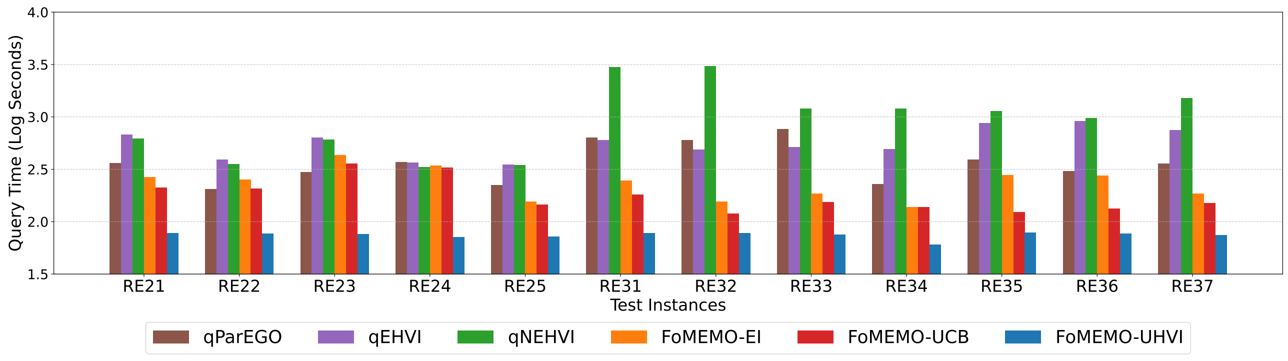

Figure 4: Comparison of query time of different model-based algorithms in sequential optimization. We record the average total time in seconds (using log scaling) taken by different algorithms to generate candidate solutions under 10 independent runs.

depicted in Figure 4. The query time encompasses only the sum of duration required to optimize the acquisition functions based on the models when generating candidate solutions, excluding the time spent on model updates (if applicable) and the time dedicated to evaluating the true functions. The comparisons are conducted on the same device, and the reported results are averages from 10 independent runs. The experimental results demonstrate the superiority of the unique in-context optimization of our methods in terms of the query time. Overall, the inclusion of expensive hypervolume calculations leads to the indicator-based algorithms qEHVI and qNEHVI experiencing the longest query time, especially for problems with higher objective dimensions. In contrast, benefiting from the fast inference of the foundation model, the in-context optimization of all acquisition functions of FoMEMO demonstrates orders of magnitude efficiency. Among these, FoMEMO-UHVI requires the least amount of time, as it samples fewer preferences to efficiently approximate the hypervolume indicator. However, this efficiency comes at the cost of some optimization performance. In contrast, FoMEMO-UCB has achieved a favorable balance between optimizing performance and running time.

6 Conclusion and Future Work

We propose a novel paradigm named FoMEMO, which, to the best of our knowledge, represents the first attempt to establish a foundation model to address a wide range of expensive MOPs in an in-context manner. Extensive experiments verify that our method exhibits superior adaptability and generalization in both synthetic problems and real-world applications, showing competitive performance in most cases compared to existing methods.

In the future, we expect that this work opens up opportunities for many exciting directions worth exploring: (i) Develop architectures that allow for datasets with larger sizes, including embedding techniques and training schemes that efficiently handle higher-dimensional feature and objective spaces. (ii) Develop efficient acquisition functions that adapt to high-dimensional and parallel settings. (iii) Explore alternative methods for generating synthetic data, such as integrating efficient calculations of hypervolume or R2 (Hansen and Jaszkiewicz 1994; Brockhoff, Wagner, and Trautmann 2012) indicators as label responses to support efficient training. (iv) Investigate whether fine-tuning can help improve the performance in domain-specific tasks.

References

Akyürek, E.; Schuurmans, D.; Andreas, J.; Ma, T.; and Zhou, D. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33: 21524–21538.

Bradford, E.; Schweidtmann, A. M.; and Lapkin, A. 2018. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2): 407–438.

Brockhoff, D.; Wagner, T.; and Trautmann, H. 2012. On the properties of the R2 indicator. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 465–472.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, C. S.; Beckham, C.; Liu, Z.; Liu, X. S.; and Pal, C. 2023. Parallel-mentoring for offline model-based optimization. *Advances in Neural Information Processing Systems*, 36: 76619–76636.

Choo, E. U.; and Atkins, D. R. 1983. Proper efficiency in nonconvex multicriteria programming. *Mathematics of Operations Research*, 8(3): 467–470.

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

Daulton, S.; Balandat, M.; and Bakshy, E. 2020. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in neural information processing systems*, 33: 9851–9864.

Daulton, S.; Balandat, M.; and Bakshy, E. 2021. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in neural information processing systems*, 34: 2187–2200.

Daulton, S.; Cakmak, S.; Balandat, M.; Osborne, M. A.; Zhou, E.; and Bakshy, E. 2022. Robust multi-objective bayesian optimization under input noise. In *International Conference on Machine Learning*, 4831–4866. PMLR.

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

Deng, J.; and Zhang, Q. 2019. Approximating hypervolume and hypervolume contributions using polar coordinate. *IEEE Transactions on Evolutionary Computation*, 23(5): 913–918.

Dooley, S.; Khurana, G. S.; Mohapatra, C.; Naidu, S. V.; and White, C. 2023. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36: 2403–2426.

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

Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Garg, S.; Tsipras, D.; Liang, P. S.; and Valiant, G. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35: 30583–30598.

Garnett, R. 2023. *Bayesian optimization*. Cambridge University Press.

Hansen, M. P.; and Jaszkiewicz, A. 1994. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark.

Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2022. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S. B.; Schirrmeister, R. T.; and Hutter, F. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045): 319–326.

Ishibuchi, H.; Masuda, H.; Tanigaki, Y.; and Nojima, Y. 2015. Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, 110–125. Springer.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4): 455–492.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Konakovic Lukovic, M.; Tian, Y.; and Matusik, W. 2020. Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems*, 33: 17708–17720.

Lin, X.; Liu, Y.; Zhang, X.; Liu, F.; Wang, Z.; and Zhang, Q. 2024. Few for many: Tchebycheff set scalarization for many-objective optimization. *arXiv preprint arXiv:2405.19650*.

Lin, X.; Yang, Z.; Zhang, X.; and Zhang, Q. 2022. Pareto set learning for expensive multi-objective optimization. *Advances in neural information processing systems*, 35: 19231–19247.

Liu, Y.; Lu, C.; Lin, X.; and Zhang, Q. 2024. Many-objective cover problem: Discovering few solutions to cover many objectives. In *International Conference on Parallel Problem Solving from Nature*, 68–82. Springer.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Müller, S.; Feurer, M.; Hollmann, N.; and Hutter, F. 2023. Pfns4bo: In-context learning for bayesian optimization. In *International Conference on Machine Learning*, 25444–25470. PMLR.

Müller, S.; Hollmann, N.; Arango, S. P.; Grabocka, J.; and Hutter, F. 2021. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*.

Nagler, T. 2023. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning*, 25660–25676. PMLR.

Nguyen, T.; Agrawal, S.; and Grover, A. 2023. Expt: Synthetic pretraining for few-shot experimental design. *Advances in Neural Information Processing Systems*, 36: 45856–45869.

Nguyen, T.; and Grover, A. 2022. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. *arXiv preprint arXiv:2207.04179*.

Nguyen, T.; Zhang, Q.; Yang, B.; Lee, C.; Bornschein, J.; Miao, Y.; Perel, S.; Chen, Y.; and Song, X. 2024. Predicting from Strings: Language Model Embeddings for Bayesian Optimization. *arXiv preprint arXiv:2410.10190*.

Paria, B.; Kandasamy, K.; and Póczos, B. 2020. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, 766–776. PMLR.

Rakotoarison, H.; Adriaensen, S.; Mallik, N.; Garibov, S.; Bergman, E.; and Hutter, F. 2024. In-context freeze-thaw bayesian optimization for hyperparameter optimization. *arXiv preprint arXiv:2404.16795*.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175.

Song, X.; Li, O.; Lee, C.; Yang, B.; Peng, D.; Perel, S.; and Chen, Y. 2024a. Omnipred: Language models as universal regressors. *arXiv preprint arXiv:2402.14547*.

Song, X.; Zhang, Q.; Lee, C.; Fertig, E.; Huang, T.-K.; Belenki, L.; Kochanski, G.; Ariafar, S.; Vasudevan, S.; Perel, S.; et al. 2024b. The vizier gaussian process bandit algorithm. *arXiv preprint arXiv:2408.11527*.

Tan, R.-X.; Chen, M.; Xue, K.; Wang, Y.; Wang, Y.; Fu, S.; and Qian, C. 2025. Towards Universal Offline Black-Box Optimization via Learning Language Model Embeddings. *arXiv preprint arXiv:2506.07109*.

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

Trabucco, B.; Kumar, A.; Geng, X.; and Levine, S. 2021. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, 10358–10368. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wada, T.; and Hino, H. 2019. Bayesian optimization for multi-objective optimization and multi-point search. *arXiv preprint arXiv:1905.02370*.

Williams, C.; and Rasmussen, C. 1995. Gaussian processes for regression. *Advances in neural information processing systems*, 8.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Xue, K.; Tan, R.-X.; Huang, X.; and Qian, C. 2024. Offline multi-objective optimization. *arXiv preprint arXiv:2406.03722*.

Ying, C.; Klein, A.; Christiansen, E.; Real, E.; Murphy, K.; and Hutter, F. 2019. Nas-bench-101: Towards reproducible neural architecture search. In *International conference on machine learning*, 7105–7114. PMLR.

Yu, S.; Ahn, S.; Song, L.; and Shin, J. 2021. Roma: Robust model adaptation for offline model-based optimization. *Advances in Neural Information Processing Systems*, 34: 4619–4631.

Yuan, Y.; Chen, C.; Pal, C.; and Liu, X. 2024. ParetoFlow: Guided Flows in Multi-Objective Optimization. *arXiv preprint arXiv:2412.03718*.

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

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

Zhang, R.; and Golovin, D. 2020. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International conference on machine learning*, 11096–11105. PMLR.

Zhang, X.; Lin, X.; Xue, B.; Chen, Y.; and Zhang, Q. 2023. Hypervolume maximization: A geometric view of pareto set learning. *Advances in Neural Information Processing Systems*, 36: 38902–38929.

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

Zitzler, E.; and Thiele, L. 2002. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4): 257–271.

Appendix

A Limitations and Potential Future Work

A.1 GP Priors Limitations

In this paper, we focus on addressing a general class of unconstrained multi-objective expensive optimization problems characterized by continuous, non-linear objective functions that exhibit smoothness and can incorporate uncertainty, making them well-suited for Gaussian Process (GP) modeling. Therefore, we can extensively sample from GP priors to generate a substantial amount of synthetic training data, simulating a wide range of potential mappings that may exist in such real-world scenarios. The experimental results further demonstrate the applicability and generalization of our method. However, when confronted with problems that involve special knowledge completely beyond GP priors, our method may struggle to generalize to these specific types of cases. Potential solutions include incorporating expert knowledge to design specialized priors during synthetic data generation, or interacting with the evaluation environment during optimization to discover and enrich potential priors. Since synthetic data can be generated from the designed priors through extensive sampling, these enhancements can be seamlessly integrated into our existing method. Furthermore, instead of permanently freezing the parameters after a single round of synthetic pre-training, it may be advantageous to fine-tune the pre-trained model using a small amount of domain-specific data.

A.2 Scalability Challenges

Although we have evaluated the scalability of our approach across problems with varying feature and objective spaces (see Section E.3), those involving significantly higher-dimensional features or objectives pose considerable challenges. This scalability issue is also widely recognized as one of the main difficulties within the multi-objective optimization community. To this end, several promising avenues worth further exploration. In the context of high-dimensional feature spaces, the development of efficient dimensionality reduction methods and embedding techniques, along with effective search strategies such as the trust region Bayesian optimization (TuRBO) algorithm (Eriksson et al. 2019), presents significant potential. For problems involving a large number of objectives, the number of solutions required to approximate the entire Pareto set increases exponentially, resulting in significant overhead for both optimization and decision-making. Therefore, rather than pursuing a comprehensive set of Pareto solutions, recent work has focused on identifying a small set of representative solutions, ensuring that each objective can be well addressed by at least one solution within the set (Lin et al. 2024; Liu et al. 2024). This motivates the development of foundational models and parallel acquisition functions based on the set scalarizations (Lin et al. 2024).

B Potential Societal Impact

The proposed method may yield the following positive impacts:

- It provides a general paradigm that can quickly adapt to numerous expensive multi-objective challenges continuously emerging in the real world, enabling users to efficiently identify superior solutions in a sample-efficient manner while minimizing the expenditure of social resources.
- The generation of the training corpus does not require access to any real-world data, showcasing significant potential for a variety of real-world scenarios where costly experimental data is severely limited or where user privacy and commercial copyright issues are critical.
- Our approach offers a user-friendly and flexible interactive platform. Users can receive uncertainty-guided feedback on any sub-problem according to their individual preferences, allowing them to explore and exploit customized solutions that better align with their needs.

On the negative side, we acknowledge that we have invested considerable computational resources in implementing the algorithm framework and tuning the hyperparameters of the pre-trained network. However, we believe that our work will significantly lower the foreseeable design barriers and computational costs for practitioners in the future, which is one of the primary goals of our work. We hope that the societal benefits across various fields from directly adapting our method to few-shot scenarios, or even fine-tuning it with limited domain knowledge, will far outweigh the costs incurred in our efforts.

C Model and Algorithm Details

C.1 Details of Foundation Model and Pre-training

Our foundational model mainly employs the recently proposed Prior-Data Fitted Networks (PFNs) (Müller et al. 2021) as the basic architecture, which is based on a Transformer encoder (Vaswani et al. 2017). We utilize linear layers to encode trajectory and query points, along with sampled preferences, as contextual inputs for the foundation model. Following the methods outlined in (Hollmann et al. 2022) and (Müller et al. 2023), we set a maximum dimension K for encoding the inputs, which are applicable to both feature and objective dimensions. We zero-pad the sampled dimension k and linearly scale the dimension by a factor of K/k to ensure that the magnitudes of the input encodings are consistent across different dimensions. Since different query points should not depend on one another, an attention mask is implemented to allow each trajectory point can only attend to each other, while the query points can only attend to the trajectory points. To maintain the permutation invariance of the

Table 2: Detailed information for the foundation model and pre-training process, including the hyperparameters and corresponding values of the model architecture, data generation and model training.

Type	Name	Value
Model Architecture	Embedding Size	512
	Hidden Size	1024
	Attention Heads	4
	Transformer Layers	12
	Output Logits Size	1000
Data Generation	GP Mean Prior	0.0
	GP Kernel	RBF
	Output Scale	1.0
	Output Noise	$\mathcal{N}(0, 10^{-4})$
	Length Scale Prior	$\Gamma(\alpha = 3.0, \beta = 6.0)$
	Maximum Feature Dimension	30
	Maximum Objective Dimension	6
	Maximum Sample Length	128
Model Training	Batch Size	256
	Steps per Epoch	2048
	Linear-warmup Epochs	5
	Total Epochs	500
	Learning Rate	0.0001

input dataset, the positional encodings are removed and the input points are fed to the model as the sum of their encoded embeddings. During training, we maintain a maximum sample length of N with n trajectory points and $N - n$ query points, as used in (Müller et al. 2021). We sample each n from $\{1, ..., N-1\}$ with the weight of $1/(N-n)$ to simulate varying trajectory sizes during the optimization process, while ensuring that each trajectory size has approximately the same number of query points for training. To model the continuous distribution for prediction, the regression head of PFNs discretizes the continuous distribution into a piecewise constant distribution, referred to as the Riemann distribution (Müller et al. 2021). This helps transform the model training from a regression task to a classification problem. The boundaries of each discrete interval are estimated by sampling a large synthetic dataset from the same synthetic prior, ensuring that each interval maintains equal probability in the prior data. Additionally, the last bar on each side is replaced with a suitably scaled half-normal distribution to enhance training stability.

During the pre-training phase, we extensively sample from a data generation mechanism to create a large and diverse set of synthetic data. Initially, we sample the number of features d, the number of objectives m, and the trajectory length n from 1 up to their predetermined maximum values. We then sample m independent latent functions from a Gaussian Process (GP) prior with zero mean and the radial basis function (RBF) kernel:

$$f \sim \mathcal{GP}(0, k_{\text{RBF}}(\cdot, \cdot)), \quad k_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2l^2}), \tag{9}$$

where σ and l represent the output scale and length scale kernel parameters, respectively. We also sample the length scale parameter for each feature dimension during the latent function sampling, along with an random observation noise to enhance function diversity. In the third step, we randomly sample input features from the d-dimensional unit cube, and then evaluate the sampled latent functions to generate the corresponding observations, resulting in n trajectory samples and $N - n$ query samples. After that, we randomly sample the artificial preferences from the simplex $\Delta = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m \mid \|\boldsymbol{\lambda}\|_1 = 1\}$ to derive the aggregation targets g given query inputs and preferences. The aggregation targets are then masked to formulate a supervised prediction problem, enabling the model to learn to predict the preference-wise aggregated posterior distribution $q_{\boldsymbol{\theta}}(g|\boldsymbol{x}, D_n; \boldsymbol{\lambda})$. We continuously synthesize data through extensive sampling and optimize the model parameters to minimize the cross-entropy loss function defined in Equation (5), which is equivalent to the expected KL divergence between the true posterior distribution and its approximation, up to an additive constant (Müller et al. 2021). The detailed information and hyperparameter settings regarding the foundational model and the pre-training process can be found in Table 2.

C.2 Visualization of Aggregated Posteriors

For the convenience of visualization, we use a simple one-dimensional ToyRobust problem (Daulton et al. 2022) as an example to demonstrate that the pre-trained foundation model can effectively learn to infer the aggregated posteriors on unknown problems, as illustrated in Figure 5. The results show that the foundation model provides reliable predictions given the evaluated

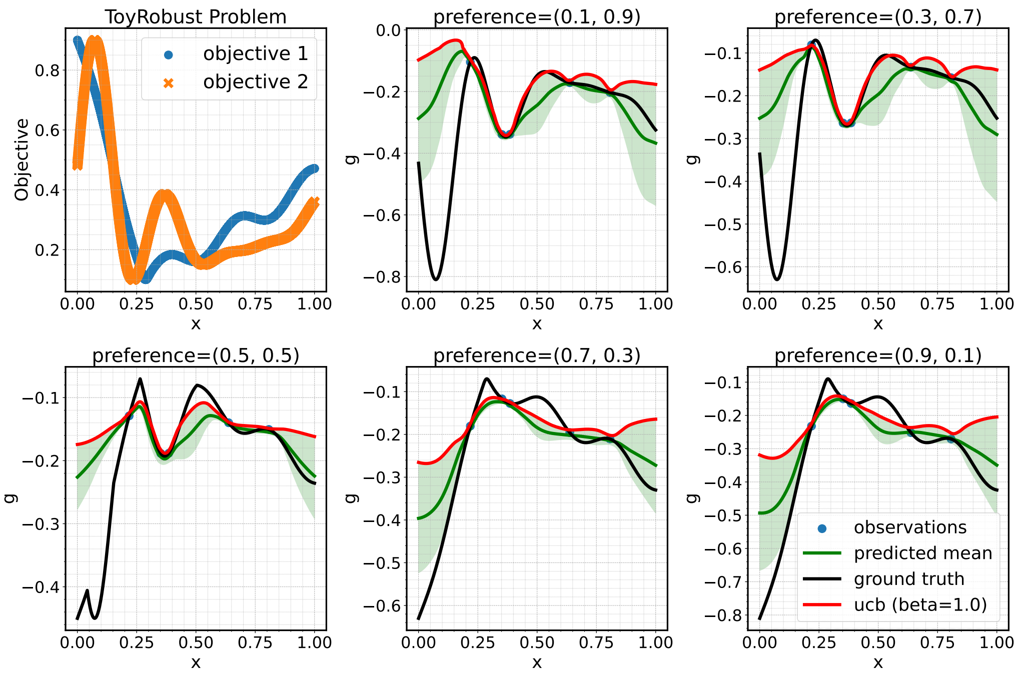

Figure 5: Details of the normalized objective functions for the ToyRobust problem, the aggregated posteriors (green solid lines with shaded regions) conditioned on 5 observations and 5 preferences, and the corresponding UCB acquisition functions (red solid lines).

trajectory of the out-of-distribution problem without any model training and update, allows the model to generalize beyond the synthetic training data. The subproblems defined by adjacent preferences exhibit similar characteristics in terms of the scalarization functions, the predicted mean captures this similarity effectively. Furthermore, the prediction provides a reliable measure of calibration uncertainty, enabling the acquisition function such as upper confidence bound (UCB) shown in the figure to effectively balance exploration and exploitation during the search for candidate solutions.

C.3 Derivation of UHVI Acquisition Function

Definitions and preliminaries. The derivation mainly depends on two types of definitions on Tchebycheff scalarizations and the theorem on hypervolume scalarization. We consider the the following multi-objective optimization problem (MOP):

$$\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})). \tag{10}$$

We denote $D_n = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$ be a set of n evaluated samples, where each $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$ is the feature vector in the decision space $\mathcal{X} \subset \mathbb{R}^d$, and each $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^T$ is the observation vector of the true objective function $\boldsymbol{f}(\boldsymbol{x})$. We denote $\boldsymbol{z}^* = (z_1^*, \cdots, z_m^*)^T$ as the ideal point (lower bound) in the objective space. We have the Tchebycheff scalarization function:

$$s_{\boldsymbol{\lambda}}^{\text{tch}}(\boldsymbol{x}) = \max_{1\le j\le m} \{\lambda_j(y_j(\boldsymbol{x}) - z_j^*)\}, \tag{11}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ represents a preference sampled from the simplex $\Delta = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m \mid \|\boldsymbol{\lambda}\|_1 = 1\}$, we can identify each Pareto optimal solution by minimizing $s_{\boldsymbol{\lambda}}^{\text{tch}}(\boldsymbol{x})$ with a specific but unknown preference (Choo and Atkins 1983).

The Tchebycheff scalarization function can be transformed in another formulation with respect to a set of reference vectors $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)^T$ sampled from $S = \{\boldsymbol{w} \in \mathbb{R}_+^m \mid \|\boldsymbol{w}\|_2 = 1\}$:

$$s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}) = \max_{1\le j\le m} \{(y_j(\boldsymbol{x}) - z_j^*)/w_j\}. \tag{12}$$

Leveraging Equation (12), we have the following theorem, which will help us approximate the hypervolume indicator and further derive UHVI acquisition function.

Theorem 1 (Hypervolume Scalarization)**.** *Let $D_n = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$ be a set of n evaluated samples, $\boldsymbol{z}^* = (z_1^*, \cdots, z_m^*)^T$ is the ideal point, $\boldsymbol{r} = (r_1, \cdots, r_m)^T$ is a pre-defined reference point. The hypervolume of the evaluated samples can be expressed* *as an expectation of Tchebycheff scalarization function over a set of uniformly sampled reference vectors* $S = \{\boldsymbol{w} \in \mathbb{R}^m_+ \mid \|\boldsymbol{w}\|_2 = 1\}$ *(Deng and Zhang 2019; Zhang and Golovin 2020; Zhang et al. 2023):*

$$\text{HV} = \prod_{j=1}^{m}(r_j - z_j^*) - c_m \mathbb{E}_{\boldsymbol{w}\sim S}\{[\min_{1\leq i\leq n} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m\}, \tag{13}$$

where $c_m = \frac{\pi^{m/2}}{2^m\Gamma(m/2+1)}$ *is a constant that depends only on* m*.*

Utilizing the hypervolume scalarization and Equation (11), we can naturally derive the utility of UHVI acquisition function:

Lemma 1 (Hypervolume Improvment with Uncertainty)**.** *Given the the distribution of the aggregation target* $g_{\boldsymbol{\lambda}}(\cdot) = -s_{\boldsymbol{\lambda}}^{\text{tch}}(\cdot)$ *conditioned on evaluated samples* $D_n = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$*, the utility of hypervolume improvement in the normalized objective space at any query point* $\boldsymbol{x}$ *can be estimated by the following expectation:*

$$\alpha_{\text{UHVI}}(\boldsymbol{x}) = c_m \mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m \max\{0, [\min_{1\leq i\leq n}\{-g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i)\}]^m - [-\text{UCB}(g_{\boldsymbol{\lambda}}(\boldsymbol{x}))]^m\}], \tag{14}$$

where $c_m = \frac{\pi^{m/2}}{2^m\Gamma(m/2+1)}$ *is a constant that depends only on* m*,* $c_{\boldsymbol{\lambda}} = \sqrt{\sum_{j=1}^m \frac{1}{\lambda_j^2}}$ *is a transformation constant that depends only on* $\boldsymbol{\lambda}$*,* $\text{UCB}(\cdot)$ *denotes the upper confidence bound (UCB) metric.*

Proof. From Equation (13), given an updated evaluated samples $D_{n+1} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{n+1}$ with newly added sample $\{\boldsymbol{x}_{n+1}, \boldsymbol{y}_{n+1}\}$, the hypervolume can be represented as:

$$\text{HV}' = \prod_{j=1}^{m}(r_j - z_j^*) - c_m \mathbb{E}_{\boldsymbol{w}\sim S}\{[\min_{1\leq i\leq n+1} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m\}, \tag{15}$$

which induces the hypervolume improvement:

$$\begin{aligned}
\text{HVI}(\boldsymbol{x}_{n+1}) &= \text{HV}' - \text{HV} \\
&= c_m \mathbb{E}_{\boldsymbol{w}\sim S}\{[\min_{1\leq i\leq n} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m - [\min_{1\leq i\leq n+1} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m\} \\
&= c_m \mathbb{E}_{\boldsymbol{w}\sim S}\{[\min_{1\leq i\leq n} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m - [\min\{\min_{1\leq i\leq n} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i), s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_{n+1})\}]^m\} \\
&= c_m \mathbb{E}_{\boldsymbol{w}\sim S}[\max\{0, [\min_{1\leq i\leq n} s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_i)]^m - [s_{\boldsymbol{w}}^{\text{tch}}(\boldsymbol{x}_{n+1})]^m\}].
\end{aligned} \tag{16}$$

Given the transformation constant $c_{\boldsymbol{\lambda}} = \sqrt{\sum_{j=1}^m \frac{1}{\lambda_j^2}}$, each reference vector $\boldsymbol{w}$ can be uniquely represented by the preference $\boldsymbol{\lambda}$ by using $w_j = 1/(\lambda_j c_{\boldsymbol{\lambda}})$, the hypervolume improvement can be represented as:

$$\text{HVI}(\boldsymbol{x}_{n+1}) = c_m \mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m \max\{0, [\min_{1\leq i\leq n} s_{\boldsymbol{\lambda}}^{\text{tch}}(\boldsymbol{x}_i)]^m - [s_{\boldsymbol{\lambda}}^{\text{tch}}(\boldsymbol{x}_{n+1})]^m\}]. \tag{17}$$

Our model predict the distribution of the aggregation target $g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i) = -s_{\boldsymbol{\lambda}}^{\text{tch}}(\boldsymbol{x}_i)$ given each $\boldsymbol{x}_i$. To ensure that objectives of varying magnitudes share the same scale, we normalize the objectives to the range of [0,1] during the optimization process. Consequently, the hypervolume improvement in the normalized objective space can be reformulated as follows:

$$\text{HVI}(\boldsymbol{x}_{n+1}) = c_m \mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m \max\{0, [\min_{1\leq i\leq n}\{-g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i)\}]^m - [-g_{\boldsymbol{\lambda}}(\boldsymbol{x}_{n+1})]^m\}]. \tag{18}$$

Here the aggregation target $g_{\boldsymbol{\lambda}}(\cdot)$ is constrained to be non-positive after the objective normalization (see Figure 5 for an example). Since we wish to maximize the aggregation target at the query point $\boldsymbol{x}_{n+1}$ for maximizing hypervolume improvement, we replace the prediction $g_{\boldsymbol{\lambda}}(\boldsymbol{x}_{n+1})$ with the $\text{UCB}(\cdot)$ metric as an upper bound estimation, which introduces the calibrated uncertainty provided by our model for searching the candidates with a good balance of exploration and exploitation:

$$\hat{\text{HVI}}(\boldsymbol{x}_{n+1}) = c_m \mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m \max\{0, [\min_{1\leq i\leq n}\{-g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i)\}]^m - [-\text{UCB}(g_{\boldsymbol{\lambda}}(\boldsymbol{x}_{n+1}))]^m\}]. \tag{19}$$

We can then derive the utility of UHVI acquisition function as $\boldsymbol{x}_{n+1}$ is arbitrary:

$$\alpha_{\text{UHVI}}(\boldsymbol{x}) = c_m \mathbb{E}_{\boldsymbol{\lambda}\sim\Delta}[(c_{\boldsymbol{\lambda}})^m \max\{0, [\min_{1\leq i\leq n}\{-g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i)\}]^m - [-\text{UCB}(g_{\boldsymbol{\lambda}}(\boldsymbol{x}))]^m\}], \tag{20}$$

which leads to our claim in the lemma. □

Intuitively, maximizing α_{UHVI} utility is equivalent to maximizing the predicted upper bounds of aggregation targets across all preference simultaneously. We find that our utility exhibits similar expression to that presented in (Song et al. 2024b). However, the key difference is that the formulation of (Song et al. 2024b) is based on the Gaussian process (GP) posterior for each objective, whereas our acquisition function is directly derived from infinite preference-wise aggregated posteriors, which is intractable for traditional GP training and updates.

Table 3: Detailed information for synthetic problems and real-world applications.

Name	d	m	Type	Pareto Front Shape
ZDT1	8	2	Continuous	Convex
ZDT2	8	2	Continuous	Concave
ZDT3	8	2	Continuous	Disconnected
ZDT4	8	2	Continuous	Convex
ZDT6	8	2	Continuous	Concave
Omnitest	2	2	Continuous	Convex
RE21 (Four bar truss design)	4	2	Continuous	Convex
RE22 (Reinforced concrete beam design)	3	2	Mixed	Mixed
RE23 (Pressure vessel design)	4	2	Mixed	Mixed, Disconnected
RE24 (Hatch cover design)	2	2	Continuous	Convex
RE25 (Coil compression spring design)	3	2	Mixed	Mixed, Disconnected
RE31 (Two bar truss design)	3	3	Continuous	Unknown
RE32 (Welded beam design)	4	3	Continuous	Unknown
RE33 (Disc brake design)	4	3	Continuous	Unknown
RE34 (Vehicle crashworthiness design)	5	3	Continuous	Unknown
RE35 (Speed reducer design)	7	3	Mixed	Unknown
RE36 (Gear train design)	4	3	Integer	Concave, Disconnected
RE37 (Rocket injector design)	4	3	Continuous	Unknown

D Experiment Setting Details

D.1 Details of Benchmark

We firstly utilize synthetic problems to evaluate the effectiveness of our method. The tested synthetic problems include ZDT1-4, ZDT6 (Zitzler, Deb, and Thiele 2000) and Omnitest (Deb and Tiwari 2008) test suits. We also conduct experiments on 12 real-world applications from the popular RE test suit (Tanabe and Ishibuchi 2020), which consists multi-objective engineering design problems from various fields with different characteristics. The detailed information can be referred to Table 3.

D.2 Details of Implementation

In many real-world scenarios, the ranges of each objective can vary significantly, sometimes even by orders of magnitude, making Gaussian Process modeling and subsequent optimization processes challenging. To this end, before executing each algorithm, we normalize each i-th objective observations $y_i(\boldsymbol{x})$ based on the evaluated samples $D_n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ using $(y_i(\boldsymbol{x}) - y_i^{\text{min}})/(y_i^{\text{max}} - y_i^{\text{min}})$, where $y_i^{\text{min}} = \min_{\boldsymbol{x}\in\mathcal{D}_n} y_i(\boldsymbol{x})$ and $y_i^{\text{max}} = \max_{\boldsymbol{x}\in\mathcal{D}_n} y_i(\boldsymbol{x})$. When assessing the quality of the candidate solutions obtained after the optimization is completed, we use the true objective observations to calculate the IGD+ or normalized hypervolume metrics, as detailed in Section D.3.

D.3 Details of Performance Metrics

Since the tested synthetic functions have clear analytical expressions with known Pareto fronts, we use the inverted generational distance plus (IGD+) indicator (Ishibuchi et al. 2015) as the metric to evaluate the quality of the final solutions obtained by each algorithm. Let $\mathcal{P}$ be the approximated Pareto front obtained by the tested algorithm, and $\mathcal{P}^*$ be a set of well-distributed objective vectors sampled from the true Pareto front. The definition of IGD+ is as follows, indicating a smaller IGD+ value represents the better performance of an algorithm:

$$\text{IGD}^+(\mathcal{P}^*, \mathcal{P}) = \frac{1}{|\mathcal{P}^*|} \sum_{\boldsymbol{v}\in\mathcal{P}^*} \min_{\boldsymbol{u}\in\mathcal{P}} \sqrt{\sum_{i=1}^{m} ([u_i - v_i]_+)^2}. \tag{21}$$

For real-world applications, the true Pareto fronts are typically unknown. For the convenience of comparing the performance of different algorithms, we utilize the approximate Pareto fronts provided by (Tanabe and Ishibuchi 2020) as the ground truth, which are generated with a large number of evaluations and have also been widely used in other related studies. To ensure fairness, we firstly generate the ideal point $\boldsymbol{y}_{\text{ideal}}$ and nadir point $\boldsymbol{y}_{\text{nadir}}$ from the approximate Pareto front of the tested problem:

$$\begin{aligned} &\boldsymbol{y}_{\text{ideal}} = (\min y_1(\boldsymbol{x}_1), \ldots, \min y_m(\boldsymbol{x}_m))^T, \\ &\boldsymbol{y}_{\text{nadir}} = (\max y_1(\boldsymbol{x}_1), \ldots, \max y_m(\boldsymbol{x}_m))^T, \\ &\forall \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \mathcal{P}_s, \end{aligned} \tag{22}$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ are solutions in approximate Pareto set $\mathcal{P}_s$ from the ground truth. After that, we normalize the resulting Pareto fronts obtained from the algorithms using the bounds of ideal point and nadir point. Finally, we set the reference point as $\boldsymbol{r}_i = 1.1, i = \{1, 2, ..., m\}$ to calculate the hypervolume indicator of the normalized Pareto front obtained by each algorithm.

E Additional Experimental Results

E.1 Training Loss

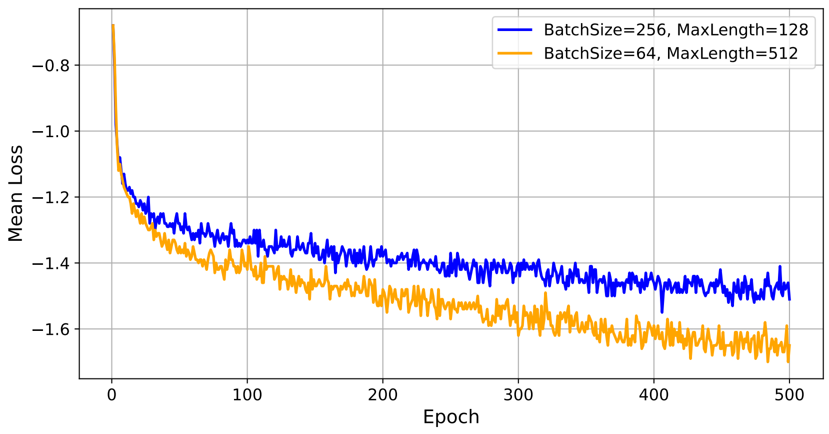

Figure 6: The mean loss convergences with the same data volume but different hyperparameter combinations.

We monitore the convergence of the mean loss during training presented by the model we used in this paper, represented by the blue solid line in Figure 6. Despite minor fluctuations, the loss curve shows that it has essentially converged. We further test the loss convergence by reducing the batch size and increasing the maximum sample length while maintaining the same amount of data. This adjustment results in the loss converging to a smaller level.

E.2 Time Usage During Model Training

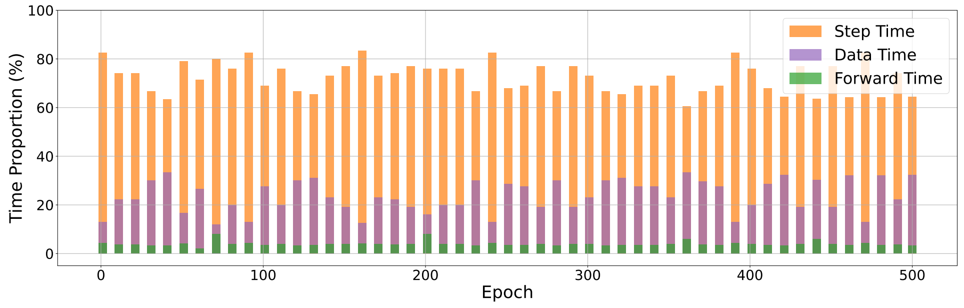

Figure 7: The proportion of time usage during model training (recorded every 10 epochs).

In our method, the data synthesis and model training are performed simultaneously. To evaluate the advantages of using synthetic data for pre-training, we monitored the proportion of time utilized during training, as illustrated in Figure 7. In the figure, ”data time” refers to the time spent synthesizing data for each epoch, ”forward time” indicates the duration spent on model inference to generate logits for loss calculation, and ”step time” denotes the time taken to compute the loss and perform gradient backpropagation to optimize the model parameters. Statistics results indicate that the model optimization and data synthesis account for the majority of pre-training time. Although a large amount of data is synthesized during training using our sampling mechanism, the time proportion spent on data synthesis is notably smaller than the time dedicated to optimizing model parameters. This efficiency can be largely attributed to the effective computation of the decomposition-based aggregation targets from extensive synthetic functions, which does not impose an unacceptable burden on pre-training as would be the case with expensive experimental data from real-world problems. Furthermore, it is noteworthy that a single forward propagation through the neural network is quite fast, accounting for only about 5% of the total time. This efficiency helps explain the rapid in-context inference observed once the pre-training is completed.

E.3 Dimensional Scalability

We further evaluate the scalability of our method in higher-dimensional spaces, including both feature and objective dimensions. We consider ZDT1-3 problems with a feature dimension of 30, as they readily allow for dimensional expansion in their problem formulations. In addition, we examine more challenging real-world problems with higher objective dimensions, including RE41 (car side impact design) and RE42 (conceptual marine design), both with 4 objectives, as well as RE61 (water resource planning), which has 6 objectives (Tanabe and Ishibuchi 2020). The experimental results are shown in Figure 8 and Table 4. Our method demonstrates strong adaptability and competitive performance on problems characterized by high-dimensional complex feature and objective spaces. Notably, the EI and UCB methods show clear advantages in synthetic problems. Despite being constrained by a small budget in the RE42 problem, their reported hypervolume metrics continue to improve compared to other methods. Although our UHVI performs slightly worse than EI and UCB, it remains competitive compared with the qParEGO method in most cases.

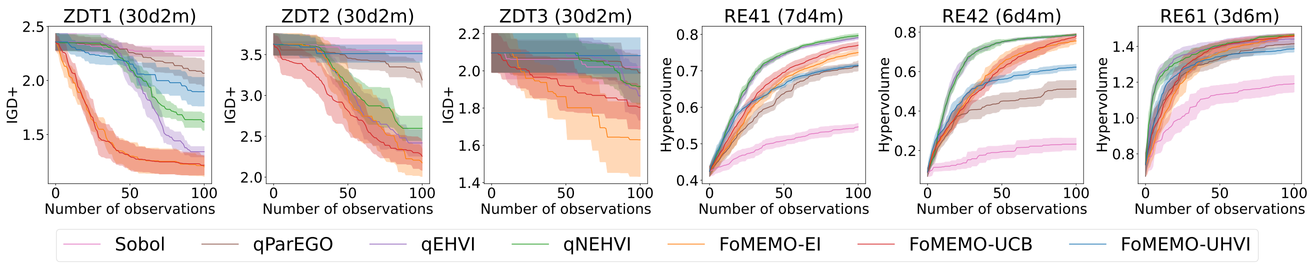

Figure 8: Sequential optimization performance of optimization metrics w.r.t. the number of observations (beyond initial points) on ZDT1-3 problems with 30 features, real-world applications RE41-42 with 4 objectives and RE61 with 6 objectives. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

Table 4: Means (stds) of best optimization metrics in sequential optimization on ZDT1-3 problems with 30 features, real-world applications RE41-42 with 4 objectives and RE61 with 6 objectives over 10 independent runs. The best and second best mean results for each row are highlighted with shaded background and underlines respectively.

Test Instances	Sobol	qParEGO	qEHVI	qNEHVI	FoMEMO-EI	FoMEMO-UCB	FoMEMO-UHVI
ZDT1 (30d2m)	2.271e+00(4.83e-02)	2.067e+00(1.19e-01)	1.341e+00(4.81e-02)	1.617e+00(7.89e-02)	1.202e+00(9.12e-02)	1.215e+00(9.11e-02)	1.896e+00(1.34e-01)
ZDT2 (30d2m)	3.541e+00(1.18e-01)	3.189e+00(9.29e-02)	2.418e+00(1.63e-01)	2.597e+00(1.51e-01)	2.175e+00(1.68e-01)	2.255e+00(1.72e-01)	3.512e+00(1.08e-01)
ZDT3 (30d2m)	2.021e+00(9.22e-02)	1.989e+00(8.86e-02)	1.866e+00(1.31e-01)	1.916e+00(9.50e-02)	1.629e+00(1.97e-01)	1.804e+00(1.18e-01)	2.082e+00(9.66e-02)
RE41 (7d4m)	5.462e-01(1.02e-02)	7.135e-01(1.41e-02)	7.905e-01(4.99e-03)	7.966e-01(5.92e-03)	7.503e-01(8.21e-03)	7.701e-01(9.51e-03)	7.148e-01(4.63e-03)
RE42 (6d4m)	2.328e-01(3.07e-02)	5.118e-01(4.33e-02)	7.850e-01(6.08e-03)	7.874e-01(6.08e-03)	7.554e-01(1.55e-02)	7.797e-01(1.37e-02)	6.224e-01(1.48e-02)
RE61 (3d6m)	1.192e+00(4.63e-02)	1.415e+00(2.20e-02)	1.441e+00(3.14e-02)	1.463e+00(6.98e-03)	1.461e+00(6.29e-03)	1.457e+00(1.52e-02)	1.387e+00(1.85e-02)

E.4 Parallel Scalability

We provide supplementary experimental results to further validate the scalability of our approach for parallel optimization. Results include the performance of all algorithms on synthetic problems with candidate batch sizes q=5 (shown in Figure 9 and Table 5) and q=10 (shown in Figure 10 and Table 6), as well as the performance of all algorithms on real-world applications with q=10 (shown in Figure 11 and Table 7). In addressing synthetic problems, our method demonstrates competitiveness with existing approaches, notably the UCB method for its robustness. Moreover, our method showcases scalability to high batch sizes of candidate solutions in various real-world applications, underscoring its potential value in diverse scenarios that allow for efficient batch experimentation. It is worth noting that the performance of our method diminishes as the batch size of candidates q increases. This decline is attributed to our use of a relatively trivial parallelization strategy for searching batches of candidate solutions. More efficient batch search methods merit further exploration to enhance performance in larger batch scenarios.

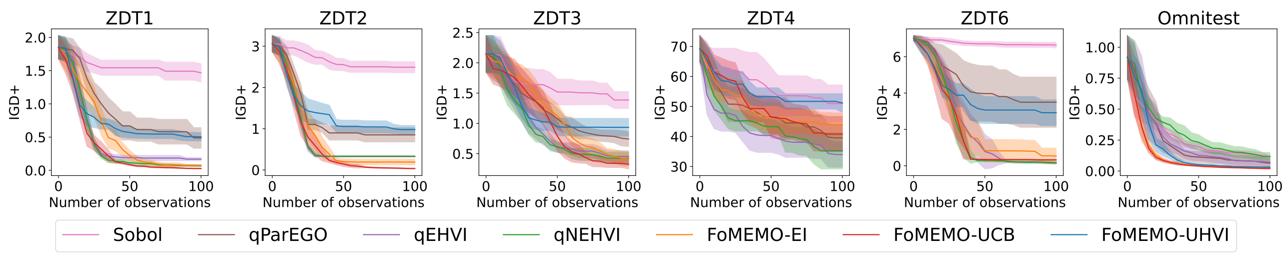

Figure 9: Parallel optimization performance (q=5) of IGD+ metrics (↓) w.r.t. the number of observations (beyond initial points) on synthetic problems. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

Table 5: Means (stds) of best IGD+ metrics (↓) in parallel optimization (q=5) on synthetic problems over 10 independent runs. The best and second best mean results for each row are highlighted with shaded background and underlines respectively.

Test Instances	Sobol	qParEGO	qEHVI	qNEHVI	FoMEMO-EI	FoMEMO-UCB	FoMEMO-UHVI
ZDT1	1.466e+00(1.38e-01)	4.873e-01(1.55e-01)	1.686e-01(2.38e-02)	6.864e-02(1.48e-02)	7.055e-02(1.48e-02)	2.711e-02(5.30e-03)	5.046e-01(6.35e-02)
ZDT2	2.490e+00(1.33e-01)	8.501e-01(1.72e-01)	3.249e-01(9.56e-03)	3.350e-01(3.44e-17)	1.920e-01(6.87e-02)	3.497e-02(9.45e-03)	9.776e-01(1.02e-01)
ZDT3	1.386e+00(1.40e-01)	7.417e-01(1.25e-01)	4.544e-01(9.44e-02)	4.232e-01(9.79e-02)	4.152e-01(1.17e-01)	3.264e-01(7.84e-02)	9.348e-01(1.48e-01)
ZDT4	5.098e+01(6.19e+00)	3.953e+01(4.18e+00)	3.400e+01(4.24e+00)	3.523e+01(6.03e+00)	4.399e+01(3.84e+00)	4.079e+01(5.87e+00)	5.115e+01(3.08e+00)
ZDT6	6.638e+00(1.59e-01)	3.497e+00(1.38e+00)	1.615e-01(5.29e-02)	1.631e-01(6.78e-02)	5.363e-01(4.32e-01)	3.239e-01(2.81e-02)	2.919e+00(7.10e-01)
Omnitest	1.154e-01(1.39e-02)	6.341e-02(3.54e-02)	7.159e-02(3.31e-02)	1.170e-01(3.26e-02)	2.262e-02(1.78e-03)	1.970e-02(2.76e-03)	3.103e-02(3.65e-03)

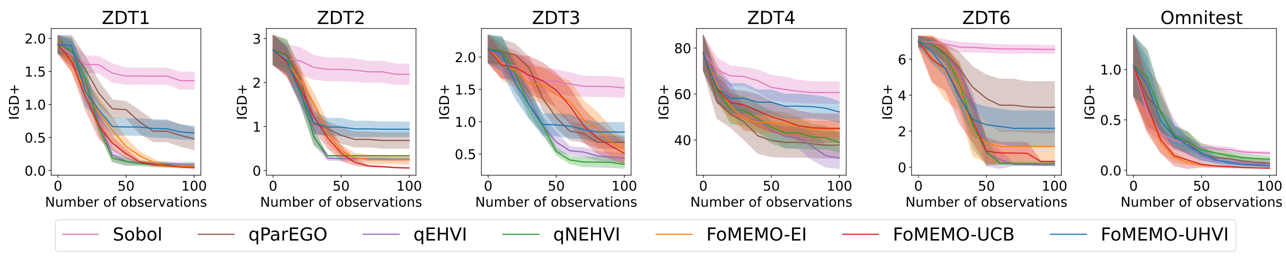

Figure 10: Parallel optimization performance (q=10) of IGD+ metrics (↓) w.r.t. the number of observations (beyond initial points) on synthetic problems. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

Table 6: Means (stds) of best IGD+ metrics (↓) in parallel optimization (q=10) on synthetic problems over 10 independent runs. The best mean result for each row is highlighted with shaded background.

Test Instances	Sobol	qParEGO	qEHVI	qNEHVI	FoMEMO-EI	FoMEMO-UCB	FoMEMO-UHVI
ZDT1	1.360e+00(1.30e-01)	4.789e-01(1.67e-01)	9.813e-02(2.38e-02)	6.958e-02(1.60e-02)	6.052e-02(1.06e-02)	4.346e-02(1.35e-02)	5.686e-01(1.01e-01)
ZDT2	2.187e+00(2.32e-01)	6.831e-01(1.80e-01)	2.592e-01(4.67e-02)	3.355e-01(3.14e-03)	2.468e-01(1.01e-01)	5.722e-02(1.22e-02)	9.372e-01(1.68e-01)
ZDT3	1.523e+00(1.51e-01)	6.811e-01(9.58e-02)	4.376e-01(6.99e-02)	3.387e-01(6.37e-02)	5.666e-01(1.57e-01)	5.108e-01(1.62e-01)	8.398e-01(1.49e-01)
ZDT4	6.065e+01(4.73e+00)	3.774e+01(5.73e+00)	3.208e+01(4.53e+00)	3.878e+01(4.05e+00)	4.474e+01(3.42e+00)	4.508e+01(6.25e+00)	5.215e+01(4.47e+00)
ZDT6	6.544e+00(2.23e-01)	3.330e+00(1.42e+00)	1.199e-01(5.66e-02)	1.978e-01(7.44e-02)	1.145e+00(8.73e-01)	3.109e-01(4.75e-02)	2.160e+00(9.74e-01)
Omnitest	1.706e-01(1.46e-02)	6.888e-02(3.79e-02)	6.188e-02(2.08e-02)	1.101e-01(3.16e-02)	2.227e-02(3.86e-03)	2.061e-02(2.36e-03)	4.201e-02(5.41e-03)

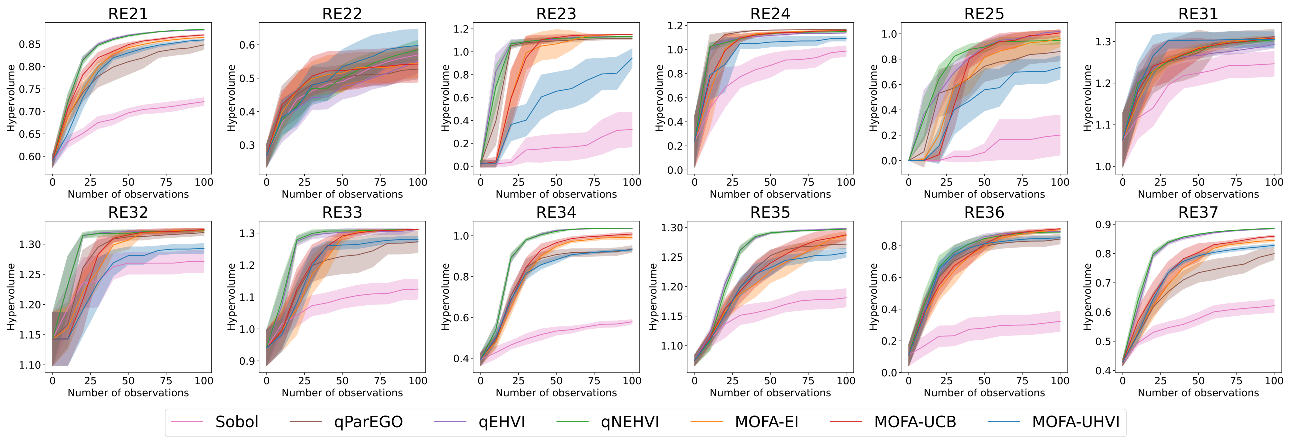

Figure 11: Parallel optimization performance (q=10) of hypervolume metrics (↑) w.r.t. the number of observations (beyond initial points) on real-world applications. The solid line is the mean value averaged over 10 independent runs, and the shaded region is the standard deviation with 95% confidence around the mean value.

Table 7: Means (stds) of best hypervolume metrics (↑) in parallel optimization (q=10) on real-world applications over 10 independent runs. The best and second best mean results for each row are highlighted with shaded background and underlines respectively.

Test Instances	Sobol	qParEGO	qEHVI	qNEHVI	FoMEMO-EI	FoMEMO-UCB	FoMEMO-UHVI
RE21	7.217e-01(9.10e-03)	8.481e-01(1.05e-02)	8.813e-01(4.48e-04)	8.823e-01(1.35e-04)	8.647e-01(1.59e-03)	8.701e-01(1.39e-03)	8.592e-01(2.11e-03)
RE22	5.695e-01(2.01e-02)	5.272e-01(3.00e-02)	5.515e-01(3.69e-02)	5.852e-01(2.88e-02)	5.461e-01(4.45e-02)	5.422e-01(5.44e-02)	5.978e-01(4.80e-02)
RE23	3.216e-01(1.53e-01)	1.116e+00(1.05e-02)	1.134e+00(3.43e-03)	1.132e+00(6.00e-03)	1.152e+00(2.89e-03)	1.150e+00(3.06e-03)	9.437e-01(8.27e-02)
RE24	9.873e-01(4.37e-02)	1.166e+00(9.05e-04)	1.138e+00(6.54e-03)	1.149e+00(5.45e-03)	1.156e+00(3.84e-03)	1.156e+00(3.01e-03)	1.090e+00(2.08e-02)
RE25	2.011e-01(1.58e-01)	8.600e-01(7.47e-02)	1.018e+00(1.09e-02)	9.487e-01(2.99e-02)	9.626e-01(6.56e-02)	1.007e+00(2.71e-02)	7.345e-01(9.27e-02)
RE31	1.246e+00(2.97e-02)	1.312e+00(1.72e-02)	1.293e+00(9.40e-03)	1.304e+00(3.66e-03)	1.308e+00(5.46e-03)	1.309e+00(3.07e-03)	1.304e+00(1.91e-02)
RE32	1.271e+00(1.82e-02)	1.319e+00(4.92e-03)	1.324e+00(3.03e-03)	1.322e+00(1.90e-03)	1.323e+00(2.85e-03)	1.324e+00(2.00e-03)	1.293e+00(8.26e-03)
RE33	1.125e+00(3.16e-02)	1.273e+00(3.43e-02)	1.311e+00(6.10e-04)	1.311e+00(9.34e-04)	1.311e+00(9.51e-04)	1.312e+00(8.16e-04)	1.281e+00(9.48e-03)
RE34	5.787e-01(1.08e-02)	9.348e-01(1.69e-02)	1.036e+00(5.77e-04)	1.037e+00(8.88e-04)	9.920e-01(9.27e-03)	1.008e+00(8.27e-03)	9.308e-01(6.52e-03)
RE35	1.181e+00(1.56e-02)	1.272e+00(7.54e-03)	1.298e+00(7.60e-04)	1.297e+00(1.10e-03)	1.287e+00(6.37e-03)	1.288e+00(8.94e-03)	1.257e+00(8.02e-03)
RE36	3.233e-01(6.46e-02)	8.424e-01(3.12e-02)	8.912e-01(1.00e-02)	8.865e-01(6.89e-03)	9.016e-01(5.60e-03)	9.073e-01(4.82e-03)	8.502e-01(1.17e-02)
RE37	6.216e-01(2.30e-02)	7.996e-01(2.07e-02)	8.850e-01(1.10e-03)	8.865e-01(8.78e-04)	8.454e-01(5.26e-03)	8.591e-01(4.19e-03)	8.279e-01(6.17e-03)