TOWARDS UNIVERSAL OFFLINE BLACK-BOX OPTIMIZATION VIA LEARNING STRING EMBEDDING SPACE

Rong-Xi Tan[*], **Ming Chen**[*] **& Ke Xue**[†]
National Key Laboratory for Novel Software Technology, Nanjing University
School of Artificial Intelligence, Nanjing University
Nanjing, China
{rongxitan,221300059}@smail.nju.edu.cn, xuek@lamda.nju.edu.cn

Yao Wang , Yaoyuan Wang & Sheng Fu
Advanced Computing and Storage Lab, Huawei Technologies Co., Ltd.
Shenzhen, China
{wangyao123,wangyaoyuan1,fusheng.824783}@huawei.com

Chao Qian[†]
National Key Laboratory for Novel Software Technology, Nanjing University
School of Artificial Intelligence, Nanjing University
Nanjing, China
qianc@lamda.nju.edu.cn

ABSTRACT

The pursuit of universal black-box optimization (BBO) algorithms is a longstanding goal. However, unlike domains such as language or vision, where scaling structured data has driven generalization, progress in offline BBO remains hindered by the lack of unified representations for heterogeneous numerical spaces. Thus, existing offline BBO approaches are constrained to single-task and fixed-dimensional settings, failing to achieve cross-domain universal optimization. Recent advances in language models (LMs) offer a promising path forward: their embeddings capture latent relationships in a unifying way, enabling universal optimization across different data types possible. In this paper, we discuss multiple potential approaches, including an end-to-end learning framework in the form of next-token prediction, as well as prioritizing the learning of latent spaces with strong representational capabilities. To validate the effectiveness of these methods, we collect offline BBO tasks and data from open-source academic works for training. Experiments demonstrate the universality and effectiveness of our proposed methods. Our findings suggest that unifying language model priors and learning string embedding space can overcome traditional barriers in universal BBO, paving the way for general-purpose BBO algorithms.

1 INTRODUCTION

The pursuit of universal black-box optimization (BBO) algorithms, i.e., capable of adapting to diverse problems, has been a long-standing challenge (Wolpert & Macready, 1997; Chen et al., 2022b; Lehre & Lin, 2024). Traditional BBO methods, including Bayesian optimization (Garnett, 2023) and evolutionary algorithms (Bäck, 1996; Zhou et al., 2019), excel in many tasks (Turner et al., 2021). However, these traditional BBO methods are not only confined to single-type and fixed-dimensional settings (Chen et al., 2022b; Song et al., 2024a; Fan et al., 2024) but also struggle to leverage large-scale offline data (Song et al., 2024d), heavily relying on online evaluations, which is a severe limitation that becomes particularly challenging in many real-world expensive scenarios.

[*]Equal Contribution
[†]Co-correspondence to Ke Xue and Chao Qian

Offline black-box optimization (Trabucco et al., 2021; 2022; Kim et al., 2025; Xue et al., 2024) aims to identify optimal designs for an unknown objective function using only a fixed and pre-collected dataset, which attempts to mitigate the issue mentioned above. Existing offline BBO approaches have shown impressive performance, such as real-world engineering design (Kumar et al., 2022; Shi et al., 2023), protein design (Chen et al., 2023b; Kim et al., 2023), and molecule design (Gaulton et al., 2012; Stanton et al., 2022; Xue et al., 2024). Their success relies on two assumptions: (1) the training and test tasks share identical input dimensions and variable types, and (2) sufficient historical data exists for each task.

While real-world optimization problems often exhibit inherent correlations across domains (Bai et al., 2023; Wang et al., 2024), existing offline BBO methods face two challenges: (1) insufficient training data for model development (Nguyen et al., 2023; Dara et al., 2022), and (2) fundamental inability to exploit cross-task relationships. These limitations create an urgent need for universal offline BBO. However, the critical barrier lies in the heterogeneity of search spaces. This limitation forces practitioners to collect large datasets for every new problem, due to their inability to unify parametric representations across domains, which is unsustainable in practical scenarios characterized by sparse data availability and diverse task requirements.

Fortunately, recent advances have demonstrated the feasibility of universal optimization in string-based search spaces using language models (LMs; Song et al., 2024b;c; Nguyen et al., 2024; Tang et al., 2025; Song & Bahri, 2025). By tokenizing numerical parameters into string sequences, LM-based optimizers outperform traditional regression models in cross-task generalization, particularly when trained on multi-task datasets spanning diverse domains. However, there are still several critical gaps in universal offline BBO. First, prior work does not discuss distinct paradigms in detail, failing to clarify their relative strengths or compatibility. Second, these works neglect the unique requirements of offline BBO, where algorithms must avoid overfitting to limited historical data (Trabucco et al., 2021; Fu & Levine, 2021; Yu et al., 2021; Chen et al., 2022a; Qi et al., 2022; Dao et al., 2024b; Tan et al., 2025). Besides, the geometric properties of learned latent spaces remain under-explored, despite their direct impact on optimization stability and sample efficiency.

In this paper, we propose a universal string-based offline BBO framework, UniSO. We first introduce several components for UniSO, including string-based data representation, and metadata formulation. Then, we use two model architectures for universal offline BBO, including token-targeted regressor (Song et al., 2024b) and numeric-targeted regressor (Nguyen et al., 2024), formulating two Vanilla UniSO variants, i.e., UniSO-T and UniSO-N, respectively. However, several issues exist when directly applying these Vanilla UniSO variants to offline BBO tasks, as shown in Figure 3. To address these issues, we propose two improvements, i.e., embedding distribution alignment via metadata guidance and local embedding smoothness enhancement. Experiments demonstrate the universality and effectiveness of our proposed methods, particularly showing that UniSO-T achieves superior cross-task generalization through multi-task training on heterogeneous search spaces.

Our findings reveal three key insights: (1) a unified representation through string-based space enables universal offline BBO possible, (2) geometric regularization of embedding spaces significantly enhances optimization stability, and (3) metadata-guided learning effectively bridges the domain gap between training and unseen test tasks. These advancements collectively overcome traditional barriers in universal offline BBO, paving the way for general-purpose optimization across various types and dimensions.

2 PRELIMINARIES

In this section, we first introduce the formal definition of offline BBO as well as several typical approaches. Then, we extend to the setting of universal offline BBO.

2.1 OFFLINE BBO

Given the design space $\mathcal{X}$, where $\mathcal{X}$ could be be `DOUBLE`, `INTEGER`, `CATEGORICAL`, or `PERMUTATION`, and a fixed offline dataset $\mathcal{D}$, offline BBO (Trabucco et al., 2022; Kim et al., 2025; Xue et al., 2024) aims to seek an optimal design $\mathbf{x}^*$ that maximizes a black-box objective function $f : \mathcal{X} \to \mathbb{R}$, i.e., $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$. This optimization process relies solely on the available dataset, with no online evaluations permitted during optimization. Specifically, an algorithm operates exclusively on a fixed dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where each instance $\mathbf{x}_i$ represents a design

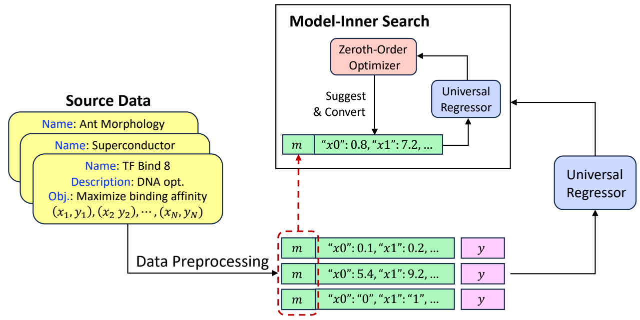

Figure 1: Framework of universal string-based offline BBO.

(e.g., the composition of a DNA sequence), and its corresponding scalar value $y_i = f(\mathbf{x}_i)$ denotes the objective score (e.g., a specific property score of the designed DNA sequence). Besides, for an offline BBO task, there usually exists task-level metadata m, which can distinguish it from other tasks and potentially hint the information of the unknown objective function f. Thus, similar to the multi-task regression case (Song et al., 2024b), an offline BBO task can be formulated as $\mathcal{T} = (\mathcal{X}, f, \mathcal{D}, m)$.

A prevalent approach for offline BBO is the *forward* approach, which first trains a surrogate model, typically a deep neural network $\hat{f}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}$ parameterized by $\boldsymbol{\theta}$, to learn a scoring function for the designs where the output corresponds to the predicting score, and then searches for the final design by maximizing the model's output. The scoring function can be learned by regression (Trabucco et al., 2021; Chen et al., 2023a) or ranking (Tan et al., 2025). However, this type of approach suffers from its error in out-of-distribution (OOD) regions, which might mislead the model-internal search (Yao et al., 2024). We deliver a detailed related work, including both the mentioned *forward* and *backward* approaches, as well as LLM for BBO in Appendix A.

2.2 Universal Offline BBO

Recent advances in foundational models for black-box optimization research (Song et al., 2024b;c; Nguyen et al., 2024; Tang et al., 2025; Song & Bahri, 2025) have shown the potential of universal optimization in string-based space. Thus, we first extend conventional offline BBO to the universal setting. The goal of universal offline BBO is to **simultaneously** address multiple tasks through a universal foundational model. Consider $n_{\mathcal{T}}$ optimization tasks $\{\mathcal{T}_i\}_{i=1}^{n_{\mathcal{T}}}$ with three key characteristics: (1) **Heterogeneous design spaces**: Mixed-type parameters and varying dimensionality across tasks; (2) **Divergent objectives**: Unique optimization functions and dataset scales for each task; and (3) **Task-specific metadata**: Distinct auxiliary information accompanying each task.

While more challenging than single-task offline BBO, the universal approach offers two key advantages. First, it enables knowledge transfer between related tasks (e.g., morphology optimization in Ant (Brockman et al., 2016) and D'Kitty (Ahn et al., 2020)), overcoming the isolation assumption of conventional methods that process tasks independently. Second, it addresses the data scarcity challenge common in real-world applications (Nguyen et al., 2023), where traditional methods fail with limited historical data, the universal offline BBO model can leverage cross-task patterns to guide optimization.

3 Method

In this section, we introduce potential methods to solve universal offline BBO. We begin by outlining the general framework of universal string-based offline BBO in Section 3.1, where we will present two variants based on modeling. Then, we discuss the issues of these variants in Section 3.2, which motivate us to improve the framework via *metadata guidance* and *smoothness enhancement* in Section 3.3 and Section 3.4, respectively. In Section 3.5, we discuss how to balance multiple losses and apply the improvement to the variants.

3.1 UniSO: Universal String-based Offline BBO

In this subsection, we present the general framework for universal offline BBO, which is illustrated in Fig. 1. The framework comprises four main components. Firstly, we convert the design-score offline data into *string representations* due to the heterogeneity of different search spaces from different tasks. Next, we discuss the importance and formulation for metadata, which facilitates the subsequent universal multi-task regressor training intended for downstream optimization. We propose two modeling variants for universal regressor instantiation based on different ways of handling objective scores, which are derived from recent advancements of string-based LLM for BBO (Song et al., 2024b; Nguyen et al., 2024; Tang et al., 2025). Finally, we elaborate on our string-based search strategy within the model. This pipeline aligns with the forward approach in traditional offline BBO, which typically encompasses training a scoring model followed by model-inner optimization, as discussed in Section 2.1.

3.1.1 STRING-BASED DATA REPRESENTATION

To solve multiple offline BBO problems with heterogeneous design spaces simultaneously, traditional methods that restrict the algorithm inside a fixed search scope are inapplicable, which calls for a more flexible representation for design-score pair data. Recently, Song et al. (2024b); Nguyen et al. (2024) show that string representation over $\mathbf{x}$ using LLM is efficient and beneficial for BBO, enabling optimization in dynamic design spaces. Following Nguyen et al. (2024), we represent each design $\mathbf{x}$ by a JSON dictionary-like format, e.g., a design $\mathbf{x} = (0, 1)^\top$ can be represented as `{"x0":0,"x1":1}`.

As for the score y, details of handling y encompass token-based method and numeric-based method, which are determined by modeling methods of the regressor, which we will introduce in Section 3.1.3.

3.1.2 FORMULATION OF METADATA

While the mentioned string-based data representation exhibits flexibility, it alone is insufficient for effectively distinguishing between tasks and performing optimization in a multi-task suite. The integration of metadata into algorithms is quite essential for universal offline BBO, due to the following reasons: (1) utilizing a single algorithm to solve multiple offline BBO problems simultaneously solely on general design-score data is impossible due to the no free lunch theorem (Wolpert & Macready, 1997; Lehre & Lin, 2024), and expert priors (Hvarfner et al., 2022; 2024) or user-defined task-specific metadata should be considered for optimization (Song et al., 2024c; Lindauer et al., 2024); (2) metadata enables training acceleration (Gao et al., 2025) and is crucial for task distinguishment when expert priors are limited (Chen et al., 2022b).

Thus, we extend each string $\mathbf{x}$ by inserting its associated metadata m in the beginning. In order to maintain concision and preserve key information for optimization, we formulate the expressions of metadata m, consisting of three text-based task specifications: (1) **task name** to distinguish from other tasks; (2) **brief description** of the task with a concise natural language summary; and (3) detailed specification of the **optimization objective**. We provide all the metadata we used in our experiments in Appendix E. We tokenize the $(m, \mathbf{x})$ data using SentencePiece tokenizer (Kudo & Richardson, 2018) by default.

3.1.3 MULTI-TASK REGRESSOR TRAINING FOR UNIVERSAL OFFLINE BBO

The scoring model for offline BBO can be modeled by regression or ranking model. Here, we initiate the scoring model by a universal multi-task regressor for simplification. As shown in Fig. 2, we consider two variants of universal end-to-end regressors, UniSO-T and UniSO-N, which differ from how to deal with the objective score y.

UniSO-T: Token-targeted regressor. UniSO-T is a regressor with the objective of predicting the y tokens. Specifically, it encodes the score y into a list of tokens, trains a sequence-to-sequence auto-regressive model to predict the objective sequence, and then maps the sequence back to numerical values. A typical modeling of UniSO-T is OmniPred (Song et al., 2024b) , which tokenizes y by digits using P10 encoding[1] from Charton (2022) and trains a universal regressor that based on next token prediction. Inspired by OmniPred, we tokenize y using P10 encoding and implement a lightweight variant of the T5 model (Raffel et al., 2020). The model architecture incorporates Prefix-LM training (Liu et al., 2018) and is trained using cross-entropy loss on our custom dataset, as the training data of OmniPred follows a distinct organizational structure and remains unavailable.

[1] For example, $y = 1.31 = 131 \times 10^{-2}$ is encoded into `<+><1><3><1><E-2>`

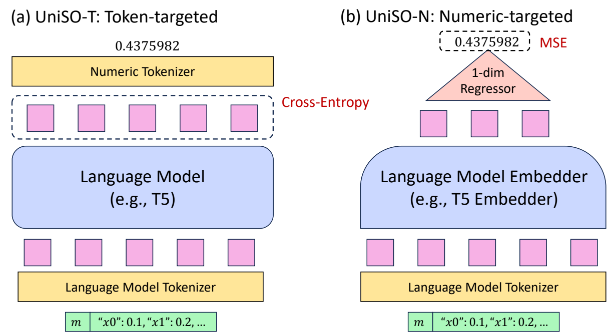

Figure 2: Model structure of UniSO-T (left) and UniSO-N (right).

UniSO-N: Numeric-targeted regressor. UniSO-N is another type of universal multi-task regressor, which first embeds the inputs into a unified latent space, and then trains a downstream regressor to map from embeddings to numerical objective scores. Inspired by Nguyen et al. (2024), which uses a pre-trained T5 encoder (Raffel et al., 2020) and trains an in-context regressor for Bayesian optimization, we also employ a pre-trained `t5-small` embedder[2], and train a MLP regressor mapping from embeddings to y with mean squared error (MSE). We also normalize y from different tasks using the same strategy in Nguyen et al. (2024), which incorporates three steps: 1) utilize z-score normalization the shift the score of a single task; 2) implement a normal distribution fitting procedure on the subset of observations falling below the median value, utilizing a percentile-to-z-score transformation, to reduce sensitivity to bad outliers; 3) transform the scoring distribution through sequential application of min-max scaling $y \leftarrow \frac{y-y_{\min}}{y_{\max}-y_{\min}}$ and logarithmic transformation for re-scaling. Besides, although regressor training in conventional offline BBO utilizes global z-score normalization, global statistics like mean value or standard variation of the embeddings are inaccessible, thus batch normalization is applied to the embeddings before objective regression.

3.1.4 String-Based Model-Inner Search

Upon completion of model training, the final designs are obtained through output maximization via model-inner search. Given the inherent challenges of computing exact gradients in discrete string space compared to continuous numerical space, we adopt black-box optimization strategies for this search process. Our implementation primarily employs evolutionary algorithms (EAs; Bäck, 1996), which have demonstrated superior performance in gradient-free optimization scenarios. The technical specifications of our EA implementation, including variation operators and selection mechanisms can be found in our code implementation.

3.2 Issues of the Vanilla UniSO

Although the typical methods for UniSO-T and UniSO-N have demonstrated success in online BBO (Nguyen et al., 2024), there still exists limitations for offline BBO. Firstly, both the regressors in UniSO-T and UniSO-N cannot effectively distinguish different tasks without billions of data. To understand this, we visualize the t-SNE (der Maaten & Hinton, 2008) plots of the embeddings of vanilla UniSO-T in the left figure of Fig 3, where the latent embeddings of vanilla UniSO-T show overlapping and circular pattern, lacking clear boundaries to distinguish tasks.

Besides, the embedding latent space for both UniSO-T and UniSO-N are not particularly designed for BBO, while properties of the latent spaces, like smoothness (Lee et al., 2023) or invariance (Qi et al., 2022), are usually important for downstream optimization.

Aiming at addressing these limitations, we introduce our proposed improvements that can be applied to both the modelings in the following subsections, which regulate the embedding space based on two different perspectives, ***metadata guidance*** and ***smoothness enhancement***.

[2] https://huggingface.co/google-t5/t5-small

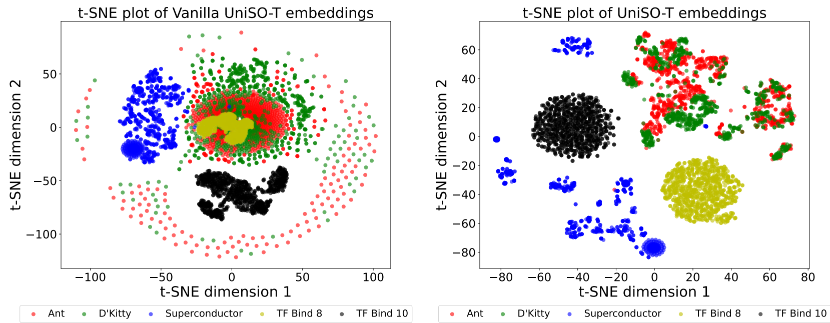

Figure 3: t-SNE plots comparing embedding distributions of vanilla UniSO-T and improved UniSO-T on five Design-Bench (Trabucco et al., 2022) tasks. The embedding distributions of vanilla UniSO-T show mixed and overlapping embeddings with a circular pattern, lacking clear task boundaries. Our improved method (right) achieves three improvements: (1) separating embeddings into distinct clusters for better task discrimination, (2) maintaining proximity between similar tasks (e.g., Ant and D'Kitty) to enable knowledge sharing between related domains, and (3) generating compact and smooth intra-task embedding distributions for stable representations.

3.3 Embedding Distribution Alignment via Metadata Guidance

As discussed above, the embeddings exhibit strange shape and cannot distinguish different tasks. Thus, we adopt a novel approach to align the distribution of input embeddings with their corresponding metadata embeddings via contrastive learning. The core idea is to guide the learning process by encouraging input embeddings to exhibit similarity patterns that mirror those observed in the metadata space.

Specifically, we first encode the input strings through the encoder to derive their embeddings, while simultaneously employing a pre-trained expert language model embedder (`t5-small` by default) to generate embeddings for metadata. After implementing mean pooling on both embeddings, we transform these intermediate representations through their respective nonlinear projection heads (initiated by two linear layers with ReLU activation) into a shared embedding space, yielding the final latent representations $\mathbf{z}^x \in \mathbb{R}^{n\times d}$ and $\mathbf{z}^m \in \mathbb{R}^{n\times d}$, where n denotes the batch size and d represents the dimension of the shared embedding space. Such a nonlinear projection is widely used in contrastive learning (Chen et al., 2020).

These projected representations are then utilized to compute the contrastive loss to align the distribution of input embeddings with their corresponding metadata embeddings:

$$\mathcal{L}_{\text{con}} = -\frac{1}{N(N-1)} \sum_{1\le i<j\le N} \hat{s}^m_{ij} \log\left(\frac{\exp(s^x_{ij}/\tau)}{\sum_{k\neq i}\exp(s^x_{ik}/\tau)}\right),$$

where: (1) $s^x_{ij} = \frac{\mathbf{z}_i^{x\top}\mathbf{z}_j^x}{\|\mathbf{z}_i^x\|\cdot\|\mathbf{z}_j^x\|}$ represents the cosine similarity between input embeddings, and similarly, s^m_{ij} is defined as the cosine similarity of the metadata embedding; (2) $\hat{s}^m_{ij} = \frac{s^m_{ij}-\min_{i,j}(s^m_{ij})}{\max_{i,j}(s^m_{ij})-\min_{i,j}(s^m_{ij})}$ is the normalized metadata similarity; (3) τ is the temperature parameter; (4) N is the data size.

The contrastive loss function enforces a dual objective by minimizing the KL divergence between the input embedding similarity distribution and the metadata-derived target distribution. An illustrative understanding of this loss function is that embeddings of the inputs with similar metadata remain proximate while dissimilar ones maintain distinct boundaries. As shown in the right figure in Fig. 3, compared to the vanilla UniSO-T case, our proposed improvement is capable of distinguishing dissimilar tasks with the clearer clusters, while similar tasks like Ant and D'Kitty remain close, enabling structured information sharing for similar tasks.

However, this contrastive loss would raise non-smoothness[3] inside a task. The input embeddings from a same task (i.e., with same metadata) can occupy arbitrary positions within a localized region of the

[3]In this work, smoothness refers to continuity in the embedding space.

high-dimensional space, forming either distinct clusters or non-uniform distributions, as long as they maintain adequate contrast (i.e., cosine similarity in the loss function) with respect to embeddings with dissimilar metadata. In Section 3.4, we further enhance the local smoothness of embeddings from a single task.

3.4 Local Embedding Smoothness Enhancement

Smoothness is crucial for neural network generalization (Nakkiran et al., 2019) and robustness (Weng et al., 2018), and smoothness of latent representation is also important for BBO (Zhang et al., 2020; Lee et al., 2023). For a given task $\mathcal{T}$, to enhance the local smoothness of embedding from $\mathcal{T}$, inspired by Lee et al. (2023), we regulate the embedding space via the Lipschitz loss:

$$\mathcal{L}_{\text{lip}_{\mathcal{T}}} = \sum_{1 \le i < j \le N_{\mathcal{T}}} \max\left(0, \frac{|y_i - y_j|}{\|\mathbf{z}_i - \mathbf{z}_j\|_2} - L\right),$$

where $N_{\mathcal{T}}$ is the dataset size of task $\mathcal{T}$ and L represents the local Lipschitz constant. We set L as the median value of the Lipschitz matrix, i.e., $L = \text{median}_{1 \le i < j \le n_{\mathcal{T}}}(\frac{y_i - y_j}{\|\mathbf{z}_i - \mathbf{z}_j\|_2})$, by default following Lee et al. (2023). The Lipschitz loss increases the correlation between the Euclidean distance of latent embeddings and the differences in their corresponding objective scores, enforcing the local embeddings to remain a local smoothness similar to the corresponding scores. Then, we compute the weighted sum of Lipschitz losses across all tasks, where the weights are designed to balance the impact of varying dataset sizes of different tasks:

$$\mathcal{L}_{\text{lip}} = \sum_{i=1}^{n_{\mathcal{T}}} \frac{\sum_{j=1}^{n_{\mathcal{T}}} N_{\mathcal{T}_j}}{N_{\mathcal{T}_i}} \mathcal{L}_{\text{lip}_{\mathcal{T}_i}},$$

where $n_{\mathcal{T}}$ denotes the number of tasks, and $\frac{\sum_{j=1}^{n_{\mathcal{T}}} N_{\mathcal{T}_j}}{N_{\mathcal{T}_i}}$ represents the weighting coefficient for task $\mathcal{T}_i$ to compensate for the disparity in dataset sizes across tasks. In our implementation, due to insufficient task-specific samples in individual batches of the shuffled training dataset, we utilize an un-shuffled dataset that presents tasks sequentially for computing the Lipschitz loss.

3.5 Loss Balancing and Application to UniSO

With the main loss $\mathcal{L}_{\text{main}}$ to train the universal regressor (i.e., cross-entropy for UniSO-T) and the mentioned regularization losses in hand, a naïve approach to handle these losses is to sum them up directly. However, simply summing them up would be effected by the scales of the losses, which focuses mainly on the loss with the largest scale during gradient backpropagation (Chen et al., 2018). Thus, a crucial problem is *how to balance the effects of these losses for training?*

To solve this, inspired by MetaBalance (He et al., 2022) which normalizes the gradients of different losses based on the gradient norm, we employ a similar yet simple technique to automatically balance the gradient of different losses based on the loss values. Specifically, for the main loss $\mathcal{L}_{\text{main}}$ and regularization losses $\mathcal{L}_{\text{con}}, \mathcal{L}_{\text{lip}}$, we calculate the gradient as:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}_{\text{main}}}{\partial \boldsymbol{\theta}} + \frac{\mathcal{L}_{\text{main}}}{\mathcal{L}_{\text{con}} + \delta} \cdot \frac{\partial \mathcal{L}_{\text{con}}}{\partial \boldsymbol{\theta}} + \frac{\mathcal{L}_{\text{main}}}{\mathcal{L}_{\text{lip}} + \delta} \cdot \frac{\partial \mathcal{L}_{\text{lip}}}{\partial \boldsymbol{\theta}}, \tag{1}$$

where $\boldsymbol{\theta}$ represents the parameters of the model and $\delta = 1 \times 10^{-10}$ is a small constant added for numerical stability. This loss propagation mechanism adjusts the relative contributions of auxiliary losses by scaling them with respect to the main loss, maintaining task balance while preserving the principal optimization direction.

The balanced loss function is incorporated into both modeling frameworks as follows. For UniSO-T modeling, since an end-to-end universal regressor is trained from scratch, we update the parameters $\boldsymbol{\theta}$ with respect to Eq.1 during training where we view the last hidden state of the encoder model as embeddings. For UniSO-N modeling, since the vanilla approach employs a pre-trained embedder without parameter updates during downstream numerical regressor training, we propose a two-stage approach: first fine-tuning the embedder via $\frac{\partial \mathcal{L}_{\text{total}}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}_{\text{con}}}{\partial \boldsymbol{\theta}} + \frac{\mathcal{L}_{\text{con}}}{\mathcal{L}_{\text{lip}} + \delta} \cdot \frac{\partial \mathcal{L}_{\text{lip}}}{\partial \boldsymbol{\theta}}$ similar to Eq. 1, and then training the regressor while keeping the embedder parameters frozen.

Table 1: Score in unconstrained tasks from Design-Bench and SOO-Bench, where the best and runner-up results on each task are **Blue** and **Violet**, respectively. $\mathcal{D}$(best) denotes the best score in the offline dataset and BN represents batch normalization for numerical input designs. Both numeric-input experts and string-input UniSO methods are trained **within a single task**.

Task	$\mathcal{D}$(best)	Numeric-input Experts		String-input UniSO	
		BN + EAs	BN + Grad	UniSO-T	UniSO-N
Ant	165.326	118.877 ± 127.688	229.462 ± 165.869	**245.212 ± 165.083**	**254.720 ± 102.151**
D'Kitty	199.363	111.205 ± 66.986	183.263 ± 62.436	**229.114 ± 25.766**	**188.642 ± 26.306**
Superconductor	74.000	**93.951 ± 7.039**	**97.137 ± 6.113**	88.720 ± 9.884	63.930 ± 6.748
TF Bind 8	0.439	**0.984 ± 0.007**	**0.959 ± 0.023**	0.948 ± 0.028	0.949 ± 0.000
TF Bind 10	0.005	**0.905 ± 0.326**	**0.888 ± 0.229**	0.623 ± 0.115	0.600 ± 0.006
GTOPX 2	-195.586	**-88.054 ± 20.878**	-128.310 ± 15.616	**-76.479 ± 4.815**	-117.022 ± 51.671
GTOPX 3	-151.190	**-64.028 ± 22.678**	-151.190 ± 0.000	**-46.526 ± 13.434**	-71.784 ± 16.665
GTOPX 4	-215.716	**-96.432 ± 10.868**	-215.716 ± 0.000	**-87.714 ± 8.795**	-101.304 ± 18.492
GTOPX 6	-112.599	**-64.217 ± 14.602**	-112.599 ± 0.000	**-48.186 ± 9.486**	-80.391 ± 12.469
Avg. Rank	/	**2.222 ± 1.030**	3.000 ± 1.054	**1.889 ± 1.100**	2.889 ± 0.875

4 Experiment

In this section, we empirically study our proposed framework of universal string-based offline BBO on various tasks. We first introduce the experimental settings and the tasks we used in Section 4.1, then we show the performance of UniSO in Section 4.2 and answer several important research questions (RQs).

4.1 Experimental Settings and Tasks

Details settings. Following prior works in string-based LLM for BBO (Song et al., 2024b; Nguyen et al., 2024), we adopt a lightweight T5 (Raffel et al., 2020) for both UniSO variants. For UniSO-N, the downstream regressor consists of a two-layer MLP with 2048 hidden units, trained using AdamW optimizer (Loshchilov & Hutter, 2019) for 200 epochs with a batch size of 128. After training, we employ EAs to search within the model for 200 iterations, maintaining a population size of 128 in accordance with standard offline BBO practices (Trabucco et al., 2022).

Selected tasks. We evaluate on unconstrained tasks from Design-Bench (Trabucco et al., 2022) and SOO-Bench (Qian et al., 2025). From Design-Bench[4], we select Ant Morphology (Brockman et al., 2016), D'Kitty Morphology (Ahn et al., 2020), Superconductor (Hamidieh, 2018), TF Bind 8 and TF Bind 10 (Barrera et al., 2016). From SOO-Bench, we include GTOPX 2,3,4,6 (Schlueter et al., 2021). These tasks span both `DOUBLE` and `CATEGORICAL` search spaces with varying dimensionality. Detailed task descriptions and properties are provided in Appendix C.

4.2 Experimental Results

In this section, we present our experimental findings, aiming to answer the following RQs.

RQ1: Is the string-based representation for designs versatile to offline BBO in the single task cases? While prior research has demonstrated the effectiveness of string-based representations in universal regression (Song et al., 2024b; Tang et al., 2025) and online BBO algorithms (Nguyen et al., 2024), their applicability to offline BBO remains an open question. To bridge this gap, we first conduct a comparative study between string-based UniSO variants and conventional numeric-input expert models in single-task offline BBO settings. For fair comparison, we implement expert models with identical architecture to UniSO-N's objective regressor, with numerical inputs undergoing batch normalization followed by model-inner optimization through both EAs and gradient ascent.

As shown in Table 1, the UniSO methods consistently exceed the best score in the offline dataset $\mathcal{D}$(best) across most tasks, with UniSO-T demonstrating superior performance by achieving an average rank of 1.889 and outperforming other methods in 5 out of 10 tasks, while the strongest baseline, BN + EAs, achieves an average rank of 2.222. Although UniSO-N shows lower performance compared to BN + EAs, it maintains superiority over BN + Grad and consistently exceeds $\mathcal{D}$(best). These results provide evidence for the versatility of string-based representations in offline BBO.

[4]Following recent works in offline BBO (Tan et al., 2025; Yun et al., 2024), we exclude three tasks from Design-Bench, and provide detailed explanation in Appendix C.1

Table 2: Score in unconstrained tasks from Design-Bench and SOO-Bench, where the best and runner-up results on each task are **Blue** and **Violet**, respectively. $\mathcal{D}$(best) denotes the best score in the offline dataset and BN represents batch normalization for numerical input designs. Single-task experts are trained within one task, while UniSO-T and UniSO-N are done in a multi-task manner.

Task	$\mathcal{D}$(best)	Single-task Experts		UniSO-T		UniSO-N	
		BN + EAs	BN + Grad	Vanilla	Improved	Vanilla	Improved
Ant	165.326	118.877 ± 127.688	229.462 ± 165.869	**275.216 ± 90.820**	**374.665 ± 56.057**	268.399 ± 82.858	269.691 ± 77.425
D'Kitty	199.363	111.205 ± 66.986	183.263 ± 62.436	**216.070 ± 23.209**	**225.752 ± 8.521**	130.655 ± 83.106	173.911 ± 46.662
Superconductor	74.000	**93.951 ± 7.039**	**97.137 ± 6.113**	86.795 ± 13.466	92.200 ± 15.209	81.266 ± 16.073	67.333 ± 16.838
TF Bind 8	0.439	**0.984 ± 0.007**	**0.959 ± 0.023**	0.940 ± 0.027	0.903 ± 0.041	0.944 ± 0.016	0.833 ± 0.005
TF Bind 10	0.005	**0.905 ± 0.326**	0.888 ± 0.229	0.830 ± 0.539	0.823 ± 0.542	0.603 ± 0.005	**0.959 ± 0.115**
GTOPX 2	-195.586	**-88.054 ± 20.878**	-128.310 ± 15.616	-132.023 ± 63.084	**-72.848 ± 9.576**	-117.022 ± 51.671	-124.995 ± 56.170
GTOPX 3	-151.190	-64.028 ± 22.678	-151.190 ± 0.000	**-60.941 ± 17.235**	**-45.602 ± 8.433**	-88.601 ± 31.865	-62.622 ± 22.261
GTOPX 4	-215.716	**-96.432 ± 10.868**	-215.716 ± 0.000	-100.943 ± 15.044	**-84.271 ± 8.307**	-99.834 ± 20.837	-110.284 ± 17.559
GTOPX 6	-112.599	-64.217 ± 14.602	-112.599 ± 0.000	-71.749 ± 28.497	**-47.794 ± 11.943**	-71.174 ± 12.932	**-57.435 ± 18.832**
Avg. Rank	/	**3.111 ± 1.728**	4.111 ± 1.792	3.667 ± 1.333	**2.111 ± 1.663**	4.222 ± 1.030	3.778 ± 1.618

RQ2: How do the universal string-based offline BBO methods perform across a wide range of offline BBO tasks? In Table 2, we report the results of our main experiments, which demonstrates the effectiveness of universal string-based offline BBO methods across diverse tasks. We compare UniSO-T and UniSO-N against batch-normalized $\mathcal{D}$(best) and single-task experts. Our findings reveal that: (1) both vanilla and improved versions of UniSO methods generally outperform $\mathcal{D}$(best) across tasks, with only three exceptions in total; (2) UniSO methods achieve comparable or superior performance to single-task experts, with improved UniSO-T showing particularly strong results with an average rank of 2.111 among six methods; (3) our latent space regularization techniques yield consistent improvements in both performance and average rank for both UniSO variants. The t-SNE visualization of improved UniSO-T (Fig. 3, right) reveals enhanced task separation while maintaining proximity between related tasks, facilitating effective information sharing, along with more coherent within-task embedding distributions in the latent space.

RQ3: Can UniSO generalize to unseen tasks? We evaluate the generalization capability of improved UniSO-T through zero-shot and few-shot experiments on three unseen tasks: RobotPush, Rover (Wang et al., 2018), and LunarLander (Brockman et al., 2016). This evaluation is motivated by the strong generalization performance demonstrated by string-based universal online BBO methods (Nguyen et al., 2024). For zero-shot inference, we directly optimize the model's output by concatenating metadata with converted string designs. In the few-shot setting, we fine-tune UniSO-T on 100 lowest-performing data provided by Wang et al. (2024) using cross-entropy loss and SGD optimizer at a learning rate of 2×10^{-5} for 5 epochs before design optimization. The results in Table 3 demonstrate the model's strong generalization ability: zero-shot performance surpasses $\mathcal{D}$(best), while few-shot results outperform the z-score normalized single-task expert.

Table 3: Zero-shot and few-shot results of improved UniSO-T on RobotPush, Rover, and LunarLander, compared to single-task experts based on z-score normalization.

Task	$\mathcal{D}$(best)	Single-Task Experts		Improved UniSO-T	
		Z-score + EAs	Z-score + Grad	Zero-shot	Few-shot
RobotPush	0.102	**2.423 ± 1.181**	5.816 ± 1.132	3.171 ± 0.984	**7.067 ± 0.169**
Rover	-16.148	-14.651 ± 4.692	-11.578 ± 0.351	**-8.888 ± 2.119**	**-8.239 ± 1.270**
LunarLander	7.038	38.164 ± 87.671	**271.838 ± 52.982**	31.186 ± 27.971	**248.573 ± 45.386**
Avg. Rank	/	4.000 ± 0.000	**2.000 ± 0.816**	2.667 ± 0.471	**1.333 ± 0.471**

In Appendix D, we show the effectiveness of each component of the proposed losses and metadata processing on UniSO-T by conducting ablation studies. Besides, we compare the best-performing UniSO variant, improved UniSO-T, to the well-studied expert single-task offline BBO methods, where improved UniSO-T obtains an average rank of 9.8 across 21 expert single-task offline BBO methods, demonstrating the potential improvement for universal offline BBO.

5 Conclusion

In this paper, we propose UniSO to overcome barriers in universal offline BBO by unifying string-based representation, latent space regularization, and metadata-guided learning, showing universality and effectiveness. Future work includes training on more data and exploring in-context learning for enhanced performance in diverse real-world scenarios.

Acknowledgments

This work was supported by the National Science and Technology Major Project (2022ZD0116600), the National Science Foundation of China (62276124, 624B1025, 624B2069), the Fundamental Research Funds for the Central Universities (14380020), and Young Elite Scientists Sponsorship Program by CAST for PhD Students. The authors want to acknowledge support from the Huawei Technology cooperation Project.

References

Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. ROBEL: Robotics benchmarks for learning with low-cost robots. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, pp. 1300–1313, Virtual, 2020.

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

Tianyi Bai, Yang Li, Yu Shen, Xinyi Zhang, Wentao Zhang, and Bin Cui. Transfer learning for Bayesian optimization: A survey. *arXiv:2302.05927*, 2023.

Luis A. Barrera, Anastasia Vedenko, Jesse V. Kurland, Julia M. Rogers, Stephen S. Gisselbrecht, Elizabeth J. Rossin, Jaie C. Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, Trevor Siggers, Leila Shokri, Raluca Gordân, Nidhi Sahni, Chris Cotsapas, Tong Hao, S. Stephen Yi, Manolis Kellis, Mark J. Daly, Marc Vidal, David E. Hill, and Martha L. Bulyk. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science*, 351(6280):1450–1454, 2016.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv:1606.01540*, 2016.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 1877–1901, Vancouver, Canada, 2020.

Francois Charton. Linear algebra with Transformers. *Transactions on Machine Learning Research*, 2022.

Yassine Chemingui, Aryan Deshwal, Trong Nghia Hoang, and Janardhan Rao Doppa. Offline model-based optimization via policy-guided gradient search. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 11230–11239, Vancouver, Canada, 2024.

Can Chen, Christopher Beckham, Zixuan Liu, Xue Liu, and Christopher Pal. Robust guided diffusion for offline black-box optimization. *Transactions on Machine Learning Research*, 2024.

Can (Sam) Chen, Yingxue Zhang, Jie Fu, Xue (Steve) Liu, and Mark Coates. Bidirectional learning for offline infinite-width model-based optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 29454–29467, New Orleans, LA, 2022a.

Can (Sam) Chen, Christopher Beckham, Zixuan Liu, Xue (Steve) Liu, and Christopher Pal. Parallel-mentoring for offline model-based optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 76619–76636, New Orleans, LA, 2023a.

Can (Sam) Chen, Yingxue Zhang, Xue (Steve) Liu, and Mark Coates. Bidirectional learning for offline model-based biological sequence design. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 5351–5366, Honolulu, HI, 2023b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, Virtual, 2020.

Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Qiuyi Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'aurelio Ranzato, Sagi Perel, and Nando de Freitas. Towards learning universal hyperparameter optimizers with Transformers. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 32053–32068, New Orleans, LA, 2022b.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 794–803, Stockholm, Sweden, 2018.

Manh Cuong Dao, Phi Le Nguyen, Thao Nguyen Truong, and Trong Nghia Hoang. Incorporating surrogate gradient norm to improve offline optimization techniques. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, Vancouver, Canada, 2024a.

Manh Cuong Dao, Phi Le Nguyen, Thao Nguyen Truong, and Trong Nghia Hoang. Boosting offline optimizers with surrogate sensitivity. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 10072–10090, Vienna, Austria, 2024b.

Suresh Dara, Swetha Dhamercherla, Surender Singh Jadav, Ch Madhu Babu, and Mohamed Jawed Ahsan. Machine learning in drug discovery: A review. *Artificial Intelligence Review*, 55(3): 1947–1999, 2022.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Lucio M. Dery, Abram L. Friesen, Nando De Freitas, Marc'Aurelio Ranzato, and Yutian Chen. Multi-step planning for automated hyperparameter optimization with OptFormer. In *Foundation Models for Decision Making Workshop at NeurIPS'22*, New Orleans, LA, 2022.

Zhou Fan, Xinran Han, and Zi Wang. Transfer learning for Bayesian optimization on heterogeneous search spaces. *Transactions on Machine Learning Research*, 2024.

Clara Fannjiang and Jennifer Listgarten. Autofocused oracles for model-based design. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 12945–12956, Virtual, 2020.

Justin Fu and Sergey Levine. Offline model-based optimization via normalized maximum likelihood estimation. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual, 2021.

Tianyu Gao, Alexander Wettig, Luxi He, Yihe Dong, Sadhika Malladi, and Danqi Chen. Metadata conditioning accelerates language model pre-training. *arXiv:2501.01956*, 2025.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can Transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 30583–30598, New Orleans, LA, 2022.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

Anna Gaulton, Louisa J. Bellis, A. Patrícia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1): D1100–D1107, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, pp. 139–144, Montreal, Canada, 2014.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, G. Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, P. Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv:1812.05905*, 2018.

Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018.

Yun He, Xue Feng, Cheng Cheng, Geng Ji, Yunsong Guo, and James Caverlee. MetaBalance: Improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022 (WWW)*, pp. 2205–2215, Lyon, France, 2022.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 6840–6851, Virtual, 2020.

Minh Hoang, Azza Fadhel, Aryan Deshwal, Jana Doppa, and Trong Nghia Hoang. Learning surrogates for offline black-box optimization via gradient matching. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 18374–18393, Vienna, Austria, 2024.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Carl Hvarfner, Danny Stoll, Artur Souza, Luigi Nardi, Marius Lindauer, and Frank Hutter. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Virtual, 2022.

Carl Hvarfner, Frank Hutter, and Luigi Nardi. A general framework for user-guided Bayesian optimization. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024.

Dario Izzo. *Global Optimization and Space Pruning for Spacecraft Trajectory Design*, pp. 178–199. Cambridge University Press, 2010.

Dario Izzo and Manuel Manuel López-Ibáñez. Optimization challenges at the European Space Agency. In *Proceedings of the 24th ACM Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1542–1553, Boston, MA, 2022.

Caigao Jiang, Xiang Shu, Hong Qian, Xingyu Lu, Jun Zhou, Aimin Zhou, and Yang Yu. LLMOPT: Learning to define and solve general optimization problems from scratch. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, 2025.

Minsu Kim, Federico Berto, Sungsoo Ahn, and Jinkyoo Park. 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.

Minsu Kim, Jiayao Gu, Ye Yuan, Taeyoung Yun, Zixuan Liu, Yoshua Bengio, and Can Chen. Offline model-based optimization: Comprehensive review. *arXiv:2503.17286*, 2025.

Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron Ferber, Yi-An Ma, Carla P. Gomes, and Chao Zhang. Diffusion models as constrained samplers for optimization with unknown constraints. In *Generative Models for Decision Making workshop at ICLR'24*, Vienna, Austria, 2024.

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

Jakub Grudzien Kuba, Pieter Abbeel, and Sergey Levine. Cliqueformer: Model-based optimization with structured Transformers. *arXiv:2410.13106*, 2024a.

Jakub Grudzien Kuba, Masatoshi Uehara, Sergey Levine, and Pieter Abbeel. Functional graphical models: Structure enables offline data-driven optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2449–2457, Valencia, Spain, 2024b.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pp. 66–71, Brussels, Belgium, 2018.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 5126–5137, Virtual, 2020.

Aviral Kumar, Amir Yazdanbakhsh, Milad Hashemi, Kevin Swersky, and Sergey Levine. Data-driven offline optimization for architecting hardware accelerators. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Virtual, 2022.

Seunghun Lee, Jaewon Chu, Sihyeon Kim, Juyeon Ko, and Hyunwoo J. Kim. Advancing Bayesian optimization via learning correlated latent space. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 48906–48917, New Orleans, LA, 2023.

Per Kristian Lehre and Shishen Lin. No free lunch theorem and black-box complexity analysis for adversarial optimisation. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, Vancouver, Canada, 2024.

Marius Lindauer, Florian Karl, Anne Klier, Julia Moosbauer, Alexander Tornede, Andreas C Mueller, Frank Hutter, Matthias Feurer, and Bernd Bischl. Position: A call to action for a human-centered AutoML paradigm. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 30566–30584, Vienna, Austria, 2024.

Fei Liu, Tong Xialiang, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 32201–32223, Vienna, Austria, 2024a.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating Wikipedia by summarizing long sequences. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019.

Huakang Lu, Hong Qian, Yupeng Wu, Ziqi Liu, Ya-Lin Zhang, Aimin Zhou, and Yang Yu. Degradation-resistant offline optimization via accumulative risk control. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*, pp. 1609–1616, Kraków, Poland, 2023.

Zeyuan Ma, Hongshu Guo, Jiacheng Chen, Guojun Peng, Zhiguang Cao, Yining Ma, and Yue-Jiao Gong. LLaMoCo: Instruction tuning of large language models for optimization code generation. *arXiv:2403.01131*, 2024.

Alexandre Maraval, Matthieu Zimmer, Antoine Grosnit, and Haitham Bou Ammar. End-to-end meta-Bayesian optimisation with Transformer neural processes. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 11246–11260, New Orleans, LA, 2023.

Satvik Mehul Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative pretraining for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 24173–24197, Honolulu, HI, 2023.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Virtual, 2022.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 25444–25470, Honolulu, HI, 2023.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 3496–3506, Vancouver, Canada, 2019.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 16569–16594, Baltimore, MD, 2022.

Tung Nguyen and Aditya Grover. LICO: Large language models for in-context molecular optimization. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, 2025.

Tung Nguyen, Sudhanshu Agrawal, and Aditya Grover. ExPT: Synthetic pretraining for few-shot experimental design. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 45856–45869, New Orleans, LA, 2023.

Tung Nguyen, Qiuyi Zhang, Bangding Yang, Chansoo Lee, Jorg Bornschein, Sagi Perel, Yutian Chen, and Xingyou Song. Predicting from strings: Language model embeddings for Bayesian optimization. *arXiv:2410.10190*, 2024.

Ian Parberry. *Introduction to Game Physics with Box2D*. CRC Press, 2017.

Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 13226–13237, New Orleans, LA, 2022.

Hong Qian, Yiyi Zhu, Xiang Shu, Xin An, Yaolin Wen, Shuo Liu, Huakang Lu, Aimin Zhou, Ke Tang, and Yang Yu. SOO-Bench: Benchmarks for evaluating the stability of offline black-box optimization. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Maching Learning Research*, 21:5485–5551, 2020.

Martin Schlueter, Mehdi Neshat, Mohamed Wahib, Masaharu Munetomo, and Markus Wagner. GTOPX space mission benchmarks. *SoftwareX*, 14:100666, 2021.

Yunqi Shi, Ke Xue, Lei Song, and Chao Qian. Macro placement by wire-mask-guided black-box optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 6825–6843, New Orleans, LA, 2023.

Lei Song, Chenxiao Gao, Ke Xue, Chenyang Wu, Dong Li, Jianye Hao, Zongzhang Zhang, and Chao Qian. Reinforced in-context black-box optimization. *arXiv:2402.17423*, 2024a.

Xingyou Song and Dara Bahri. Decoding-based regression. *arxiv:2501.19383*, 2025.

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. OmniPred: Language models as universal regressors. *Transactions on Machine Learning Research*, 2024b.

Xingyou Song, Yingtao Tian, Robert Tjarko Lange, Chansoo Lee, Yujin Tang, and Yutian Chen. Position: Leverage foundational models for black-box optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 46168–46180, Vienna, Austria, 2024c.

Xingyou Song, Qiuyi Zhang, Chansoo Lee, Emily Fertig, Tzu-Kuo Huang, Lior Belenki, Greg Kochanski, Setareh Ariafar, Srinivas Vasudevan, Sagi Perel, and Daniel Golovin. The Vizier Gaussian process bandit algorithm. *arXiv: 2408.11527*, 2024d.

Samuel Stanton, Wesley J. Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. 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.

Rong-Xi Tan, Ke Xue, Shen-Huan Lyu, Haopu Shang, Yao Wang, Yaoyuan Wang, Sheng Fu, and Chao Qian. Offline model-based optimization by learning to rank. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, 2025.

Eric Tang, Bangding Yang, and Xingyou Song. Understanding LLM embeddings for regression. *Transactions on Machine Learning Research*, 2025.

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

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. 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.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. 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.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pp. 3–26, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 6000–6010, Long Beach, CA, 2017.

Shukuan Wang, Ke Xue, Lei Song, Xiaobin Huang, and Chao Qian. Monte Carlo tree search based space transfer for black box optimization. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, Vancouver, Canada, 2024.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 745–754, Lanzarote, Spain, 2018.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

Ke Xue, Rong-Xi Tan, Xiaobin Huang, and Chao Qian. Offline multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 55595–55624, Vienna, Austria, 2024.

Michael S. Yao, Yimeng Zeng, Hamsa Bastani, Jacob Gardner, James C. Gee, and Osbert Bastani. Generative adversarial model-based optimization via source critic regularization. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, Vancouver, Canada, 2024.

Peiyu Yu, Dinghuai Zhang, Hengzhi He, Xiaojian Ma, Ruiyao Miao, Yifan Lu, Yasi Zhang, Deqian Kong, Ruiqi Gao, Jianwen Xie, Guang Cheng, and Ying Nian Wu. Latent energy-based Odyssey: Black-box optimization via expanded exploration in the energy-based latent space. *arXiv:2405.16730*, 2024.

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

Ye Yuan, Can (Sam) Chen, Zixuan Liu, Willie Neiswanger, and Xue (Steve) Liu. Importance-aware co-teaching for offline model-based optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 55718–55733, New Orleans, LA, 2023.

Ye Yuan, Youyuan Zhang, Can (Sam) Chen, Haolun Wu, Zixuan Li, Jianmo Li, James J Clark, and Xue (Steve) Liu. Design editing for offline model-based optimization. *arXiv:2405.13964*, 2024.

Taeyoung Yun, Sujin Yun, Jaewoo Lee, and Jinkyoo Park. Guided trajectory generation with diffusion models for offline model-based optimization. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, Vancouver, Canada, 2024.

Fengxue Zhang, Yair Altas, Louise Fan, Kaustubh Vinchure, Brian Nord, and Yuxin Chen. Design of physical experiments via collision-free latent space optimization. In *Machine Learning and the Physical Sciences Workshop at NeurIPS'20*, Virtual, 2020.

Zhi-Hua Zhou, Yang Yu, and Chao Qian. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer, 2019.

A Related Work

Offline BBO. Offline BBO (Kim et al., 2025; Trabucco et al., 2022; Xue et al., 2024; Qian et al., 2025) methods can be generally categorized into two types, one of which is called **forward approach**. This category usually trains a scoring surrogate and maximizes the output to obtain the final design. However, these methods suffer from OOD issue such that errors made by the surrogate in OOD region would mislead the search procedure. To mitigate it, NEMO (Fu & Levine, 2021), COMs (Trabucco et al., 2021), IOM (Qi et al., 2022), RoMA (Yu et al., 2021), BOSS (Dao et al., 2024b) and IGNITE (Dao et al., 2024a) regulate the model from different perspectives, while Tri-Mentoring (Chen et al., 2023a) and ICT (Yuan et al., 2023) ensemble surrogates to enhance robustness. Besides, BDI (Chen et al., 2022a), MATCH-OPT (Hoang et al., 2024), and Cliqueformer (Kuba et al., 2024b;a) incorporate more information distilled from the offline dataset, and ARCOO (Lu et al., 2023), PGS (Chemingui et al., 2024), GABO (Yao et al., 2024), and DEMO (Yuan et al., 2024) employ different techniques to guide the model-inner search. Recently, Tan et al. (2025) point out that regression is not suitable for offline BBO and call for a paradigm shift to rank the designs. **Backward approach** typically fits a probabilistic model $p(\mathbf{x}|y)$, and samples promising designs from the model, which can be instantiated by prevalent generative models. For example, MINs (Kumar & Levine, 2020) uses GAN (Goodfellow et al., 2014), while DDOM (Krishnamoorthy et al., 2023), RGD (Chen et al., 2024), and DiffOPT (Kong et al., 2024) utilize diffusion model (Ho et al., 2020). Recent works also focus on learning from trajectories (Mashkaria et al., 2023; Yun et al., 2024), synthetic priors (Nguyen et al., 2023), or sampling-free latent variables (Yu et al., 2024).

LLM for BBO. Recent progress in LLM has demonstrated the potential applicability of LLM to solve BBO problems (Song et al., 2024c), mainly incorporating prompt-based, attention-based, token-based, and string-based methods. **Prompt-based** methods mimick the optimization via query-answer-based prompts using LLM. However, they do not involve fine-tuning the LLM parameters but only relies on the in-context learning abilities, which may limit optimization performance. EoH (Liu et al., 2024a), LLaMoCo (Ma et al., 2024), and LLMOPT (Jiang et al., 2025) generate optimization code via directly utilizing LLM, fine-tuning pre-trained LLM, and training from scratch, respectively. Many of other works employ LLM as components of BBO optimizers, e.g., LLAMBO (Liu et al., 2024b) simulates components of Bayesian optimization, such as regressor and acquisition function, with LLM. However, such method is limited by the training corpus and lacks interpretability. Given that **attention-based** Transformers (Vaswani et al., 2017) has superior scalability (Brown et al., 2020) and can learn in-context well (Garg et al., 2022), TNPs (Nguyen & Grover, 2022; Maraval et al., 2023) employs raw Transformers as in-context regressor for BBO. Note that Transformers trained with synthetic priors performs well on tabluar data (Hollmann et al., 2025) and Bayesian inference (Müller et al., 2022), works like PFNs4BO (Müller et al., 2023), LICO (Nguyen & Grover, 2025) and ExPT (Nguyen et al., 2023) utilize such priors for in-context optimization on both online and offline BBO. Attention-based methods exhibit superior performance in BBO field, but they usually optimize in a fixed search space. **Token-based** methods instantiate a design or a parameter as a token, and learn in-context from historical trajectories to simulate optimization. For example, OptFormer (Chen et al., 2022b) views a parameter value as a token for hyper-parameter optimization and employs Transformer structure, combined with metadata, to represent an optimization trajectory, while Dery et al. (2022) consider a multi-step improvement for OptFormer and Song et al. (2024a) plug a regret-to-go token into the algorithm history to solve BBO problems. Recent pioneering works utilize **string-based** representation on designs for BBO. OmniPred (Song et al., 2024b) regresses in multi-task suite based on string-based representation for arbitrary designs, metadata and scores, while Nguyen et al. (2024); Tang et al. (2025) use LLM embeddings to pre-train an in-context regressor for BBO. Motivated by the superior performance of these methods, in this work, we use string-based representation to solve universal offline BBO, as discussed in Section 3.

B Experimental Settings

We adopt the T5X architecture (Raffel et al., 2020) as our base model architecture, which can be accessed through the open-source repository. For embedder in UniSO-N and metadata embedder in UniSO-T, we directly leveraged the pretrained T5-small model. The important hyperparameters, most of which are defaulted.

C Details of Tasks and Dataset

In this section, we introduce details of different tasks and datasets we use in our experiments, which range across 7 test suites with 28 tasks, 222 trials, and a dataset size of 204K in total.

C.1 Design-Bench Tasks

Design-Bench (Trabucco et al., 2022)[5] is a famous benchmark suite for offline BBO. It includes various realistic tasks from real-world optimization problems, and each task corresponds to an oracle function for evaluation and a large static offline dataset. In this paper, we mainly consider 6 tasks in Design-Bench, and we directly use the open-sourced dataset of Design-Bench[6] as a part of the training data. Details of these tasks are as follows.

Ant and D'Kitty Morphology. These two tasks are both robot morphology optimization problems, and the goal is to optimize the morphological structure of two simulated robots: Ant from OpenAI Gym (Brockman et al., 2016) and D'Kitty from ROBEL (Ahn et al., 2020). Specifically, the objective of Ant Morphology is to make the Ant robot run quickly and that of D'Kitty Morphology is to let the D'Kitty robot move exactly to the predefined location. Both these tasks use a task-specific pre-trained Soft Actor Critic (Haarnoja et al., 2018) policy as controller, and simulate in the MuJoCo (Todorov et al., 2012) simulator with a timestep of 100. The design dimension of Ant Morphology is 56 and that of D'Kitty Morphology is 60, both of which include size, orientation, and location of the limbs. The dataset sizes of these two tasks are both 10004 with `DOUBLE` design spaces.

Superconductor. The objective of the Superconductor task (Hamidieh, 2018) is to maximize the critical temperature which derived from the chemical formula for a superconducting material. Although the actual critical temperature of a decoded material is inaccessible, which needs physical experiments, the evaluation is conducted using a pre-trained random forest regressor from Fannjiang & Listgarten (2020) as oracle function, which achieves a Spearman rank-correlation coefficient of 0.9210 on a held-out validation set. The design space consists of 86 `DOUBLE` variables, and the dataset size is 17010.

TF Bind 8 and TF Bind 10. The objective of TF Bind 8 and TF Bind 10 (Barrera et al., 2016) is to maximize the binding affinity score of the designed length 8 and length 10 DNA sequence with a prevalent human transcription factor `SIX6_REF_R1`. Scores are obtained via direct lookup from the exhaustive evaluation database established by Barrera et al. (2016). The design dimension is 8 for TF Bind 8 and 10 for TF Bind 10, and the design spaces are both `CATEGORICAL` where the number of categories of all dimensions are 4. The TF Bind 8 dataset contains 32898 design-score pairs. Although the TF Bind 10 dataset in Design-Bench includes 4161482 design-score pairs, which is too large and beyond our computation budget, we sample a subset with size of 10000 following recent work in the field of offline BBO (Tan et al., 2025).

We exclude two tasks in Design-Bench, following Yu et al. (2024); Yun et al. (2024); Tan et al. (2025). Specifically, we exclude ChEMBL (Gaulton et al., 2012) because almost all methods produce the same oracle prediction results, as shown in Krishnamoorthy et al. (2023); Mashkaria et al. (2023), which is not appropriate for comparison. We exclude synthetic NAS on CIFAR10 (Hinton et al., 2012) due to its high computation cost for exact evaluation over multiple seeds, which exceeds our computation resources. We also excluder Hopper Controller (Brockman et al., 2016) in Design-Bench since it has a normalization and de-normalization issue due to the stochastic policy (see https://github.com/brandontrabucco/design-bench/issues/8#issuecomment-1086758113 for details) and recent works in offline BBO (Yu et al., 2024; Yun et al., 2024; Tan et al., 2025) do not benchmark on this task as well.

C.2 SOO-Bench Tasks

SOO-Bench (Qian et al., 2025) is a recent benchmark for offline BBO, which not only provides more test problems but also proposes a novel method to evaluate the stability of the forward methods for offline BBO. We select the unconstraint tasks in SOO-Bench, GTOPX 2, GTOPX 3, GTOPX 4, and GTOPX 6 from GTOPX benchmark (Schlueter et al., 2021), except for hybrid 1 due to environment

[5] https://github.com/brandontrabucco/design-bench
[6] https://huggingface.co/datasets/beckhamc/design_bench_data

conflicts during installation. We use the open-source code of SOO-Bench[7] and generate task data following the default settings in the SOO-Bench paper. Specifically, for a given task, the dataset size is 1000 times the variable dimension and the data is uniformly drawn from the middle 50% of the overall distribution with respect to the score values. We set the random seed of the data generation procedure as 1 by default. Both these tasks are under DOUBLE design spaces and their evaluations are done through simulation library modules provided by Schlueter et al. (2021)[8]. Detailed description of our selected tasks are as follows.

GTOPX encompasses a comprehensive suite of real-world space trajectory optimization problems(Izzo & Manuel López-Ibáñez, 2022; Izzo, 2010). Specifically, **GTOPX 2** encompasses the optimization of an intricate interplanetary trajectory for a Saturn rendezvous mission, featuring 22 decision variables with the objective of minimizing the total velocity increment (ΔV) required throughout the mission. **GTOPX 3** and **GTOPX 4** both address trajectory optimization for Mercury missions, where the primary objective is to minimize the total mission ΔV. These problems are distinguished by their treatment of resonant flybys: GTOPX 3 explicitly excludes such maneuvers and operates in an 18-dimensional design space, while GTOPX 4 incorporates them within a 26-dimensional space. **GTOPX 6** focuses on the optimization of multi-gravity-assist trajectories targeting Comet 67P/Churyumov-Gerasimenko, employing 22 design variables to minimize the total ΔV requirements.

C.3 Real-World Tasks

Following Wang et al. (2024), we evaluate our method on three real-world tasks, LunarLander, RobotPush, and Rover, and use the open-sourced dataset by Wang et al. (2024). Detailed information of these tasks are as follows.

LunarLander. LunarLander is a 12-dimensional DOUBLE task implemented in OpenAI Gym (Brockman et al., 2016)[9] that aims to learn the parameters of a controller for a lunar lander. The objective is to maximize the expected average return over 50 randomly generated environments.

RobotPush. RobotPush aims to minimize the distance between a designated target location and a pair of robotically-controlled objects, where there are 14 DOUBLE controllable variables, e.g., orientation and speed. The function is implemented in Wang et al. (2018)[10] with a physics engine Box2D (Parberry, 2017).

Rover. Rover is a 2D trajectory optimization task that simulates a rover navigation task, which is defined by Wang et al. (2018)[11]. The trajectory is optimized within a 60-dimensional DOUBLE unit hypercube. A cost function $c(\cdot)$ is defined in the hypercube to measure the trajectory quality, and the objective is to minimize the total cost.

To generate diverse task trials, Wang et al. (2024) employ a transformation from RIBBO (Song et al., 2024a)[12] that introduces a scaling factor s and a translation vector $\mathbf{t}$, where the transformed objective is computed as $y = s \cdot f(\mathbf{x} - \mathbf{t})$. These factors are deterministically generated based on random seeds, allowing different seeds to map to distinct task. We randomly choose one seed for one task for evaluation.

D Additional Experimental Results

In this section, we provide additional experimental results, including ablation studies on UniSO-T and its comparison to traditional single-task offline BBO methods.

[7] https://anonymous.4open.science/r/SOO-Bench-9025

[8] https://www.midaco-solver.com/index.php/about/benchmarks/gtopx

[9] https://www.gymlibrary.dev/environments/box2d/lunar_lander

[10] https://github.com/zi-w/Ensemble-Bayesian-Optimization/blob/master/test_functions/push_function.py

[11] https://github.com/zi-w/Ensemble-Bayesian-Optimization/blob/master/test_functions/rover_function.py

[12] https://github.com/songlei00/RIBBO/blob/dcfbed5326a411e8c285d226a6899d922317c7d6/problems/real_world_problem.py#L139

Ablation studies: How do the metadata and designed improvement losses contribute to the performance? We additionally analyze the effects of each component of the proposed losses and metadata processing on UniSO-T. To better validate the effectiveness of the two components of the improved losses, we evaluate the model by removing each one. As shown in Table 4, the superior performance of UniSO-T over others show the contribution of each loss.

Table 4: Ablation studies of each component of the improved losses in UniSO-T few-shot performance on RobotPush, Rover, and LunarLander.

Task	$\mathcal{D}$(best)	UniSO-T	UniSO-T w/o $\mathcal{L}_{\text{lip}}$	UniSO-T w/o $\mathcal{L}_{\text{con}}$	UniSO-T w/o $\{\mathcal{L}_{\text{lip}}, \mathcal{L}_{\text{con}}\}$
Ant	165.326	**374.665 ± 56.057**	**292.644 ± 65.145**	232.399 ± 66.642	275.216 ± 90.820
D'Kitty	199.363	**225.752 ± 8.521**	175.796 ± 64.079	**224.668 ± 23.942**	216.070 ± 23.209
Superconductor	74.000	**92.200 ± 15.209**	**90.910 ± 6.538**	80.910 ± 14.995	86.795 ± 13.466
TF Bind 8	0.439	0.903 ± 0.041	0.916 ± 0.044	**0.945 ± 0.049**	**0.940 ± 0.027**
TF Bind 10	0.005	**0.823 ± 0.542**	0.623 ± 0.062	**0.823 ± 0.542**	**0.830 ± 0.539**
GTOPX 2	-195.586	**-72.848 ± 9.576**	-112.357 ± 39.361	**-87.305 ± 24.428**	-132.023 ± 63.084
GTOPX 3	-151.190	**-45.602 ± 8.433**	**-50.294 ± 6.184**	-56.067 ± 11.332	-60.941 ± 17.235
GTOPX 4	-215.716	**-84.271 ± 8.307**	-96.550 ± 12.745	**-84.152 ± 15.571**	-100.943 ± 15.044
GTOPX 6	-112.599	**-47.794 ± 11.943**	-67.276 ± 24.305	**-43.334 ± 8.402**	-71.749 ± 28.497
Avg. Rank	/	**1.667 ± 0.943**	2.889 ± 0.737	**2.333 ± 1.155**	3.111 ± 0.994

Furthermore, since our improvement takes metadata into account, we compare both improved and vanilla UniSO-T in cases with or without metadata. We present the universal offline BBO performance in Table 5, and the zero-shot and few-shot optimization results in Table 6 and Table 7, respectively. From the results in Table 5, vanilla UniSO-T combined with metadata generally performs better than without metadata, except for GTOPX 6, while improved UniSO-T performs similarly with or without metadata, which might result from incorporating metadata in the metadata guidance loss. However, the differences between these two cases significantly increase in the further zero-shot results in Table 6 and few-shot results in Table 7, where improved UniSO-T with metadata performs better than without metadata.

Table 5: Ablation studies of the metadata in UniSO-T on unconstrained tasks from Design-Bench and SOO-Bench. $\mathcal{D}$(best) denotes the best score in the offline dataset.

Task	$\mathcal{D}$(best)	Improved UniSO-T		Vanilla UniSO-T	
		w/ metadata	w/o metadata	w/ metadata	w/o metadata
Ant	165.326	**374.665 ± 56.057**	**358.379 ± 64.211**	275.216 ± 90.820	233.853 ± 85.996
D'Kitty	199.363	**225.752 ± 8.521**	**227.169 ± 12.278**	216.070 ± 23.209	198.221 ± 42.524
Superconductor	74.000	**92.200 ± 15.209**	**90.871 ± 10.611**	86.795 ± 13.466	88.715 ± 13.042
TF Bind 8	0.439	0.903 ± 0.041	**0.950 ± 0.025**	**0.940 ± 0.027**	0.935 ± 0.041
TF Bind 10	0.005	**0.823 ± 0.542**	0.651 ± 0.121	**0.830 ± 0.539**	**0.823 ± 0.542**
GTOPX 2	-195.586	**-72.848 ± 9.576**	**-79.864 ± 13.338**	-132.023 ± 63.084	-254.215 ± 414.579
GTOPX 3	-151.190	**-45.602 ± 8.433**	**-48.178 ± 12.638**	-60.941 ± 17.235	-75.579 ± 28.713
GTOPX 4	-215.716	**-84.271 ± 8.307**	**-79.887 ± 14.729**	-100.943 ± 15.044	-172.288 ± 196.360
GTOPX 6	-112.599	**-47.794 ± 11.943**	**-45.764 ± 7.685**	-71.749 ± 28.497	-69.066 ± 37.746
Avg. Rank	/	**1.778 ± 0.916**	**1.778 ± 0.916**	**2.889 ± 0.875**	3.556 ± 0.497

Table 6: Ablation studies of the metadata in UniSO-T zero-shot performance on RobotPush, Rover, and LunarLander. $\mathcal{D}$(best) denotes the best score in the offline dataset.

Task	$\mathcal{D}$(best)	Improved UniSO-T		Vanilla UniSO-T	
		w/ metadata	w/o metadata	w/ metadata	w/o metadata
RobotPush	0.102	**3.171 ± 0.984**	**2.517 ± 1.640**	2.451 ± 2.335	2.080 ± 0.877
Rover	-16.148	**-8.888 ± 2.119**	**-9.089 ± 3.070**	-11.188 ± 3.949	-14.920 ± 2.768
LunarLander	7.038	**31.186 ± 27.971**	6.251 ± 53.042	**21.183 ± 51.881**	-58.135 ± 40.322
Avg. Rank	/	**1.000 ± 0.000**	**2.333 ± 0.471**	2.667 ± 0.471	4.000 ± 0.000

Is the best-performing UniSO variant comparable to the well-studied expert single-task offline BBO methods? In Table 1 and Table 2, we use batch normalization for numerical inputs on single-task expert methods for a fair comparison. However, a more widely-used practice in the field of offline BBO is to employ a global z-score normalization for inputs (Trabucco et al., 2022). Thus, to better

Table 7: Ablation studies of the metadata in UniSO-T few-shot performance on RobotPush, Rover, and LunarLander. $\mathcal{D}$(best) denotes the best score in the offline dataset.

Task	$\mathcal{D}$(best)	Improved UniSO-T		Vanilla UniSO-T	
		w/ metadata	w/o metadata	w/ metadata	w/o metadata
RobotPush	0.102	**7.067 ± 0.169**	**6.155 ± 1.495**	3.071 ± 2.007	5.602 ± 1.939
Rover	-16.148	**-8.239 ± 1.270**	**-10.511 ± 2.070**	-13.451 ± 4.753	-10.833 ± 2.930
LunarLander	7.038	**248.573 ± 45.386**	**233.919 ± 60.467**	61.819 ± 118.019	214.205 ± 66.742
Avg. Rank	/	**1.000 ± 0.000**	**2.000 ± 0.000**	4.000 ± 0.000	3.000 ± 0.000

Table 8: Normalized score in Design-Bench, where the best and runner-up results on each task are **Blue** and **Violet**. $\mathcal{D}$(best) denotes the best score in the offline dataset. All methods are trained within on task, while UniSO-T are done in a multi-task manner. Results of all compared methods are referred from Tan et al. (2025).

Method	Venue	Ant	D'Kitty	Superconductor	TF-Bind-8	TF-Bind-10	Avg. Rank
$\mathcal{D}$(best)	/	0.565	0.884	0.400	0.439	0.467	/
BO-qEI	Baselines	0.812 ± 0.000	0.896 ± 0.000	0.382 ± 0.013	0.802 ± 0.081	0.628 ± 0.036	17.8 / 22
CMA-ES	Baselines	**1.712 ± 0.754**	0.725 ± 0.002	0.463 ± 0.042	0.944 ± 0.017	0.641 ± 0.036	11.4 / 22
REINFORCE	Baselines	0.248 ± 0.039	0.541 ± 0.196	0.478 ± 0.017	0.935 ± 0.049	**0.673 ± 0.074**	13.8 / 22
Grad. Ascent	Baselines	0.273 ± 0.023	0.853 ± 0.018	0.510 ± 0.028	0.969 ± 0.021	0.646 ± 0.037	11.2 / 22
Grad. Ascent Mean	Baselines	0.306 ± 0.053	0.875 ± 0.024	0.508 ± 0.019	**0.985 ± 0.008**	0.633 ± 0.030	10.6 / 22
Grad. Ascent Min	Baselines	0.282 ± 0.033	0.884 ± 0.018	**0.514 ± 0.020**	0.979 ± 0.014	0.632 ± 0.027	11.2 / 22
CbAS	ICML'19	0.846 ± 0.032	0.896 ± 0.009	0.421 ± 0.049	0.921 ± 0.046	0.630 ± 0.039	15.6 / 22
MINs	ICML'19	0.906 ± 0.024	0.939 ± 0.007	0.464 ± 0.023	0.910 ± 0.051	0.633 ± 0.034	12.6 / 22
DDOM	ICML'23	0.908 ± 0.024	0.930 ± 0.005	0.452 ± 0.028	0.913 ± 0.047	0.616 ± 0.018	14.2 / 22
BONET	ICML'23	0.921 ± 0.031	0.949 ± 0.016	0.390 ± 0.022	0.798 ± 0.123	0.575 ± 0.039	14.6 / 22
GTG	NeurIPS'24	0.855 ± 0.044	0.942 ± 0.017	0.480 ± 0.055	0.910 ± 0.040	0.619 ± 0.029	13.6 / 22
COMs	ICML'21	0.916 ± 0.026	0.949 ± 0.016	0.460 ± 0.040	0.953 ± 0.038	0.644 ± 0.052	9.0 / 22
RoMA	ICML'21	0.430 ± 0.048	0.767 ± 0.031	0.494 ± 0.025	0.665 ± 0.000	0.553 ± 0.000	18.0 / 22
IOM	NeurIPS'22	0.889 ± 0.034	0.928 ± 0.008	0.491 ± 0.034	0.925 ± 0.054	0.628 ± 0.036	12.6 / 22
BDI	NeurIPS'22	**0.963 ± 0.000**	0.941 ± 0.000	0.508 ± 0.013	0.973 ± 0.000	0.658 ± 0.000	**5.4 / 22**
ICT	NeurIPS'23	0.915 ± 0.024	0.947 ± 0.009	0.494 ± 0.026	0.897 ± 0.050	0.659 ± 0.024	8.8 / 22
Tri-Mentoring	NeurIPS'23	0.891 ± 0.011	0.947 ± 0.005	0.503 ± 0.013	0.956 ± 0.000	0.662 ± 0.012	7.0 / 22
PGS	AAAI'24	0.715 ± 0.046	**0.954 ± 0.022**	0.444 ± 0.020	0.889 ± 0.061	0.634 ± 0.040	13.4 / 22
FGM	AISTATS'24	0.923 ± 0.023	0.944 ± 0.014	0.481 ± 0.024	0.811 ± 0.079	0.611 ± 0.008	12.8 / 22
MATCH-OPT	ICML'24	0.933 ± 0.016	0.952 ± 0.008	0.504 ± 0.021	0.824 ± 0.067	0.655 ± 0.050	7.6 / 22
RaM-ListNet	ICLR'25	0.949 ± 0.025	**0.962 ± 0.015**	**0.517 ± 0.029**	**0.981 ± 0.012**	**0.670 ± 0.035**	**2.0 / 22**
UniSO-T (Ours)	/	0.850 ± 0.062	0.915 ± 0.015	0.489 ± 0.062	0.947 ± 0.036	**0.673 ± 0.136**	9.8 / 22

understand the performance of UniSO methods, in Table 8, we compare improved UniSO-T, the best-performing universal offline BBO method in Table 2, to a wide range of offline BBO methods, which are mainly based on z-score normalization, on Design-Bench. Here the improved UniSO-T method is trained solely using Design-Bench tasks, and the detailed results of compared methods are referred from a recent work on offline BBO (Tan et al., 2025). As shown in Table 8, UniSO-T achieved an average rank of 9.8 across 21 expert single-task offline BBO methods, outperforming some recently proposed methods specifically designed for single-task offline BBO, and even achieving the best results on TF-Bind-10. However, compared to the state-of-the-art single-task offline BBO methods, UniSO still has room for improvement, making it crucial to propose better techniques specifically for universal scenarios as important future work.

E METADATA IN THE EXPERIMENTS

In Table 9, we deliver all the metadata we used in our experiments.

Table 9: Metadata illustration we used in the experiments.

Name	Description	Objective
Ant Morphology	a quadruped robot morphology optimization	to run as fast as possible
D'Kitty Morphology	D'Kitty robot morphology optimization	to navigate the robot to a fixed location
Superconducor	critical temperature maximization	to design the chemical formula for a superconducting material that has a high critical temperature
TF Bind 8	DNA sequence optimization	to find the length-8 DNA sequence with maximum binding affinity with SIX6_REF_R1 transcription factor
TF Bind 10	DNA sequence optimization	to find the length-10 DNA sequence with maximum binding affinity with SIX6_REF_R1 transcription factor
Cassini 2	Complex interplanetary missions to Saturn	to achieve a rendezvous with Saturn, aiming to minimize the total velocity change
Messenger (reduced)	Simulation of interplanetary missions to Mercury	to minimize the total velocity change over the course of the mission
Messenger (full)	Interplanetary missions to Mercury, with resonant flybys of the planet	to minimize the total velocity change incurred throughout the mission
Rosetta	Simulation of multi-gravity-assisted space missions to Comet 67P/Churyumov-Gerasimenko	to minimize the total velocity change required throughout the mission