npj | artificial intelligence

Article

https://doi.org/10.1038/s44387-025-00055-1

Learning design-score manifold to guide diffusion models for offline optimization

Check for updates

Tailin Zhou[1,2,3], Zhilin Chen[3], Wenlong Lyu[3], Zhitang Chen[3], Danny H. K. Tsang[1,2] & Jun Zhang[1]

Optimizing complex systems—from discovering therapeutic drugs to designing high-performance materials—remains a fundamental challenge across science and engineering, as the underlying rules are often unknown and costly to evaluate. Offline optimization aims to optimize designs for target scores using pre-collected datasets without system interaction. However, conventional approaches may fail beyond training data, predicting inaccurate scores and generating inferior designs. This paper introduces ManGO, a diffusion-based framework that learns the design-score manifold, capturing the design-score interdependencies holistically. Unlike existing methods that treat design and score spaces in isolation, ManGO unifies forward prediction and backward generation, attaining generalization beyond training data. Key to this is its derivative-free guidance for conditional generation, coupled with adaptive inference-time scaling that dynamically optimizes denoising paths. Extensive evaluations demonstrate that ManGO outperforms 24 single- and 10 multi-objective optimization methods across diverse domains, including synthetic tasks, robot control, material design, DNA sequence, and real-world engineering optimization.

Across scientific and industrial domains, from drug discovery [1] to engineering superconducting materials [2], researchers face a common bottleneck: creating new designs to optimize specific property scores in complex systems where the rules governing performance are unknown or costly to evaluate. Numerous methods require either expensive trial-and-error experimentation or building system models with long-term accumulation of domain knowledge. Consider the decades-long quest for fusion reactor materials—each physical test costs millions and risks equipment damage [3] or the ethical constraints in developing neuroactive drugs where failed designs could harm patients [4, 5]. These limitations call for a general optimization framework that learns directly from historical data while eliminating the need for iterative evaluation.

Offline optimization [6], also named offline model-based optimization, has emerged as a promising solution, enabling design improvement using pre-collected datasets. This approach has proven valuable in molecule generation [7], protein properties [8], and hardware accelerators [9]. Current methods adopt a unidirectional strategy: (i) Training surrogate models to predict scores from designs. These predicted scores are then utilized by various optimizers to identify the optimal designs (forward modeling) [10, 11]; (ii) Generating designs that are conditioned on desired scores through the use of generative models (backward generation). An example of this approach is the employment of diffusion models [12], wherein small amounts of noise are incrementally added to a design sample, and a neural network is trained to reverse this noise-adding procedure. However, forward methods mislead optimizers with overconfident out-of-distribution predictions [6], while backward methods struggle with out-of-distribution generation (OOG) on unseen conditions [13]. These struggles stem from a deeper oversight: existing works operate in isolated design or score spaces, missing the underlying design-score manifold where optimal designs reside.

We propose to learn the design-score manifold to guide diffusion models for offline optimization (ManGO). As illustrated in Fig. 1, we leverage score-augmented datasets to train an unconditional diffusion model. By learning on the manifold, the model effectively captures the bidirectional relationships between designs and their corresponding scores. We then introduce a derivative-free guidance mechanism for conditional generation, which enables bidirectional guidance—generating designs based on target scores and predicting scores for given designs, thus eliminating the reliance on error-prone forward models. To further enhance generation quality, we implement adaptive inference-time scaling for ManGO, which computes model fidelity on unconditional samples. This scaling approach dynamically optimizes denoising paths through ManGO's self-supervised rewards. Moreover, ManGO is adaptable to both single-objective optimization (SOO) and multi-objective optimization (MOO) tasks, making it a comprehensive solution for various optimization scenarios.

The comprehensive evaluation showcases ManGO's state-of-the-art versatility in offline optimization, consistently surpassing existing approaches across both SOO and MOO tasks. In comparison with the baselines,

[1]The Hong Kong University of Science and Technology, Hong Kong, China. [2]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. [3]Huawei Technologies Co. Ltd., Shenzhen, China. e-mail: eejzhang@ust.hk

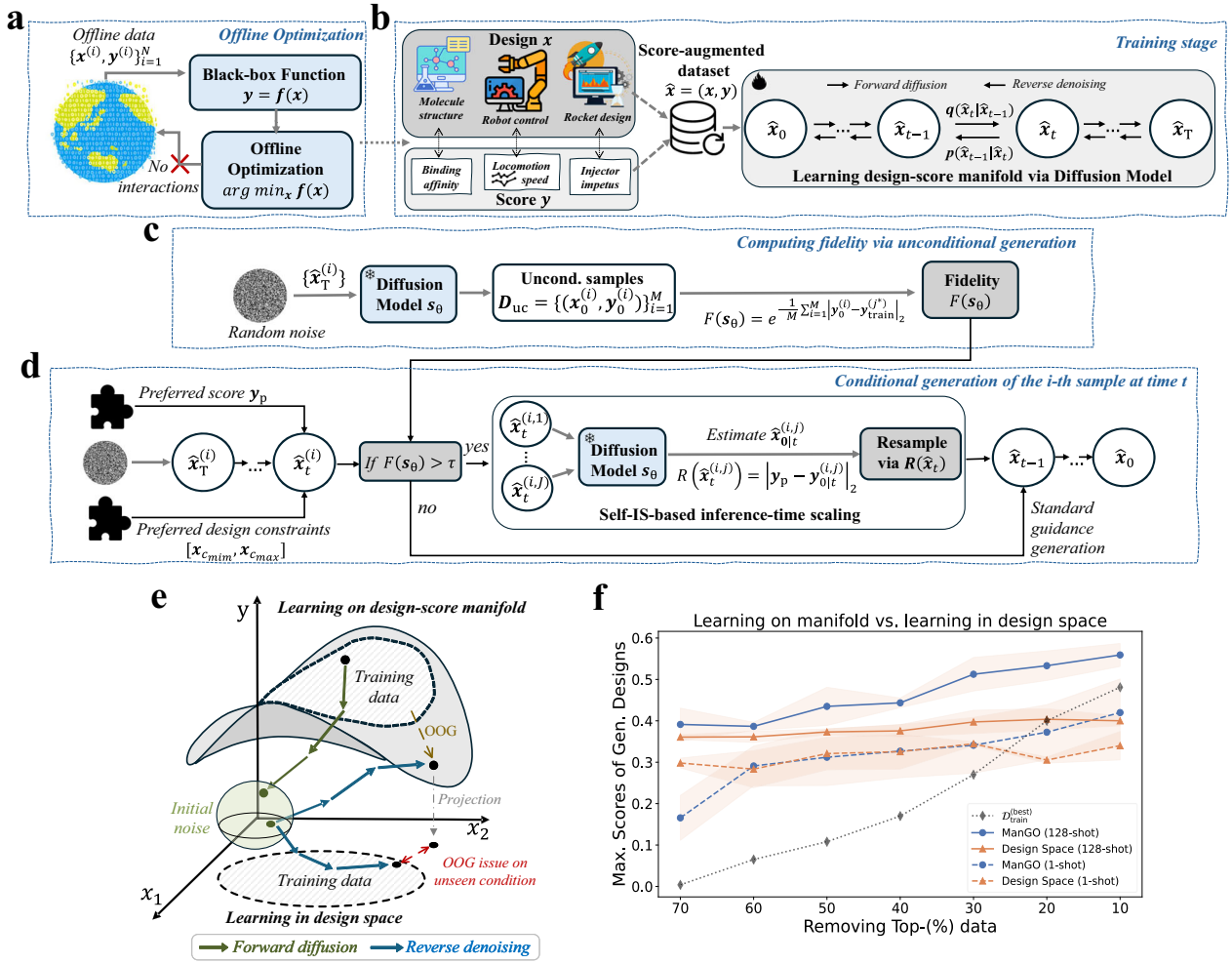

Fig. 1 | Overview of the ManGO framework for offline optimization. a Illustration of offline optimization: it identifies optimal designs for an unknown black-box function using an offline dataset (no environment interaction), where designs represent function inputs and scores correspond to outputs. **b** Training a diffusion model on score-augmented data to learn the joint design-score manifold. **c** Fidelity estimation via unconditional samples generated by the trained ManGO model: the fidelity metric determines whether to activate inference-time scaling during conditional generation. **d** Bidirectional conditional generation: it leverages preferred-score or preferred-design conditions to generate corresponding designs or scores, illustrated via the self-supervised importance sampling (self-IS) method at denoising timestep t for sample i. **e** Conceptual illustration of ManGO: it learns on the design-score manifold to enhance out-of-distribution generation (OOG) capability, contrasted with design-space learning that struggles with OOG issues under unseen conditions [13]. **f** Case study on *superconductor's temperature optimization* [37]: it demonstrates the superior OOG performance via ManGO versus the design-space approach (i.e., DDOM) across varying ratios of top data removal.

ManGO significantly enhances performance, attaining the top position among 24 SOO methods and 10 MOO methods. We anticipate that ManGO will serve as a valuable tool for data-driven design. By unifying optimization through learning on the design-score manifold, it offers a scalable and accessible solution for future scientific and industrial challenges.

Results

In this section, we first introduce preliminaries on offline optimization, the basics of diffusion models, baseline methods and performance metrics. Subsequently, to show the motivation and advantages of ManGO, we compare the learned versus the original design-score manifold and visualize the trajectory generation. We then conduct extensive experimental validation on offline SOO and MOO using Design-Bench [6] and Off-MOO-Bench [14]. Finally, we perform systematic ablation studies to analyze the contributions of ManGO's core components.

Preliminaries

Offline optimization [6], also referred to as offline model-based optimization, seeks to identify an optimal design $\boldsymbol{x}^*$ within a design space $\mathcal{X} \subseteq \mathbb{R}^d$ without requiring online evaluations, where d denotes the design dimension. Based on the number of objective functions $\boldsymbol{f}(\cdot) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) : \mathcal{X} \to \mathbb{R}^m$, offline optimization can be classified into two types [14]: (i) offline SOO when $m = 1$, and (ii) offline MOO when $m > 1$.

Offline SOO aims to identify the optimal design $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ using only a pre-collected offline dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i$ denotes a specific design (also referred to as a solution) and $y_i = f(\boldsymbol{x}_i)$ represents its corresponding score (or objective value). Offline MOO aims to identify a set of designs that achieve optimal trade-offs among conflicting objectives using a pre-collected dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, where $\boldsymbol{y}_i$ denotes the vector of scores corresponding to design $\boldsymbol{x}_i$. The problem is defined as [15]: Find $\boldsymbol{x}^* \in \mathcal{X}$ such that $\nexists \boldsymbol{x} \in \mathcal{X}$ with $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$, where $\prec$ denotes Pareto dominance. A design $\boldsymbol{x}'$ is said to *Pareto dominate* another design $\boldsymbol{x}$, denoted as $\boldsymbol{f}(\boldsymbol{x}') \prec \boldsymbol{f}(\boldsymbol{x})$, if $\exists i \in \{1, \ldots, m\}, f_i(\boldsymbol{x}') < f_i(\boldsymbol{x})$ and $\forall j \in \{1, \ldots, m\}, f_j(\boldsymbol{x}') \leq f_j(\boldsymbol{x})$. Namely, $\boldsymbol{x}'$ is superior to $\boldsymbol{x}$ in at least one objective while being at least as good in all others. A design $\boldsymbol{x}^*$ is *Pareto optimal* if no other design $\boldsymbol{x} \in \mathcal{X}$ Pareto dominates $\boldsymbol{x}^*$. The set of all Pareto optimal designs is referred to as the *Pareto set (PS)*, and the set of their scores $\{\boldsymbol{f}(\boldsymbol{x}^*) | \boldsymbol{x}^* \in \text{PS}\}$ constitutes the *Pareto front*. The goal of offline MOO is to identify the PS

using a pre-collected dataset, thereby achieving optimal trade-offs among conflicting objectives.

Diffusion models are a type of deep generative models that learn to reverse a gradual noising process, transforming random noise into realistic data through iterative denoising. Let $\boldsymbol{x}_t$ denote the state of a data sample $\boldsymbol{x}_0$ at time $t \in [0, T]$, where $\boldsymbol{x}_0$ is drawn from an unknown data distribution $p_0(\boldsymbol{x})$. Here, $\boldsymbol{x}_t$ represents a noisy version of $\boldsymbol{x}_0$ at time t, and $\boldsymbol{x}_T$ corresponds to a point sampled from a prior noise distribution $p_T(\boldsymbol{x})$, typically chosen as the standard normal distribution $p_T(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

The forward diffusion process, also known as the noise-adding process, can be modeled as a stochastic differential equation (SDE) [16]: $d\boldsymbol{x} = \mathbf{f}(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$, where $\boldsymbol{w}$ denotes the standard Wiener process, $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is the drift coefficient, and $g(t) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient of $\boldsymbol{x}_t$. The denoising process is defined by the reverse-time SDE: $d\boldsymbol{x} = \left[\mathbf{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] dt + g(t)d\tilde{\boldsymbol{w}}$, where dt represents an infinitesimal step backward in time, and $d\tilde{\boldsymbol{w}}$ is the reverse-time Wiener process.

Baseline methods and performance metrics

We consider existing baseline methods for offline SOO based on three methodological paradigms: (i) *Surrogate-based methods*: optimizing with surrogate models, including BO-qEI [17, 18], CMA-ES [10], REINFORCE [19], Gradient Ascent and its variants of mean ensemble and min ensemble. (ii) *Forward-modeling methods*: employing advanced neural networks like generative models as surrogate models and integrating with surrogate-based methods, including COMs [20], RoMA [21], IOM [22], BDI [11], ICT [23], Tri-Mentoring [24], PGS [25], FGM [26], Match-OPT [27], and RaM [28]. (iii) *Inverse-modeling methods*: applying score as a condition to reverse design with generative models, including CbAS [29], MINs [30], DDOM [12], BONET [31], and GTG [32].

For offline MOO, existing approaches remain relatively underexplored compared to offline SOO. Our evaluation focuses on three representative approaches: (i) *Multiple Models (MMs)-based* NSGA-II: We implement NSGA-II with independent objective predictors and perform predictors' ensemble as the surrogate model for evolutionary optimization, which outperforms end-to-end and multi-head variants [14]. (ii) *Multi-objective Bayesian Optimization* (MOBO): We adapt the canonical MOBO by substituting Gaussian Processes with the MM ensemble and employ an HV-based acquisition function, qNEHVI [33], which outperforms scalarization and information-theoretic alternatives [14]. (iii) *Generative methods*: ParetoFlow [34], a flow-model-based method utilizing adaptive weights for multiple predictors to guide flow sampling toward PF; MO-DDOM, a diffusion-model-based method where we extend DDOM through multi-score conditioning and adding MM-based design evaluation.

In offline optimization, where environment interaction is prohibited, it is essential to evaluate multiple candidate solutions; thus, standard benchmarks adopt k-shot evaluation with the 100th percentile (best candidate) as the performance metric [6, 14]. All results are normalized using task-specific references for comparison: For Design-Bench with maximization tasks (Table 1), we use standardization normalization for the Superconduct task and min-max normalization for other tasks based on the unobserved

Table 1 | The 100th percentile normalized score (k = 128) in the Design-Bench benchmark, where the best and runner-up results on each task are bold and underlined numbers

Method $\mathcal{D}_{train}^{(best)}$ (Preferred)	Ant 0.565 (1.0)	D'Kitty 0.884 (1.0)	Superconductor 0.400 (1.0)	TF-Bind-8 0.439 (1.0)	TF-Bind-10 0.467 (1.0)	Mean Rank /
BO-qEI	0.812 ± 0.000	0.896 ± 0.000	0.382 ± 0.013	0.802 ± 0.081	0.628 ± 0.036	20.0 / 24
CMA-ES	**1.214 ± 0.732**	0.725 ± 0.002	0.463 ± 0.042	0.944 ± 0.017	0.641 ± 0.036	13.0 / 24
REINFORCE	0.248 ± 0.039	0.541 ± 0.196	0.478 ± 0.017	0.935 ± 0.049	0.673 ± 0.074	16.0 / 24
Grad. Ascent	0.273 ± 0.023	0.853 ± 0.018	0.510 ± 0.028	0.969 ± 0.021	0.646 ± 0.037	13.6 / 24
Grad. Ascent Mean	0.306 ± 0.053	0.875 ± 0.024	0.508 ± 0.019	0.985 ± 0.008	0.633 ± 0.030	13.2 / 24
Grad. Ascent Min	0.282 ± 0.033	0.884 ± 0.018	0.514 ± 0.020	0.979 ± 0.014	0.632 ± 0.027	13.5 / 24
COMs	0.916 ± 0.026	0.949 ± 0.016	0.460 ± 0.040	0.953 ± 0.038	0.644 ± 0.052	11.5 / 24
RoMA	0.430 ± 0.048	0.767 ± 0.031	0.494 ± 0.025	0.665 ± 0.000	0.553 ± 0.000	20.3 / 24
IOM	0.889 ± 0.034	0.928 ± 0.008	0.491 ± 0.034	0.925 ± 0.054	0.628 ± 0.036	15.1 / 24
BDI	0.963 ± 0.000	0.941 ± 0.000	0.508 ± 0.013	0.973 ± 0.000	0.658 ± 0.000	7.7 / 24
ICT	0.915 ± 0.024	0.947 ± 0.009	0.494 ± 0.026	0.897 ± 0.050	0.659 ± 0.024	11.4 / 24
Tri-Mentoring	0.891 ± 0.011	0.947 ± 0.005	0.503 ± 0.013	0.956 ± 0.000	0.662 ± 0.012	9.7 / 24
PGS	0.715 ± 0.046	0.954 ± 0.022	0.444 ± 0.020	0.889 ± 0.061	0.634 ± 0.040	15.2 / 24
FGM	0.923 ± 0.023	0.944 ± 0.014	0.481 ± 0.024	0.811 ± 0.079	0.611 ± 0.008	15.2 / 24
Match-OPT	0.933 ± 0.016	0.952 ± 0.008	0.504 ± 0.021	0.824 ± 0.067	0.655 ± 0.050	10.0 / 24
RaM-RankCosine	0.940 ± 0.028	0.951 ± 0.017	0.514 ± 0.026	0.982 ± 0.012	0.675 ± 0.049	4.5 / 24
RaM-ListNet	0.949 ± 0.025	0.962 ± 0.015	0.517 ± 0.029	0.981 ± 0.012	0.670 ± 0.035	4.2 / 24
CbAS	0.846 ± 0.032	0.896 ± 0.009	0.421 ± 0.049	0.921 ± 0.046	0.630 ± 0.039	17.5 / 24
MINs	0.906 ± 0.024	0.939 ± 0.007	0.464 ± 0.023	0.910 ± 0.051	0.633 ± 0.034	15.0 / 24
BONET	0.921 ± 0.031	0.949 ± 0.016	0.390 ± 0.022	0.798 ± 0.123	0.575 ± 0.039	17.1 / 24
GTG	0.855 ± 0.044	0.942 ± 0.017	0.480 ± 0.055	0.910 ± 0.040	0.619 ± 0.029	15.9 / 24
DDOM	0.908 ± 0.024	0.930 ± 0.005	0.452 ± 0.028	0.913 ± 0.047	0.616 ± 0.018	16.6 / 24
ManGO	0.960 ± 0.017	**0.971 ± 0.004**	0.523 ± 0.040	**0.985 ± 0.004**	0.673 ± 0.033	2.2 / 24
ManGO$^{+\text{Self-IS}}$	0.968 ± 0.013	0.969 ± 0.009	**0.543 ± 0.037**	**0.985 ± 0.004**	**0.679 ± 0.023**	**1.4 / 24**

$\mathcal{D}_{\text{train}}^{(\text{best})}$ denotes the best score in the offline training dataset. Each task's results are normalized by the best scores in the unobserved dataset. DDOM represents a conditional diffusion-based method.

Table 2 | Averaged normalized HV and IGD values of synthetic tasks (upper) and RE tasks (lower) in the Off-MOO benchmark, where the best and runner-up results on each task are highlighted by bold and underlined numbers.

	ZDT(n=2)		OmniTest(n=2)		DTLZ(n=3)		Mean Rank	
$\mathcal{D}_{train}^{(best)}$ (Preferred)	Avg. HV(↑) 1.0 (1.118)	Avg. IGD(↓) 1.0 (0.0)	Avg. HV(↑) 1.0 (1.056)	Avg. IGD(↓) 1.0 (0.0)	Avg. HV(↑) 1.0 (1.098)	Avg. IGD(↓) 1.0 (0.0)	HV Rank(↓) /	IGD Rank(↓) /
MM-NSGA2(k=1)	0.888 ± 0.013	2.555 ± 0.118	1.044 ± 0.010	0.454 ± 0.129	0.987 ± 0.012	0.635 ± 0.048	8.3 / 10	6.7 / 10
MO-DDOM (k=1)	0.948 ± 0.009	1.674 ± 0.150	0.983 ± 0.002	0.827 ± 0.046	0.932 ± 0.014	0.795 ± 0.063	9.3 / 10	8.0 / 10
ManGO (k=1)	1.097 ± 0.004	0.729 ± 0.064	1.050 ± 0.002	0.399 ± 0.068	0.971 ± 0.016	0.739 ± 0.104	6.0 / 10	6.0 / 10
ManGO$^{+\text{Self-IS}}$ (k=1)	1.099 ± 0.005	0.702 ± 0.065	1.050 ± 0.001	0.359 ± 0.017	1.053 ± 0.015	0.250 ± 0.084	4.3 / 10	3.7 / 10
MM-MOBO	0.963 ± 0.007	4.723 ± 0.164	**1.056 ± 0.000**	0.206 ± 0.019	1.075 ± 0.000	0.362 ± 0.016	4.0 / 10	6.0 / 10
ParetoFlow	1.000 ± 0.008	2.867 ± 0.405	0.953 ± 0.057	1.523 ± 0.567	0.998 ± 0.009	0.672 ± 0.115	7.7 / 10	8.3 / 10
MM-NSGA2 (k=256)	1.055 ± 0.003	3.592 ± 0.044	1.046 ± 0.002	1.008 ± 0.019	**1.086 ± 0.000**	0.752 ± 0.016	4.0 / 10	9.0 / 10
MO-DDOM (k=256)	0.981 ± 0.006	1.052 ± 0.144	1.033 ± 0.001	0.270 ± 0.010	1.054 ± 0.006	0.255 ± 0.044	6.7 / 10	4.3 / 10
ManGO (k=256)	**1.107 ± 0.002**	**0.420 ± 0.030**	1.051 ± 0.002	0.118 ± 0.015	1.066 ± 0.003	0.172 ± 0.013	2.7 / 10	1.7 / 10
ManGO$^{+\text{Self-IS}}$ (k=256)	1.106 ± 0.002	0.445 ± 0.043	1.052 ± 0.000	**0.094 ± 0.002**	1.079 ± 0.003	**0.123 ± 0.033**	**2.0 / 10**	**1.3 / 10**
	RE(n=2)		**RE(n=3)**		**RE(n=4)**		**Mean Rank**	
	Avg. HV(↑)	**Avg. IGD(↓)**	**Avg. HV(↑)**	**Avg. IGD(↓)**	**Avg. HV(↑)**	**Avg. IGD(↓)**	**HV Rank(↓)**	**IGD Rank(↓)**
$\mathcal{D}_{train}^{(best)}$ (Preferred)	1.0 (1.037)	1.0 (0.0)	1.0 (1.082)	1.0 (0.0)	1.0 (1.310)	1.0 (0.0)	/	/
MM-NSGA2(k=1)	1.016 ± 0.004	56.958 ± 12.159	0.935 ± 0.007	2.979 ± 0.159	0.780 ± 0.009	1.504 ± 0.060	9.3 / 10	9.7 / 10
MO-DDOM (k=1)	1.010 ± 0.012	2.261 ± 0.220	1.046 ± 0.002	0.579 ± 0.020	1.058 ± 0.003	0.845 ± 0.008	8.7 / 10	4.7 / 10
ManGO (k=1)	1.024 ± 0.004	6.809 ± 0.740	1.051 ± 0.011	0.782 ± 0.083	1.234 ± 0.009	0.421 ± 0.018	5.7 / 10	6.0 / 10
ManGO$^{+\text{Self-IS}}$ (k=1)	1.022 ± 0.004	4.717 ± 2.081	1.066 ± 0.005	0.588 ± 0.027	1.240 ± 0.016	0.304 ± 0.020	4.7 / 10	4.3 / 10
MM-MOBO	1.027 ± 0.002	**1.156 ± 0.133**	1.071 ± 0.001	0.912 ± 0.093	1.123 ± 0.009	0.816 ± 0.029	3.7 / 10	5.0 / 10
ParetoFlow	1.017 ± 0.004	12.403 ± 0.052	1.010 ± 0.004	1.806 ± 0.024	0.668 ± 0.001	1.887 ± 0.006	9.0 / 10	8.7 / 10
MM-NSGA2 (k=256)	**1.034 ± 0.000**	98.945 ± 0.011	1.069 ± 0.001	2.160 ± 0.029	1.249 ± 0.012	0.320 ± 0.052	2.0 / 10	6.7 / 10
MO-DDOM (k=256)	1.020 ± 0.001	1.476 ± 0.031	1.056 ± 0.001	0.552 ± 0.007	1.073 ± 0.004	0.808 ± 0.008	6.7 / 10	3.3 / 10
ManGO (k=256)	1.026 ± 0.002	3.373 ± 0.428	1.063 ± 0.005	0.611 ± 0.026	1.248 ± 0.006	0.388 ± 0.014	4.0 / 10	4.7 / 10
ManGO$^{+\text{Self-IS}}$ (k=256)	1.028 ± 0.001	2.264 ± 0.237	**1.072 ± 0.002**	**0.498 ± 0.013**	**1.264 ± 0.002**	**0.234 ± 0.020**	**1.3 / 10**	**2.0 / 10**

Higher HV values indicate better performance, while lower IGD values are preferred. $\mathcal{D}_{train}^{(best)}$ (Preferred) denotes the best (preferred) HV/IGD in the offline training dataset. For compact presentation, reported numbers represent tasks' performance averaged by the objective number n. The sets of ZDT(n = 2), OmniTest (n = 2), and DTLZ (n = 3) consist of 5 ZDT tasks [41], 1 OmniTest task [52], and 2 DTLZ tasks [42], respectively. The sets of RE (n = 2), RE (n = 3), and RE(n = 4) comprise 5, 7, and 2 real-world application tasks [43] with n = 2, 3, and 4, respectively. Note that each task's results are normalized by the best HV and IGD of its training dataset, and RE (n = 2) presents higher averaged IGD values because the RE22 task has ten times more IGD value than other tasks. Note that MO-DDOM represents a standard conditional diffusion-based baseline method.

dataset's highest score. For the Off-MOO-Bench with minimization tasks (Tables 2), we use min-max normalization with the best HV and IGD values of training datasets.

Motivation and Advantages of ManGO

Compared to conventional manifold learning, our diffusion-based approach provides unique advantages for offline optimization. Unlike conventional methods like kernel-based methods that struggle with complex nonlinear geometries [35], diffusion models excel at capturing intricate manifold structures through stochastic denoising. Crucially, diffusion models inherently support conditional generation, enabling direct generation of high-performing designs conditioned on target scores-a capability absent in standard manifold learning. Furthermore, compared to GANs or VAEs, diffusion models offer superior training stability and generation fidelity [36], which are critical for reliable optimization from offline data. This combination of strong representational capacity and built-in conditional generation makes diffusion models suited for learning the design-score manifold.

Compared to existing methods for offline optimization, our proposed ManGO framework explicitly learns the design-score manifold and leverages the underlying manifold geometry to co-generate designs and scores. Specifically, ManGO learns a bidirectional generation between designs and scores: (i) *Design-to-Score Prediction*: Given any design configuration, ManGO predicts its corresponding score; (ii) *Score-to-Design Generation*: For any preferred score, ManGO generates its corresponding design. As illustrated in Figure 1e, this bidirectional mapping provides ManGO with a robust OOG capability to extrapolate beyond training distributions. Let $\hat{\boldsymbol{x}}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ denotes a score-augmented design vector at timestep t, where $\boldsymbol{x}$ and $\boldsymbol{y}$ represent the design and its score, respectively. The denoising update can be represented as:

$$\hat{\boldsymbol{x}}_{t-1} = (\boldsymbol{x}_t, \boldsymbol{y}_t) + \underbrace{(\Delta\boldsymbol{x}_t, \Delta\boldsymbol{y}_t)}_{\text{Co-update}} = \frac{1}{2}\underbrace{(\boldsymbol{x}_t + 2\Delta\boldsymbol{x}_t, \boldsymbol{y}_t)}_{\text{Design update}} + \frac{1}{2}\underbrace{(\boldsymbol{x}_t, \boldsymbol{y}_t + 2\Delta\boldsymbol{y}_t)}_{\text{Score update}},$$

where the co-update term indicates the denoising update based on the learned manifold geometry, the design-update term indicates the denoising update of $\boldsymbol{x}_t$ conditioned on $\boldsymbol{y}_t$, and the score-update term indicates the denoising update of $\boldsymbol{y}_t$ conditioned on $\boldsymbol{x}_t$. Unlike the design-space approaches that treat $\boldsymbol{y}_t$ as a fixed condition and only update $\boldsymbol{x}_t$ unidirectionally, ManGO jointly updates both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ based on the bidirectional mapping. The design-space methods struggle with extrapolation under unseen conditions. In contrast, ManGO leverages co-updates to dynamically capture the geometric relationship between design and score: each denoising step not only pushes $\boldsymbol{x}_t$ toward the local manifold conditioned on $\boldsymbol{y}_t$ (design-update) but also refines $\boldsymbol{y}_t$ to align with the evolving $\boldsymbol{x}_t$ (score-update). This bidirectional feedback enables progressive

extrapolation and converges to the conditioned points on the manifold, achieving a robust OOG capability.

We conduct a controlled experiment to elucidate the advantage of the bidirectional mapping of ManGO on a *superconductor* [37] task, an 86-D materials design optimization task to maximize a critical temperature, using DDOM [12] as the design-space baseline. In terms of the 128-shot evaluation, Figure 1f shows that ManGO achieves a consistent score gain of greater than 0.1 over the design-space approach across varying levels of top-data removal (from 70% to 10%). The gain grows to nearly 0.2 when only 10% of the top data are removed. Regarding the 1-shot evaluation, both methods exhibit comparable performance under severe data removal (from 70% to 30%). However, ManGO demonstrates progressively better results as data availability increases. At 10% data removal, ManGO achieves higher scores than the design-space method with 128 shots. These results confirm that: (i) ManGO captures the bidirectional design-score relationships, enabling robustness to OOG challenges; (ii) ManGO exhibits nonlinear scaling of sample efficiency with data quality, achieving superior few-shot performance.

Manifold and trajectory generation visualization

To demonstrate the bidirectional mapping capability, we visualize the performance of ManGO on two canonical minimization tasks. (i) *Branin function* (for SOO): A well-studied 2D function containing three global minima within $x_1 \in [-5, 10]$, $x_2 \in [0, 15]$ and $y_{\min} = 0.398$, serving as an ideal testbed to capture multimodal landscapes. Specifically, $f_{\text{br}}(x_1, x_2) = a\left(x_2 - bx_1^2 + cx_1 - r\right)^2 - s(1-t)\cos x_1 - s$, where $a = 1, b = \frac{5.1}{4\pi^2}$, $c = \frac{5}{\pi}, r = 6, s = 10$, and $t = \frac{1}{8\pi}$. (ii) *OmniTest* (for MOO): A synthetic 2D problem generating 9 disconnected Pareto-optimal points with $\boldsymbol{x} \in [0, 6]^2$ and $\boldsymbol{y} \in [-2, 2]^2$, challenging optimization methods in maintaining diverse designs. Specifically, $f_1(\boldsymbol{x}) = \sum_{i=1}^{2} \sin(\pi x_i), f_2(\boldsymbol{x}) = \sum_{i=1}^{2} \cos(\pi x_i)$ with Pareto designs at all combinations of $(x_1, x_2) \in \{1, 3, 5\} \times \{1, 3, 5\}$.

Fig. 2 shows that ManGO reconstructs the entire design-score manifold despite removing the top 40% of low-scoring data. Its generated manifold recovers the erased global minima locations and maintains the overall topographic trends in the Branin task in Figure 2a. ManGO also recovers all 9 disconnected Pareto fronts and preserves the negative correlation between f_1 and f_2 objectives in the OmniTest task in Figure 2b (visualizing two regions for ease of observation). Across both tasks, the generated manifolds exhibit only minimal deviations from the original manifolds, even in out-of-distribution regions. This demonstrates ManGO's robust OOG capability, validating its ability to extrapolate beyond the training distribution accurately. Furthermore, both generated manifolds exhibit consistent minor elevation, with a deviation of less than 5%. For example, Figure 2c reveals a gradual score elevation between unconditional and expected scores in Branin's training region, where higher expected scores correspond to sparser training samples. This conservative estimation under uncertainty serves as an advantage in offline optimization, where reliable performance outweighs aggressive extrapolation [20].

Unlike design-space approaches limited to score-based guidance, ManGO's manifold learning framework enables additional conditioning on design constraints, providing more flexible control over the generation process. Figures 2a, b present ManGO's generated trajectories with minimal score and varying design constraints as conditional guidance. For instance, in Figure 2a with design constraints $x_1 \in [-5, 0]$, $x_2 \in [0, 15]$ and minimum score condition $y_{\min} = 0.398$, ManGO successfully guides a randomly initialized point (violating the constraints) to converge to the constrained minimum. Notably, ManGO exhibits accelerated convergence as noisy samples approach preferred points. This indicates its ability to exploit favorable noise points for enhanced output quality, naturally aligning with our inference-time scaling framework. On the other hand, ManGO directly transports samples from random initializations to preferred points in the joint design-score space, simultaneously generating both designs and their scores. This eliminates two key requirements of conventional approaches: (i) iterative score evaluation on noisy designs via external forward models, and (ii) gradient computation along manifold geometry.

Figures 2 and Fig. 3 demonstrate the critical role of conditional guidance on ManGO's OOG capability. Unconditional generation faithfully

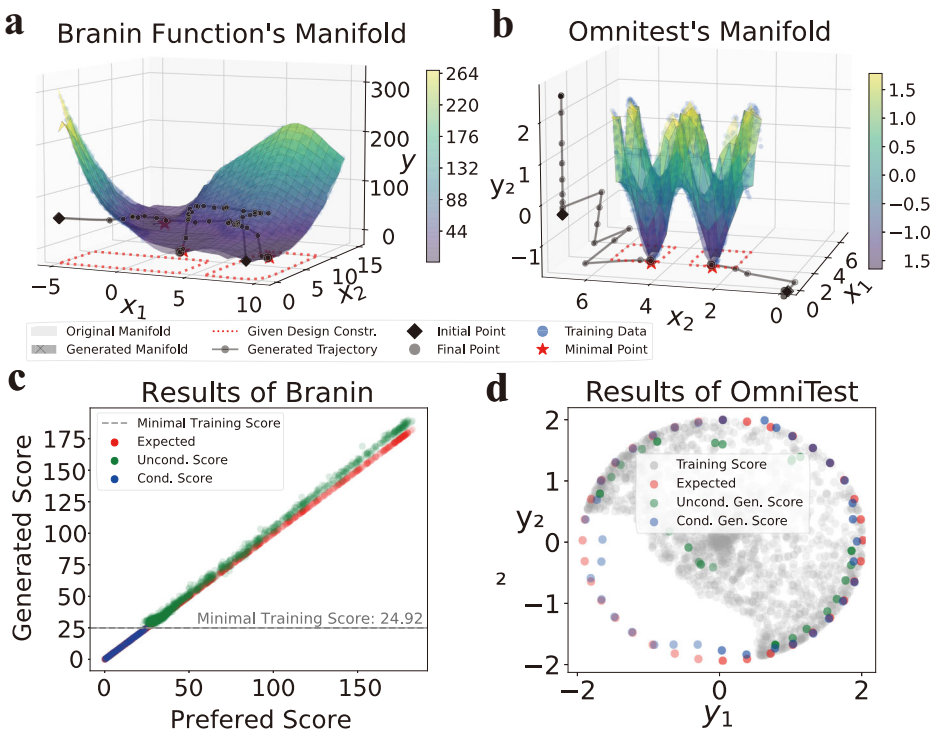

Fig. 2 | Visualization of manifold learning, trajectory generation, and generation capabilities of ManGO. Note that unconditional and conditional samples are generated via ManGO without guidance and with preferred-score guidance, respectively. **a**, **b** Manifold and trajectory comparisons for the Branin (SOO) and OmniTest (MOO) tasks. The generated manifold is constructed via ManGO's design-to-score prediction within the feasible region of designs. Close alignment between the ManGO-generated and original manifold, confirming the model's proficiency in learning complex design-score relationships. Generated trajectories visualize ManGO's score-to-design mapping under minimal score and design constraints, highlighting its capacity to perform targeted denoising toward desired regions. **c** Branin task: Unconditional samples (green) match preferred scores from the training dataset, while conditional samples (blue) extrapolate beyond the training minimum (grey dashed line). **d** OmniTest task: Conditional samples better approximate preferred scores and Pareto-dominate the training data (grey) compared to unconditional samples. These results indicate that ManGO effectively reconstructs in-distribution samples during unconditional generation—reflecting well-learned manifold structure—while enabling OOG of superior samples through conditional guidance, demonstrating robust extrapolation based on the learned manifold.

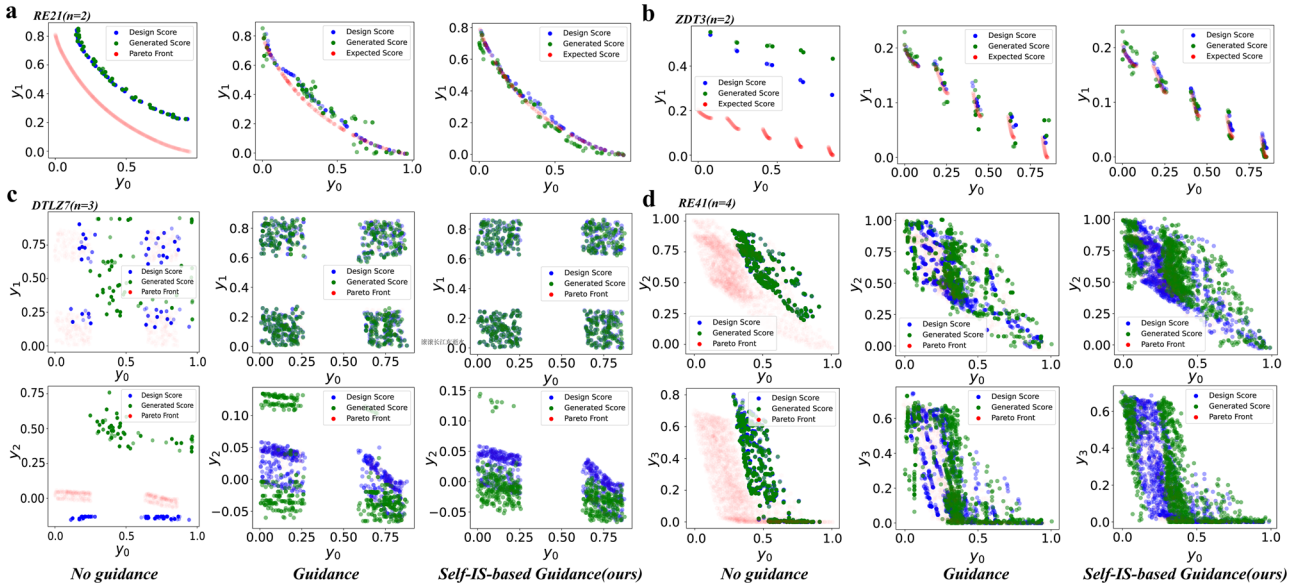

Fig. 3 | Pareto front generation under different guidance conditions. Across all subfigures of (**a**) RE21, (**b**) ZDT3, (**c**) DTLZ7, and (**d**) RE41, columns from left to right respectively show the results of no guidance, standard guidance, and the proposed self-IS-based guidance. The progressive improvement in generation quality highlights ManGO's capability in OOG under conditional guidance. Furthermore, the enhanced performance with self-IS-based guidance illustrates the ManGO's feasibility for more delicate guidance mechanisms.

reproduces in-distribution samples (matching training score ranges in Fig. 2c, d), while conditional generation produces designs that extrapolate beyond the training distribution. This key distinction reveals that although ManGO learns the complete manifold structure, explicit guidance is essential to unlock its full OOG capabilities. The consistent results across RE21, ZDT3, DTLZ7, and RE41 benchmarks (Fig. 3) robustly confirm this fundamental behavior. Specifically, ManGO without guidance exhibits conservative behavior, remaining within the training distribution. Meanwhile, when guided by Pareto front (PF) reference points, ManGO approaches the complete PF, and our self-supervised scaling guidance achieves better precision than standard guidance.

Evaluation on single-objective optimization

We employ five representative tasks from Design-Bench [6] and sample 10,000 offline design samples per task [28]: (i) *Ant Morphology* [38] (60-D parameter optimization for quadruped locomotion speed), (ii) *D'Kitty Morphology* [39] (56-D parameter optimization for movement efficiency enhancement of a quadruped robot), and (iii) *Superconductor* [37], (iv) *TF-Bind-8* [40] and (v) *TF-Bind-10* [40] (discrete DNA sequence optimization for transcription factor binding affinity with sequence lengths 8 and 10). We follow the maximization setting of Design-Bench and normalize scores based on the maximal score in the unobserved dataset [6], where higher scores indicate better performance.

As shown in Table 1, we compare ManGO with 22 baseline methods and report normalized scores of top k = 128 candidates (100th percentile). ManGO establishes a state-of-the-art performance across diverse domains (materials, robotics, bioengineering) on five datasets. ManGO with standard guidance attains a mean rank of 2.2/24 (securing the second position), while the self-supervised importance sampling (self-IS)-based variant further improves this to 1.4/24 (the first). ManGO ranks first on four tasks, including D'Kitty, Superconductor, TF-Bind-8, and TF-Bind-10, and secures second place on Ant, trailing only CMA-ES. This cross-domain advantage suggests that the effectiveness of learning the design-score manifold is general and not limited to specific problems.

The 13.6-rank leap over Mins (rank 15.0, the previous best of inverse-modeling baselines) demonstrates the superiority of the manifold-learned generation to design-space-learned methods. Meanwhile, the 2.8-rank lead over RaM (the best of forward-modeling baselines) suggests that score-conditioned diffusion can better exploit offline data than ranking-based approaches. It also outperforms the top surrogate-based methods (CMA-ES, rank 13.0) by 11.6 ranks, without the need for designing acquisition functions. On the other hand, the self-IS variant shows consistent improvements over standard ManGO: score boosts on Superconductor (+3.8%) and modest gains on Ant (+0.8%) and TF-Bind-10 (+0.9%). A deviation occurs in D'Kitty, where a slight average score reduction (−0.2%) accompanies improved peak performance (+0.3%). The marginal gains reflect that standard ManGO reaches near-optimal performance, leaving limited room for improvement.

Evaluation on multi-objective optimization

We utilize Off-MOO-Bench [14] and sample 60,000 samples per task [34]: (i) *Synthetic Functions* (an established collection of MOO evaluation tasks with $2-3$ objectives exhibiting diverse PF characteristics, such as ZDT [41] and DTLZ [42]), and (ii) *real-world engineering (RE) applications* [43] (a suite of practical design tasks with $2-4$ competing objectives, such as four-bar truss design and rocket injector design). We employ two standard evaluation metrics: (i) *Hypervolume (HV)*[44], which quantifies the dominated volume between candidate designs and nadir point (each dimension of which corresponds to the worst value of one objective), and (ii) *Inverted Generational Distance (IGD)* [45], which measures the average minimum distance between candidate designs and the ground-true PF, both metrics applied to non-dominated sorting [46] with k = 256 candidate designs (100th percentile). While generating high-quality single solutions from purely offline data remains challenging, we also report our method's performance at $k = 1$ to demonstrate competitive performance. Note that while we replace the online query in MOBO/NSGA2 with a surrogate forward model for offline adaptation, performance degradation occurs versus online operation as the surrogate model cannot perfectly emulate environment feedback.

We follow the minimization setting of Off-MOO-Bench and normalize HV (IGD) values based on the best HV (IGD) of the training dataset, where higher HV (lower IGD) indicates better performance. ManGO outperforms all baseline methods across both synthetic and real-world MOO benchmarks

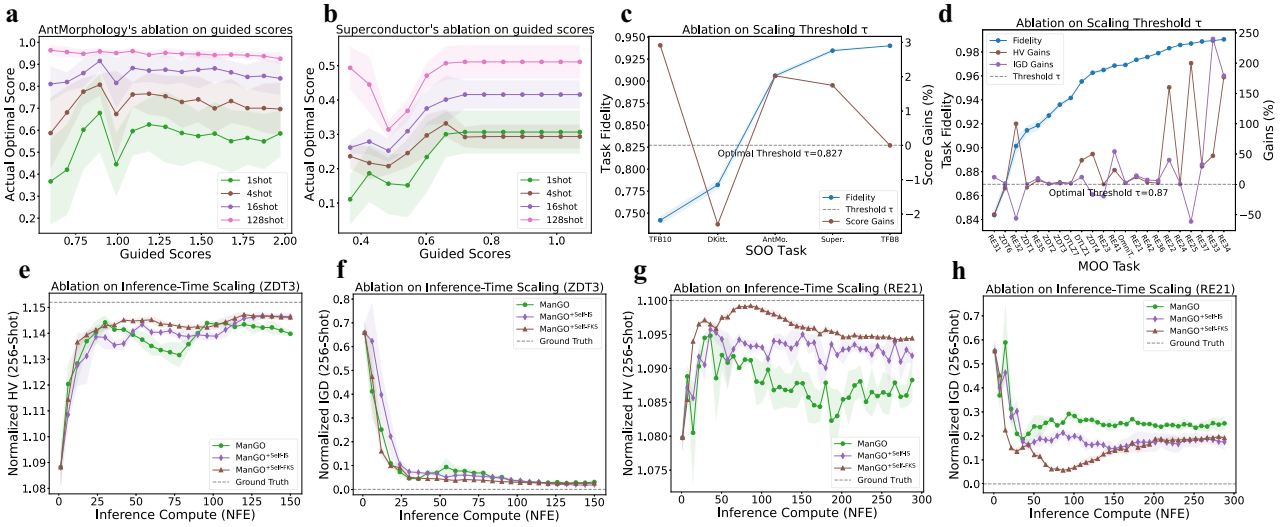

Fig. 4 | Comprehensive ablation studies validating key components of the ManGO framework. a, **b** Flexibility to guidance specification: Performance sensitivity to deviations between guided scores and true optima on Ant (**a**) and Superconductor (**b**) tasks, demonstrating ManGO's stability under suboptimal guidance conditions. **c**, **d** Fidelity-adaptive scaling: Performance gains relative to baseline fidelity thresholds in offline SOO (**c**) and MOO (**d**) tasks, with dashed lines indicating empirically optimal thresholds (τ_{opt} = 0.827 for SOO; τ_{opt} = 0.87 for MOO). **e**–**h** Inference-time scaling efficiency: HV (**e**, **g**) and IGD (**f**, **h**) versus number of function evaluations (NFE) for ZDT3 (**e**, **f**) and RE21 (**g**, **h**) task for three approaches, comparing standard denoising, self-IS scaling, and FKS scaling methods. Consistent performance improvements are achieved through adaptive noise-space exploration.

according to Table 2. Regarding synthetic tasks, the self-IS-based ManGO achieves the best mean rankings of 2.0 (HV) and 1.3 (IGD) out of 10 competing methods, while the standard guidance version follows closely with ranks of 2.7 (HV) and 1.7 (IGD), securing the top two positions. The superiority extends to RE tasks, where self-IS-based ManGO dominates with average ranks of 1.3 (HV) and 2.0 (IGD), establishing itself as the overall leader.

As presented in the upper part of Table 2, ManGO shows consistent superiority across ZDT, OmniTest, and DTLZ series. Under the most challenging OOG scenarios where preferred designs are distant from ZDT's training data, self-IS-based ManGO outperforms the best baselines by 60.1% in IGD (vs DDOM) and 3.9% in HV (vs NSGA-2). For 1-shot evaluation settings (i.e., $k = 1$), ManGO matches the performance of baseline methods requiring 256-shot evaluations. This shows that the efficient learning on the design-score manifold enables high-quality guidance generation even with minimal sampling.

As task complexity escalates with increasing objectives, the self-IS-based variant consistently achieves top performance in both HV and IGD metrics in the lower part of Table 2. This confirms its OOG capability in high-dimensional objective spaces. Compared to synthetic tasks, learning on manifolds poses greater challenges for RE tasks. Consequently, this diminishes ManGO's performance advantage in 1-shot and standard guidance modes. However, self-IS guidance effectively offsets this by exploring more noise points, and its performance gains become more pronounced as task complexity increases.

Ablation study on robustness to preferred scores

Although we use the maximum unobserved score as a guided score (i.e., $\boldsymbol{y}_p = 1.0$) in Table 1, practical scenarios may lack precise knowledge of optimal scores. We evaluate ManGO's robustness to suboptimal guided scores by analyzing the best score of generated candidate designs across varying shot numbers. Fig. 4a, b reveal two key insights: (i) Increasing the shot number enhances robustness, with stable optimal designs emerging when guided scores exceed 0.7 (Ant) or 5.5 (Superconductor) at 128 shots. (ii) Decreasing the shot number to 1, peak performance occurs near but not exactly at $\boldsymbol{y}_p$. This demonstrates that the generation diversity of diffusion models provides practical robustness for ManGO when $\boldsymbol{y}_p$ is unknown.

Ablation study on optimal fidelity threshold

We quantitatively evaluate the design-to-score prediction accuracy through the fidelity metric (Eq. (10)), which measures the distance between the ground-truth scores and the generated scores of unconditional samples. Figures 4c, d show that self-IS-based scaling achieves performance gains on the majority of SOO (optimal fidelity threshold $\tau_{opt} = 0.827$) and MOO tasks ($\tau_{opt} = 0.87$). Furthermore, MOO tasks exhibit higher fidelity than SOO tasks due to SOO's higher-dimensional design spaces. It is more likely to increase performance gains with increasing fidelity values because diffusion models with higher fidelity generate more accurate self-reward signals during inference, enabling more effective noise-space exploration.

Ablation study on inference-time scaling

We evaluate computation-performance tradeoffs by controlling NFE on ZDT3 in Figures 4e, f and RE21 in Fig. 4g, h, comparing three approaches: (1) standard guidance with more denoising steps, (2) self-IS-based scaling, and (3) Feynman-Kac-steering (FKS)-based scaling. For ZDT3, standard guidance achieves competitive performance at NFE = 31, demonstrating ManGO's sample efficiency. However, performance degrades at intermediate NFE before recovering, revealing instability in simple step extension. In contrast, scaling-based methods show monotonic improvement with increasing NFE, where FKS scaling temporarily outperforms IS scaling during mid-range NFEs before converging at NFE = 150. The RE21 task exhibits different characteristics: all methods display bell-shaped performance curves, peaking at NFE = 78 (FKS), NFE = 36 (IS), and NFE = 36 (standard). Scaling-based methods attain higher peak performance over standard guidance while maintaining a sustained advantage after NFE = 36. These results indicate that scaling methods yield superior computational performance compared to simple step extension.

Discussion

This work introduces ManGO, a framework that fundamentally rethinks offline optimization by learning the underlying design-score manifold through diffusion models. Unlike existing approaches that operate in isolated design or score spaces, ManGO's bidirectional modeling unifies forward prediction and backward generation while overcoming OOG

challenges. The derivative-free guidance mechanism eliminates reliance on error-prone forward models, while the adaptive inference-time scaling dynamically optimizes denoising paths. Extensive validation across synthetic tasks and real-world applications (robot control, material design, DNA optimization) demonstrates ManGO's consistent superiority among 24 offline-SOO and 10 offline-MOO methods, establishing a new paradigm for data-driven design generation for complex system optimization.

We envision three critical future directions. First, extending ManGO to high-dimensional and discrete design spaces (e.g., 3D molecular structures) requires developing techniques for learning the latent-based manifold via encoding designs as latents to maintain computational efficiency. Recent work on latent diffusion models suggests potential pathways for improvement. Second, integrating physics-informed constraints beyond current design-clipping guidance could enhance physical plausibility in domains like metamaterial design, where conservation laws must be preserved. Preliminary experiments with physics-informed neural networks show promising results. Third, developing distributed ManGO variants would enable collaborative optimization across institutions while preserving data privacy, particularly valuable for pharmaceutical development where proprietary molecule datasets exist in isolation.

Several limitations warrant discussion. ManGO's current implementation assumes quasi-static system environments, while gradually evolving scenarios would require incremental manifold adaptation mechanisms. While our adaptive scaling provides partial mitigation, improvement for non-stationary distributions remains an open challenge. Meanwhile, although ManGO demonstrates strong robustness to preferred scores in identifying optimal designs, it lacks an iterative refinement mechanism to further improve designs post-generation. Recent advances in post-training adaptation of diffusion models, such as controllable fine-tuning and editing, suggest promising pathways to augment ManGO with such capability.

Methods

In this section, we delve into the core components of our approach: (i) training the diffusion model to learn the design-score manifold and (ii) bidirectional guidance generation with the preferred condition. Finally, we present the training and inference settings of our proposed approach.

Unconditional training of diffusion model on score-augmented dataset

To better explore unknown design-score pairs, our method aims to capture the prior probability distribution of the design-score manifold using a diffusion model. This is a critical step to improve OOG limitation of backward methods to solve offline optimization problems. To achieve this, we train an unconditional diffusion model based on a variance-preserving (VP) SDE[16]. The model is trained on a joint design-score dataset $\hat{\mathcal{D}} = \{\hat{\boldsymbol{x}}_i : \hat{\boldsymbol{x}}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^{(d+m)}\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ denotes the design and $\boldsymbol{y}_i$ represents its corresponding score vector. Specifically, $\hat{\mathcal{D}}$ is constructed by augmenting the original design dataset $\mathcal{D}_x = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^{N}$ with score information.

Unlike classifier-free diffusion models[47], which perform conditional training by incorporating score information as a condition with random dropout during training, our method directly learns the joint distribution of designs and scores. This eliminates the need to learn the design distribution under varying conditions, instead focusing on capturing the underlying structure of the design-score manifold. The diffusion process of our manifold-trained model is denoted by the following SDE:

$$d\hat{\boldsymbol{x}} = -\frac{1}{2}\beta_t \hat{\boldsymbol{x}}\, dt + \sqrt{\beta_t}\, d\boldsymbol{w}, \tag{1}$$

where $\beta_t = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$ with $t \in [0, 1]$. The denoising process is represented by the following reverse-time SDE:

$$d\hat{\boldsymbol{x}} = -\beta_t \left[\frac{\hat{\boldsymbol{x}}}{2} + \nabla_{\hat{\boldsymbol{x}}} \log p_t(\hat{\boldsymbol{x}})\right] dt + \sqrt{\beta_t}\, d\tilde{\boldsymbol{w}}, \tag{2}$$

where $d\tilde{\boldsymbol{w}}$ denotes the reverse-time Wiener process. Using this process, the pre-trained diffusion model can transform a prior noise point $\hat{\boldsymbol{x}}_T \sim p_T(\hat{\boldsymbol{x}})$ into a design-score-joint point $\hat{\boldsymbol{x}}_0 = (\boldsymbol{x}_0, \boldsymbol{y}_0)$ on the design-score manifold.

Learning to reverse superior designs via score-based loss reweighting

When uniformly sampling data points from the augmented dataset $\hat{\mathcal{D}}$, the diffusion model learns the entire manifold. However, this does not fully align with the goal of offline optimization, which aims to find superior or optimal designs. We introduce a score-based reweighting mechanism into the loss function to steer the model toward regions of the manifold associated with lower scores (indicating better designs). This mechanism assigns higher weights to samples with lower scores, thereby encouraging the model to prioritize the generation of superior designs and reducing the learning complexity by focusing on the most promising regions of the manifold rather than learning the entire manifold.

Specifically, the weight vector for offline SOO (or MOO) tasks is obtained by applying max-min normalization to the sample scores (or the sample frontier index) based on non-dominated sorting[46] across the entire dataset. The reweighting mechanism allows the diffusion model to focus on the most promising region of the manifold, which is expressed as:

$$\boldsymbol{w}(\hat{\boldsymbol{x}} := (\boldsymbol{x}, \boldsymbol{y})) = \begin{cases} \frac{y_{\max} - y}{y_{\max} - y_{\min}}, & m = 1, \\ \frac{l_{\text{all}} - l_{\text{NDS}(y)}}{l_{\text{all}} - 1}, & m > 1, \end{cases} \tag{3}$$

where $y_{\min}$ and $y_{\max}$ denote the minimum and maximum scores, respectively, and l_{all} and $l_{\text{NDS}(y)}$ represent the total number of frontier layers and the frontier index of $\hat{\boldsymbol{x}}$, respectively. Equation (3) indicates that samples with lower scores (or those located on a more advanced frontier) are assigned higher weights in offline SOO (or MOO) tasks. We note that the normalization-based implementation serves as a foundational step, and future work may explore more efficient reweighting mechanisms.

The score function $\nabla_{\hat{\boldsymbol{x}}} \log p_t(\hat{\boldsymbol{x}})$ in Equation (2) is approximated by a time-dependent neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t)$ with parameters $\boldsymbol{\theta}$ based on VP-SDE[16]. The network is optimized by minimizing the following loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{t}\left[\lambda(t) \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}_0}\left[\boldsymbol{w}(\hat{\boldsymbol{x}}_0) \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}_t|\hat{\boldsymbol{x}}_0}\left[\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t) - \nabla_{\hat{\boldsymbol{x}}} \log p_t(\hat{\boldsymbol{x}}_t|\hat{\boldsymbol{x}}_0)\right\|_2^2\right]\right]\right], \tag{4}$$

where $\lambda(t)$ is a positive weighting function dependent on t, and $\nabla_{\hat{\boldsymbol{x}}} \log p_t(\hat{\boldsymbol{x}}_t|\hat{\boldsymbol{x}}_0)$ can be obtained by the diffusion process.

Bidirectional generation on design-score samples via derivative-free guidance

Diffusion models generate data by iteratively refining random noise into preferred samples through a guided denoising process. The pre-trained diffusion model directly generates design-score samples, where each design is paired with a score predicted by the model itself, eliminating the need for an additional surrogate score model. This capability allows us to jointly guide the model on both the design and its associated score, enabling the generation of samples that align with our preferences.

In terms of score-to-design generation, the proposed method enables guided generation of preferred design-score samples through two mechanisms: (1) leveraging preferred scores $\boldsymbol{y}_{\text{P}}$ to guide the score

component of the generated samples, ensuring alignment with desired performance; and (2) specifying a design constraint range $[\boldsymbol{x}_{c_{min}}, \boldsymbol{x}_{c_{max}}]$ to guide the design component, ensuring design feasibility. This dual-guidance strategy generates samples that satisfy both design and score constraints without requiring an additional surrogate score model.

Concretely, during the reverse process, the guidance on the score component of the generated sample $\hat{\boldsymbol{x}}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ is formulated as the gradient of the MSE between $\boldsymbol{y}_p$ and the current score $\boldsymbol{y}_t$, i.e., $\nabla_{\boldsymbol{y}_t} \| \boldsymbol{y}_p - \boldsymbol{y}_t \|_2^2$. Similarly, the guidance on the design component is formulated as the gradient of the MSE from the current design $\boldsymbol{x}_t$ to the constraint range $[\boldsymbol{x}_{c_{min}}, \boldsymbol{x}_{c_{max}}]$, i.e., $\nabla_{\boldsymbol{x}_t} \| \boldsymbol{x}_t - \text{clip}(\boldsymbol{x}_t, \boldsymbol{x}_{c_{min}}, \boldsymbol{x}_{c_{max}}) \|_2^2$, where $\text{clip}(\boldsymbol{x}_t, \boldsymbol{x}_{c_{min}}, \boldsymbol{x}_{c_{max}})$ ensures that $\boldsymbol{x}_t$ is projected onto the constraint range.

Since both MSE gradients can be computed analytically, our method effectively implements a derivative-free guidance scheme that operates without relying on differentiable models, avoiding the computational overhead of backpropagating (e.g., classifier guidance). Using the Euler-Maruyama method [48], the reverse process can be expressed as:

$$\hat{\boldsymbol{x}}_{t-\Delta t} = \hat{\boldsymbol{x}}_t + \beta_t \Big[\tfrac{\hat{\boldsymbol{x}}_t}{2} + \boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t) + \alpha_x \Big(\hat{\boldsymbol{x}}_t - \text{clip}(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_{c_{min}}, \hat{\boldsymbol{x}}_{c_{max}}) \Big) + \alpha_y (\hat{\boldsymbol{y}}_p - \hat{\boldsymbol{y}}_t) \Big] \Delta t + \sqrt{\beta_t \Delta t}\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{5}$$

where α_x and α_y control the guidance scale on the design and score components, respectively. Here, $\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_{c_{min}}, \hat{\boldsymbol{x}}_{c_{max}}, \hat{\boldsymbol{y}}_p$, and $\hat{\boldsymbol{y}}_t$ are padded versions of their original vectors with a zero vector $\boldsymbol{0}_d$ to match the dimensionality of $\hat{\boldsymbol{x}}_t$, like $\hat{\boldsymbol{y}}_p = (\boldsymbol{0}_d, \boldsymbol{y}_p)$.

Similarly, the design-to-score prediction uses the zero-vector-augmented preferred design $\hat{\boldsymbol{x}}_p = (\boldsymbol{x}_p, \boldsymbol{0}_m)$ as guidance:

$$\hat{\boldsymbol{x}}_{t-\Delta t} = \hat{\boldsymbol{x}}_t + \beta_t \left[\frac{\hat{\boldsymbol{x}}_t}{2} + \boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t) + \alpha_x (\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_t) \right] \Delta t + \sqrt{\beta_t \Delta t}\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{6}$$

where the estimated score of the preferred design $\boldsymbol{x}_p$ is $\boldsymbol{y}_0$ from $\hat{\boldsymbol{x}}_0 = (\boldsymbol{x}_0, \boldsymbol{y}_0)$.

Achieving inference-time scaling via self-supervised reward

Diffusion models have been shown to exhibit *inference-time scaling* behavior [49–51]: their generation quality can be improved by allocating additional computational budgets during inference, even without fine-tuning. This stems from a mechanism of noise space exploration, where the pre-trained model optimizes the generation trajectory by adaptively selecting high-reward noise points based on reward feedback. The reward feedback can be formulated through posterior mean approximation as:

$$\mathcal{R}(\boldsymbol{x}_t) := \mathbb{E}_{\boldsymbol{x}_0 \sim p_t^{\text{pre}}(\boldsymbol{x}_t)}[r(\boldsymbol{x}_0)|\boldsymbol{x}_t] = r(\mathbb{E}_{\boldsymbol{x}_0 \sim p_t^{\text{pre}}}[\boldsymbol{x}_0|\boldsymbol{x}_t]) = r(\boldsymbol{x}_{0|t}), \tag{7}$$

where $\mathcal{R}(\boldsymbol{x}_t)$ evaluates the expected reward of samples generated from the noise point $\boldsymbol{x}_t$, p_t^{pre} represents the approximated distribution of the pre-trained model at t, $r(\cdot)$ denotes the reward model evaluated at $t = 0$, and $\boldsymbol{x}_{0|t} = \mathbb{E}_{\boldsymbol{x}_0 \sim p_t^{\text{pre}}}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ denotes that the pre-trained model denoises the noise point from t to 0. Based on this mechanism, noise points with high rewards are retained for further refinement, while those with low rewards are discarded.

The dual-output capability (design-score generation) of our pre-trained model enables a self-supervised inference-time scaling scheme. For any intermediate noise point $\hat{\boldsymbol{x}}_t$, we estimate the denoised state $\hat{\boldsymbol{x}}_{0|t}$ via Tweedie's Formula [36] as:

$$\hat{\boldsymbol{x}}_{0|t} = (\boldsymbol{x}_{0|t}, \boldsymbol{y}_{0|t}) = \left(\frac{\boldsymbol{x}_t + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{s}_{\boldsymbol{\theta}}^{(x)}(\hat{\boldsymbol{x}}_t, t)}{\sqrt{\bar{\alpha}_t}}, \frac{\boldsymbol{y}_t + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{s}_{\boldsymbol{\theta}}^{(y)}(\hat{\boldsymbol{x}}_t, t)}{\sqrt{\bar{\alpha}_t}} \right), \tag{8}$$

where $\bar{\alpha}_t = \exp[-(\beta_{\max} - \beta_{\min})\frac{t^2}{2} - \beta_{\min} t]$, and $\boldsymbol{s}_{\boldsymbol{\theta}}^{(x)}, \boldsymbol{s}_{\boldsymbol{\theta}}^{(y)}$ denote the design and score components of the model output $\boldsymbol{s}_{\boldsymbol{\theta}}$, respectively. The estimated score $\boldsymbol{y}_{0|t}$ provides self-supervised reward feedback by measuring its deviation from the preferred score:

$$\mathcal{R}(\hat{\boldsymbol{x}}_t) = r(\hat{\boldsymbol{x}}_{0|t}) = \| \boldsymbol{y}_p - \boldsymbol{y}_{0|t} \|_2. \tag{9}$$

This eliminates the need for external reward models, enabling the model to autonomously reject low-reward noise points and refine its generation process through self-supervised feedback.

As the effectiveness of inference-time scaling depends on the reward quality [49], the effectiveness of our self-supervised reward depends on the fidelity of the pre-trained diffusion model, denoted by $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}})$. We propose to quantitatively assess $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}})$ through the unconditional generation of $\boldsymbol{s}_{\boldsymbol{\theta}}$. Specifically, we first generate a set of unconditional samples $\{\boldsymbol{x}_{uc} = (\boldsymbol{x}_{uc}, \boldsymbol{y}_{uc})\}$ by Equation (5) without guidance. We then filter the samples that are better than training data (i.e., $y_{uc}^{(i)} < y_{train}^{(best)}$ for SOO and $\boldsymbol{y}_{uc}^{(i)} \prec \boldsymbol{y}_{train}^{(best)}$ for MOO) and get the filtered set $\mathcal{D}_{uc} = \{\hat{\boldsymbol{x}}_{uc}^{(i)}\}_{i=1}^{M}$. The fidelity metric $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}})$ is subsequently computed as:

$$\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}}) = \exp\left(-\frac{1}{M} \sum_{i=1}^{M} \| \boldsymbol{y}_{uc}^{(i)} - \boldsymbol{y}_{train}^{(j^*)} \| \right), \tag{10}$$

where $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}}) \in (0, 1]$, $j_i^* = \arg\min \| \boldsymbol{x}_{uc}^{(i)} - \boldsymbol{x}_{train}^{(j)} \|_2$ denotes the index of the nearest training data to $\boldsymbol{x}_{uc}^{(i)}$ in $\mathcal{D} = \{\hat{\boldsymbol{x}}_{train}^{(j)} = (\boldsymbol{x}_{train}^{(j)}, \boldsymbol{y}_{train}^{(j)})\}_{j=1}^{N}$. A higher value of $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}})$ indicates a stronger alignment between the generated and ground-truth scores. As shown in Figure 4, this metric exhibits a strong correlation with inference-time scaling performance based on the self-supervised reward mechanism. Based on this observation, we implement a conditional logic with $\mathcal{F}(\boldsymbol{s}_{\boldsymbol{\theta}}) > \tau$ to activate the inference-time scaling scheme during guidance generation. In this way, we establish a general inference-time scaling framework for offline optimization by substituting external reward models in existing schemes with self-supervised rewards.

This framework is compatible with various methods, including the IS-based scaling [50] and the FKS-based scaling [51] methods. For example, we illustrate the complete process of ManGO based on IS with the self-supervised reward in Algorithm 1. Self-IS-based ManGO is designed to explore the denoising paths with higher rewards. Specifically, the algorithm first duplicates each noise point $\hat{\boldsymbol{x}}_t^{(i)}$ J times. Each copy then undergoes an independent denoising step as defined in Eq. (5), where the inherent randomness in the reverse process—introduced through the Wiener noise term. This causes the denoising paths to diverge and results in divergent candidates $\{\hat{\boldsymbol{x}}_{t-1}^{(i,j)}\}_{j=1}^{J}$ at the next timestep. For each candidate, ManGO estimates its corresponding denoised state to compute self-supervised rewards $\mathcal{R}(\hat{\boldsymbol{x}}_{t-1}^{(i,j)})$ Eq.s (8) and (9). These rewards drive an importance sampling mechanism that prioritizes higher-reward paths, effectively steering the generative process toward more promising regions of the design-score manifold. This framework leverages the innate stochasticity of diffusion models to explore the noise space dynamically, eliminating the need for explicit external perturbations or reward models. Our experiments validate the integration of the IS and FKS approaches with self-supervised rewards.

Algorithm 1. Self-IS-based ManGO

Input: Offline dataset $\mathscr{D}$, scaling threshold τ, preferred scores $\boldsymbol{y}_{\mathrm{p}}$, preferred design constraints $[\boldsymbol{x}_{\mathrm{c}_{\min}}, \boldsymbol{x}_{\mathrm{c}_{\max}}]$, conditional sample number K, unconditional sample number M, duplication size J, importance ratio α_{I}
Output: K candidate optimal designs.
/* *Phase 1: Training* */
Augment $\mathscr{D}$ and get $\hat{\mathscr{D}} \leftarrow \{\hat{\boldsymbol{x}}_i : \hat{\boldsymbol{x}}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$
Calculate the weight vector $\boldsymbol{w}$ with Eq.(3)
Training $\boldsymbol{s}_{\boldsymbol{\theta}}$ with Eq.(4) on $\hat{\mathscr{D}}$
/* *Phase 2: Guided generation* */
Unconditional generation and get $\mathscr{D}_{\mathrm{uc}} = \{\hat{\boldsymbol{x}}_{\mathrm{uc}}^{(i)}\}_{i=1}^{M}$
Compute fidelity $\mathscr{F}(\boldsymbol{s}_{\boldsymbol{\theta}})$ with Eq.(10) on $\mathscr{D}_{\mathrm{uc}}$
Sample random noise $\boldsymbol{X} \leftarrow \{\hat{\boldsymbol{x}}_T^{(i)} \sim \mathscr{N}(0, \boldsymbol{I})\}_{i=1}^{K}$
for $t = T - 1$ **to** 0 **do**
  if $\mathscr{F}(\boldsymbol{s}_{\boldsymbol{\theta}}) > \tau$ **then**
    Duplicate J times on $\hat{\boldsymbol{x}}_t^{(i)}$ and get $\boldsymbol{X}_i = \{\hat{\boldsymbol{x}}_t^{(i,j)}\}_{j=1}^{J}$ for $i \in [K]$
    Guide to reverse $\boldsymbol{X}_i \leftarrow \{\hat{\boldsymbol{x}}_t^{i,j}\}_{j=1}^{J}$ with Eq.(5) on $\boldsymbol{y}_{\mathrm{p}}$ and $[\boldsymbol{x}_{\mathrm{c}_{\min}}, \boldsymbol{x}_{\mathrm{c}_{\max}}]$ for $i \in [K]$
    Estimate the denoised state $\hat{\boldsymbol{x}}_{0|t}^{(i,j)}$ with Eq.(8) for $i, j \in [K], [J]$
    Compute self-supervised reward $\mathscr{R}(\hat{\boldsymbol{x}}_t^{(i,j)})$ with Eq.(9) for $i, j \in [K], [J]$
    Compute importance $w_t^{(i,j)} = \exp(\mathscr{R}(\hat{\boldsymbol{x}}_t^{(i,j)})/\alpha_{\mathrm{I}})$ for $i, j \in [K], [J]$
    Resample $j_i^* = \mathrm{Cat}\left(\{w_t^{(i,j)} / \sum_{j=1}^{J} w_t^{(i,j)}\}_{j=1}^{J}\right)$ for $i \in [K]$
    Update $\hat{\boldsymbol{x}}_t^{(i)} \leftarrow \hat{\boldsymbol{x}}_t^{(i,j_i^*)}$ for $i \in [K]$ and get $\boldsymbol{X} \leftarrow \{\hat{\boldsymbol{x}}_t^{(i)}\}_{i=1}^{K}$
  else
    Guide to reverse $\boldsymbol{X} \leftarrow \{\hat{\boldsymbol{x}}_t^{(i)}\}_{i=1}^{K}$ with Eq.(5) on $\boldsymbol{y}_{\mathrm{p}}$ and $[\boldsymbol{x}_{\mathrm{c}_{\min}}, \boldsymbol{x}_{\mathrm{c}_{\max}}]$
  end if
end for

Training and inference settings

Our model architecture processes three input components: design, scores, and timestep. The timestep undergoes standard cosine embedding, while design and score are independently projected via fully connected layers, both to 128-D features. These features interact bidirectionally through cross-attention layers, with outputs fused via two multi-layer perceptron (MLP) layers (128 hidden units, Swish activation). The fused features are then combined with time embeddings and processed through a three-layer MLP (2048 hidden units, Swish) for reconstruction. We employ AdamW optimizer with a learning rate (LR) of 5×10^{-5}, a weight decay coefficient of 1×10^{-4}, and a one-cycle LR scheduler with cosine annealing. Training converges in 800 epochs for SOO and 400 epochs for MOO, maintaining original baseline configurations. The diffusion process uses $\beta_{\max}, \beta_{\min} = 1 \times 10^{-4}, 5 \times 10^{-2}$ for SOO and $1 \times 10^{-4}, 5 \times 10^{-3}$ for MOO.

During inference, we configure 200 denoising steps for all MOO tasks, Ant and DKittyMorphology of Design-bench tasks, 5 for Superconduct tasks, and 250 for TF-Bind-8 and TF-Bind-10. For the guidance scaler, we set $\alpha_x = 1, \alpha_y = 1$ for design-constraint trajectory generation in Figure 2, and $\alpha_x = 0$, $\alpha_y = 1$ for benchmark evaluation. For the sake of comparison, we disable the fidelity-adaptive activation of the inference-time scaling, denoted as (10), in the self-supervised importance sampling-based ManGO. The scaling methods are activated every five denoising steps, where the IS-based scaling uses beam search with duplication size $J = 16$, and the FKS-based scaling uses accumulated maximal rewards. Complete implementation details are provided in the supplementary material.

Data availability

The datasets used in this paper are publicly available. The Design-Bench benchmark datasets are available at https://huggingface.co/datasets/beckhamc/design_bench_data. The Off-MOO-Bench datasets are available at https://github.com/lamda-bbo/offline-moo. The usage of these datasets in this work is permitted under their licenses.

Code availability

Codes for this work are available at https://github.com/TailinZhou/ManGO_SOOfor offline SOO tasks and https://github.com/TailinZhou/ManGO_MOOfor offline MOO tasks. All experiments and implementation details are thoroughly described in the Experiments section, Methods section, and Supplementary Information.

Received: 26 June 2025; Accepted: 7 November 2025;
Published online: 07 January 2026

References

1. Jumper, J. et al. Highly accurate protein structure prediction with Alphafold. *Nature* **596**, 583–589 (2021).
2. Pogue, E. A. et al. Closed-loop superconducting materials discovery. *npj Computational Mater.* **9**, 181 (2023).
3. Jha, D. et al. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nat. Commun.* **10**, 5316 (2019).
4. Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M. & Ahsan, M. J. Machine learning in drug discovery: A review. *Artif. Intell. Rev.* **55**, 1947–1999 (2022).
5. Stahl, B. C., Ogoh, G., Schumann, G. & Walter, H. Rethinking ethics in interdisciplinary and big data-driven neuroscience projects. *Nat. Ment. Health* **2**, 1128–1130 (2024).
6. Trabucco, B., Geng, X., Kumar, A. & Levine, S. Design-Bench: Benchmarks for data-driven offline model-based optimization. In

Proceedings of the 39th International Conference on Machine Learning (ICML), 21658–21676 (Baltimore, MD, 2022).
7. Wang, J. et al. Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence* (2021).
8. Stanton, S. et al. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine Learning (ICML)* (Baltimore, Maryland, USA, 2022).
9. Kumar, A., Yazdanbakhsh, A., Hashemi, M., Swersky, K. & Levine, S. Data-driven offline optimization for architecting hardware accelerators. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)* (Virtual, 2022).
10. Hansen, N. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation - Advances in the Estimation of Distribution Algorithms*, vol. 192 of *Studies in Fuzziness and Soft Computing*, 75–102 (Springer, 2006).
11. Chen, C. S., Zhang, Y., Fu, J., Liu, X. S. & Coates, M. Bidirectional learning for offline infinite-width model-based optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 29454–29467 (New Orleans, LA, 2022).
12. Krishnamoorthy, S., Mashkaria, S. M. & Grover, A. Diffusion models for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 17842–17857 (Honolulu, HI, 2023).
13. Yang, J. et al. Locating what you need: Towards adapting diffusion models to OOD concepts in-the-wild. In *Advances in Neural Information Processing Systems 38 (NeurIPS)* (Vancouver, Canada, 2024).
14. Xue, K., Tan, R.-X., Huang, X. & Qian, C. Offline multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 55595–55624 (Vienna, Austria, 2024).
15. Miettinen, K.*Nonlinear Multiobjective Optimization* (Kluwer, 1998).
16. Song, Y. et al. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)* (Virtual, 2021).
17. Garnett, R.*Bayesian Optimization* (Cambridge University Press, 2023).
18. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **104**, 148–175 (2016).
19. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).
20. Trabucco, B., Kumar, A., Geng, X. & Levine, S. Conservative objective models for effective offline model-based optimization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 10358–10368 (Virtual, 2021).
21. Yu, S., Ahn, S., Song, L. & Shin, J. RoMA: Robust model adaptation for offline model-based optimization. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 4619–4631 (Virtual, 2021).
22. Qi, H., Su, Y., Kumar, A. & Levine, S. Data-driven offline decision-making via invariant representation learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 13226–13237 (New Orleans, LA, 2022).
23. Yuan, Y., Chen, C. S., Liu, Z., Neiswanger, W. & Liu, X. S. Importance-aware co-teaching for offline model-based optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 55718–55733 (New Orleans, LA, 2023).
24. Chen, C. S., Beckham, C., Liu, Z., Liu, X. S. & Pal, C. Parallel-mentoring for offline model-based optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 76619–76636 (New Orleans, LA, 2023).
25. Chemingui, Y., Deshwal, A., Hoang, T. N. & Doppa, J. R. Offline model-based optimization via policy-guided gradient search. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 11230–11239 (Vancouver, Canada, 2024).
26. Grudzien, K., Uehara, M., Levine, S. & Abbeel, P. Functional graphical models: Structure enables offline data-driven optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2449–2457 (Valencia, Spain, 2024).
27. Hoang, M., Fadhel, A., Deshwal, A., Doppa, J. & Hoang, T. N. Learning surrogates for offline black-box optimization via gradient matching. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 18374–18393 (Vienna, Austria, 2024).
28. Tan, R.-X. et al. Offline Model-Based Optimization by Learning to Rank. In *Proceedings of the Thirteenth International Conference on Learning Representations* (Singapore, 2025).
29. Brookes, D., Park, H. & Listgarten, J. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 773–782 (Long Beach, CA, 2019).
30. Kumar, A. & Levine, S. Model inversion networks for model-based optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 5126–5137 (Virtual, 2020).
31. Krishnamoorthy, S., Mashkaria, S. M. & Grover, A. Generative pretraining for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 24173–24197 (Honolulu, HI, 2023).
32. Yun, T., Yun, S., Lee, J. & Park, J. Guided trajectory generation with diffusion models for offline model-based optimization. In *Advances in Neural Information Processing Systems 38 (NeurIPS)* (Vancouver, Canada, 2024).
33. Daulton, S., Balandat, M. & Bakshy, E. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems* (2021).
34. Yuan, Y., Chen, C., Pal, C. & Liu, X. ParetoFlow: Guided Flows in Multi-Objective Optimization. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)* (Singapore, 2025).
35. Meilă, M. & Zhang, H. Manifold learning: What, how, and why. *Annu. Rev. Stat. Its Application* **11**, 393–417 (2024).
36. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 6840–6851 (Virtual, 2020).
37. Hamidieh, K. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Mater. Sci.* **154**, 346–354 (2018).
38. Brockman, G. et al. OpenAI Gym. *arXiv:1606.01540* (2016).
39. Ahn, M. et al. ROBEL: Robotics benchmarks for learning with low-cost robots. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, 1300–1313 (Virtual, 2020).
40. Barrera, L. A. et al. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science* **351**, 1450–1454 (2016).
41. Zitzler, E., Deb, K. & Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolut. Comput.* **8**, 173–195 (2000).
42. Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, 825–830 (2002).
43. Tanabe, R. & Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Appl. Soft Comput.* **89**, 106078 (2020).
44. Zitzler, E. & Thiele, L. Multiobjective optimization using evolutionary algorithms: A comparative case study. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN)*, 292–304 (Amsterdam, The Netherlands, 1998).

45. Bosman, P. A. N. & Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evolut. Comput.* **7**, 174–188 (2003).
46. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transact. Evol. Comput.* **6**, 182−197 (2002).
47. Ho, J. & Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications* (2021). https://openreview.net/forum?id=qw8AKxfYbI.
48. Kloeden, P. E., Platen, E., Kloeden, P. E. & Platen, E.*Stochastic differential equations* (Springer, 1992).
49. Ma, N. et al. Inference-time scaling for diffusion models beyond scaling denoising steps (2025).
50. Uehara, M. et al. Reward-guided controlled generation for inference-time alignment in diffusion models: Tutorial and review (2025).
51. Singhal, R. et al. A general framework for inference-time scaling and steering of diffusion models (2025).
52. Deb, K. & Tiwari, S. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *Eur. J. Operational Res.* **185**, 1062–1087 (2008).

Acknowledgements

The ManGO's main idea and training part were done during Tailin's internship at Huawei Noah's Ark Lab, and the inference part was done at HKUST. This work was supported in part by the Hong Kong Research Grants Council under the Areas of Excellence scheme grant AoE$/$E-601$/$22-R, in part by NSFC$/$RGC Collaborative Research Scheme grant CRS$_$HKUST603$/$22, in part by Guangzhou Municipal Science and Technology Project under Grant 2023A03J0011, Guangdong Provincial Key Laboratory of Integrated Communications, Sensing and Computation for Ubiquitous Internet of Things, and National Foreign Expert Project, Project Number G2022030026L. Some icons in Figure 1 are from the website https://www.cleanpng.com/.

Author contributions

T.Z. conceived the idea of this work, implemented the models for experiments, and analyzed the results. Z.L.C. and W.L. contributed to the core idea discussion and result interpretation. J.Z. provided critical feedback on the design of inference-time scaling methodology and coordinated the research efforts. All authors participated in manuscript writing and refinement.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44387-025-00055-1.

Correspondence and requests for materials should be addressed to Jun Zhang.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025