LEARNING WHERE TO GO: SPECTRALLY ALIGNED SURROGATES AND SEARCH FOR OFFLINE MODEL-BASED OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In offline model-based optimization (MBO), the quality of a learned surrogate is determined not only by how well it captures preferences among observed designs, but also by whether its geometry provides reliable directions for optimization beyond the training distribution. Standard learning objectives can preserve useful predictive structure while still admitting high-frequency variations that produce unstable or misleading gradients during out-of-distribution search. We introduce SASS (Spectrally Aligned Surrogates and Search), a framework that aligns surrogate learning and search through a shared spectral inductive bias. SASS trains the surrogate to preserve its induced preferences under high-frequency perturbations, encouraging reliance on stable low-frequency structure while suppressing variations that distort the optimization landscape. At search time, gradient updates are projected onto the same low-frequency subspace, so that surrogate learning and optimization are governed by a common notion of stable structure. Across five Design-Bench tasks, SASS improves over both ranking baselines in mean normalized score on every task. Further analyses assess how spectral projection preserves objective values and rankings, and identify conditions under which the low-frequency assumption breaks down.

1 INTRODUCTION

Design optimization from offline data is a fundamental problem in many scientific and industrial domains, including molecule design, protein engineering, and hardware optimization. In these settings, evaluating the true objective function is often expensive or unsafe, making iterative online evaluation impractical. Offline model-based optimization (MBO) addresses this challenge by learning a surrogate model from a fixed dataset of designs and scores, and then optimizing this surrogate to identify high-quality candidates (Trabucco et al., 2022; Kim et al., 2025).

A common approach to offline MBO is to train a regression-based surrogate by minimizing mean squared error (MSE), then optimize designs through gradient-based search (Kim et al., 2025). However, the surrogate can overestimate the quality of designs outside the training distribution, misleading the search toward poor solutions (Trabucco et al., 2021; Yu et al., 2021). Motivated by the goal of identifying high-quality designs rather than accurately predicting their scores, recent ranking-based approaches learn the relative ordering of designs and demonstrate strong empirical performance (Cao et al., 2007; Tan et al., 2025).

These advances raise a fundamental question: *is accurate ranking alone sufficient for reliable offline MBO?* Offline MBO requires more than selecting among observed designs: it searches for new designs by optimizing the learned surrogate, typically through gradient ascent. Accurate ordering on observed designs does not, by itself, determine how the surrogate behaves along the directions explored during optimization. Surrogates with similar rankings on the offline data can therefore induce different search trajectories and lead to candidates of different quality. Successful optimization depends not only on the surrogate's induced preferences, but also on whether its gradient field provides reliable directions beyond the training distribution.

This gap motivates a structural bias that connects surrogate learning to the directions used during search. We consider a spectral decomposition of the input representation into retained low-frequency

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

modes and complementary high-frequency modes. When the retained structure preserves objective-relevant preferences, sensitivity to the complementary modes can introduce variation that is unnecessary for identifying promising designs. Encouraging preference consistency across these input views addresses this sensitivity during learning, but does not itself restrict the directions taken by the optimizer. Applying the same spectral projection during search makes this restriction explicit, connecting the structure favored by the learning objective to the updates used to generate new designs.

Based on this principle, we introduce SASS (Spectrally Aligned Surrogates and Search), a framework that aligns surrogate learning and search through a shared spectral inductive bias. SASS combines (i) a frequency-aware surrogate that jointly encodes original inputs and their low-frequency projections, (ii) a spectral ranking-consistency regularizer that encourages preservation of induced preferences when high-frequency components are removed, and (iii) a search procedure that projects gradient updates onto the same low-frequency subspace. These components use a common spectral decomposition across representation, learning, and optimization: the surrogate retains access to the original input, the regularizer encourages agreement across input views, and search explicitly restricts updates to the retained subspace.

The contributions of this work are summarized as follows:

- We identify a fundamental limitation of ranking-based offline MBO: *good ranking does not guarantee reliable optimization*, due to instability of the surrogate's gradient field in OOD regions.
- We propose SASS, a framework that aligns a frequency-aware surrogate, spectral ranking consistency, and projected search through a shared low-frequency projection.
- Across five Design-Bench tasks, SASS improves over both ranking baselines in mean normalized score on every task; paired seed-level comparisons in the appendix show consistent, if individually modest, gains in 23 of 25 task–seed comparisons.

2 PRELIMINARIES

Offline model-based optimization. Given a fixed dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with $y_i = f(x_i)$, offline MBO seeks designs that maximize an unknown objective $f : \mathcal{X} \to \mathbb{R}$ without additional evaluations of f during training or search. A surrogate s_θ is learned from $\mathcal{D}$ and optimized to generate candidates. Throughout this work, the surrogate operates on a continuous representation: a normalized design vector for continuous tasks or a logit representation for discrete tasks. A standard gradient-based search update in this representation is

$$x_{t+1} = x_t + \eta \nabla_x s_\theta(x_t), \tag{1}$$

where $\eta > 0$ is the step size. When required, updates are followed by a projection onto the feasible domain; for discrete tasks, the optimized logits are decoded into designs for evaluation. The true objective is used only for evaluation, not to guide surrogate training or candidate search.

Listwise ranking. We learn the surrogate to preserve the relative ordering of designs using a listwise ranking objective. For a list of m designs with observed scores $Y = (y_1, \ldots, y_m)$ and surrogate predictions $Z = (s_\theta(x_1), \ldots, s_\theta(x_m))$, define

$$P_V(i) = \frac{\exp(v_i/\tau)}{\sum_{j=1}^{m} \exp(v_j/\tau)}, \qquad V \in \{Y, Z\},$$

where $\tau > 0$ is a temperature. This distribution expresses relative preference within the list: higher-scoring designs receive greater probability, and a lower temperature concentrates more probability on them. The ListNet objective trains the surrogate to match the target preference distribution:

$$\mathcal{L}_{\text{rank}} = -\sum_{i=1}^{m} P_Y(i) \log P_Z(i). \tag{2}$$

This objective supervises preferences among designs in the list rather than directly minimizing pointwise score errors. In SASS, the same softmax construction also defines the surrogate-induced preference distributions compared across original and projected inputs.

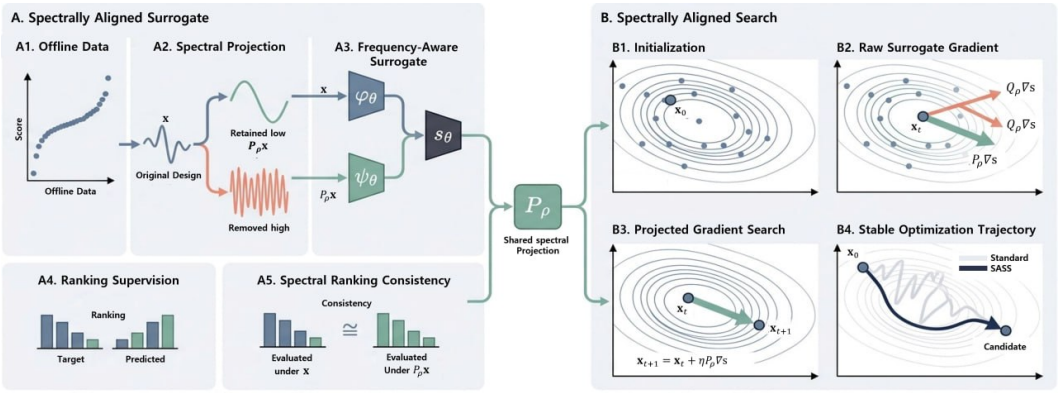


Figure 1: Overview of SASS. (A) Spectrally aligned surrogate learning combines a frequency-aware surrogate (C1) with spectral ranking consistency between original and low-frequency inputs (C2). (B) Spectrally aligned search projects gradient updates onto the same low-frequency subspace (C3). A shared projection P_ρ aligns surrogate learning and search. The trajectories in B4 are schematic.

3 METHOD

Figure 1 provides an overview of SASS (Spectrally Aligned Surrogates and Search), which aligns surrogate learning and search through a shared spectral projection. The framework consists of two stages: (A) spectrally aligned surrogate learning and (B) spectrally aligned search. These stages combine (i) a frequency-aware surrogate (C1), (ii) spectral ranking consistency (C2), and (iii) projected gradient updates (C3).

3.1 SHARED SPECTRAL STRUCTURE

Let x denote the continuous input representation introduced in the preliminaries. Let P_ρ be an orthogonal projection onto a low-frequency subspace, where $\rho \in (0, 1]$ controls the retained-frequency ratio. We decompose the input into its low-frequency representation and complementary residual:

$$\tilde{x} = P_\rho x, \qquad x - \tilde{x} = (I - P_\rho)x.$$

Writing $Q_\rho = I - P_\rho$, the orthogonal projections satisfy

$$P_\rho^\top = P_\rho, \qquad P_\rho^2 = P_\rho, \qquad P_\rho Q_\rho = 0. \tag{3}$$

They define complementary subspaces

$$\mathcal{S}_\rho = \mathrm{range}(P_\rho), \qquad \mathcal{H}_\rho = \mathrm{range}(Q_\rho),$$

containing the retained and removed input modes, respectively. SASS uses this same decomposition to construct the surrogate's input views, regularize their induced preferences, and restrict search directions. Here, frequency refers to modes of the chosen input representation; it does not directly characterize the Fourier spectrum of the surrogate function over the design space.

In our implementation, the projection retains low-frequency modes in the real Fourier representation:

$$P_\rho x = \mathrm{irFFT}(m_\rho \odot \mathrm{rFFT}(x)), \tag{4}$$

where m_ρ is a binary low-pass mask. The same projection, input ordering, and cutoff are used throughout learning and search. For discrete designs, filtering is applied to logits rather than categorical token IDs.

This construction defines a representation-dependent inductive bias. SASS is motivated by settings in which the retained subspace preserves objective-relevant preferences and search directions; low-frequency structure is a modeling premise rather than a universal property of design objectives. We examine this premise through oracle-based diagnostics of value and ranking preservation and controlled failure cases in the appendix. These diagnostics are used for analysis, not during offline training or search.

3.2 SPECTRALLY ALIGNED SURROGATE LEARNING

As illustrated in Figure 1(A), offline designs (A1) and their spectral projections (A2) provide the input views for the frequency-aware surrogate (A3; C1). Learning combines ranking supervision (A4) with spectral ranking consistency (A5; C2).

Frequency-aware surrogate (C1). The frequency-aware surrogate (C1) uses a dual-branch architecture to jointly encode the original input and its low-frequency projection:

$$s_\theta(x) = h_\theta([\phi_\theta(x); \psi_\theta(P_\rho x)]), \tag{5}$$

where ϕ_θ and ψ_θ are multilayer perceptrons that encode the original and projected inputs, respectively, and h_θ predicts a score from their concatenated representations.

The two input views serve complementary roles. The original-input branch retains access to information that the projection removes, while the projected-input branch explicitly represents the retained spectral structure. This architecture therefore does not require the surrogate score to depend exclusively on low-frequency inputs. However, providing a projected view alone does not ensure that the learned preferences remain consistent when high-frequency components are removed. C1 supplies the representation, while C2, introduced next, encourages agreement between the preference distributions obtained by evaluating the same surrogate on original and projected designs.

Spectral ranking consistency (C2). For a list $X = (x_1, \ldots, x_m)$, we evaluate the same surrogate on original and projected designs:

$$z_i = s_\theta(x_i), \qquad \tilde{z}_i = s_\theta(P_\rho x_i).$$

Applying the softmax definition in the preliminaries to Z and $\tilde{Z}$, we penalize discrepancies between their induced preference distributions:

$$\mathcal{L}_{\text{freq}} = D_{\text{KL}}(P_Z \,\|\, P_{\tilde{Z}}) + D_{\text{KL}}(P_{\tilde{Z}} \,\|\, P_Z). \tag{6}$$

The training objective combines supervised ranking with spectral consistency:

$$\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda_{\text{freq}} \mathcal{L}_{\text{freq}}, \tag{7}$$

where $\lambda_{\text{freq}} \geq 0$ controls the regularization strength. The consistency term encourages the surrogate to preserve its preference distribution when high-frequency components are removed.

What the consistency loss constrains. At a fixed temperature $\tau > 0$, equality of the softmax distributions is equivalent to a shared additive shift in the scores:

$$P_Z = P_{\tilde{Z}} \quad \Longleftrightarrow \quad \tilde{Z} = Z + c\mathbf{1} \quad \text{for some } c \in \mathbb{R}. \tag{8}$$

Consequently, $\tilde{z}_i - \tilde{z}_j = z_i - z_j$ for every pair within the list. Exact distributional consistency therefore preserves all pairwise score differences while allowing a common shift. It constrains more than the ordering alone, without requiring the absolute scores of the two views to coincide.

To characterize small deviations, let $p = P_Z$, $q = P_{\tilde{Z}}$, and $\Delta z = \tilde{Z} - Z$. The symmetric KL has the exact form

$$\mathcal{L}_{\text{freq}} = (q - p)^\top \Delta z / \tau,$$

because the softmax normalization terms cancel when summed against $q - p$. Expanding the softmax at fixed Z and τ gives

$$\mathcal{L}_{\text{freq}} = \frac{1}{\tau^2} \Delta z^\top \left(\text{Diag}(p) - pp^\top\right) \Delta z + O(\|\Delta z\|^3). \tag{9}$$

The leading term is a preference-weighted variance:

$$\frac{1}{\tau^2} \sum_i p_i \left(\Delta z_i - \sum_j p_j \Delta z_j \right)^2. \tag{10}$$

Locally, a common score shift is unpenalized, whereas unequal changes incur a penalty weighted by the preference distribution. This explains how the loss encourages consistency of relative preferences across the two views.

For a small projection residual and a surrogate differentiable along the corresponding local neighborhood, a first-order approximation yields

$$\Delta z_i = s_\theta(P_\rho x_i) - s_\theta(x_i) \approx -\nabla_x s_\theta(x_i)^\top Q_\rho x_i. \tag{11}$$

Thus, the local interpretation concerns variations in directional responses along the observed residuals. It does not establish a uniform gradient bound over $\mathcal{H}_\rho$, nor does it justify a first-order approximation for arbitrarily large residuals. This distinguishes the effect encouraged by training from the explicit restriction imposed during search.

3.3 Spectrally Aligned Search

As illustrated in Figure 1(B), search starts from high-scoring offline designs (B1), computes the surrogate gradient (B2), and projects each update using the same spectral projection (B3; C3):

$$x_{t+1} = x_t + \eta P_\rho \nabla_x s_\theta(x_t), \tag{12}$$

where $\eta > 0$ is the step size. The projection removes high-frequency components from each proposed update, aligning search with the spectral structure used during learning. Unlike projecting the design itself, this update preserves the initial high-frequency component:

$$(I - P_\rho)x_{t+1} = (I - P_\rho)x_t.$$

This identity holds before any additional feasibility projection or discrete decoding. When required, candidates are projected back onto the feasible domain after the update; optimized logits are decoded into discrete designs for evaluation.

The search direction also admits a constrained optimization interpretation. For $\mathcal{S}_\rho = \{\delta : P_\rho \delta = \delta\}$,

$$\delta_t^\star = \arg\max_{\delta \in \mathcal{S}_\rho} \left\{ \nabla_x s_\theta(x_t)^\top \delta - \frac{1}{2\eta}\|\delta\|^2 \right\} = \eta P_\rho \nabla_x s_\theta(x_t). \tag{13}$$

The equality follows by optimizing the concave quadratic over the orthogonal subspace. Each step therefore maximizes the local linear surrogate approximation with a quadratic step penalty within the shared low-frequency subspace.

Alignment of learning and search. At the same design x, the standard and projected steps are

$$\delta_{\mathrm{std}} = \eta \nabla_x s_\theta(x), \qquad \delta_{\mathrm{SASS}} = \eta P_\rho \nabla_x s_\theta(x).$$

Their discrepancy is exactly

$$\delta_{\mathrm{std}} - \delta_{\mathrm{SASS}} = \eta Q_\rho \nabla_x s_\theta(x). \tag{14}$$

Training encourages preference consistency across projected views, while search explicitly removes the complementary gradient component from each update. The two mechanisms therefore share a spectral decomposition but impose different constraints.

Conditional trajectory analysis. Let $\{u_t\}$ and $\{v_t\}$ follow standard and projected ascent, respectively, with $u_0 = v_0$:

$$u_{t+1} = u_t + \eta \nabla s_\theta(u_t), \qquad v_{t+1} = v_t + \eta P_\rho \nabla s_\theta(v_t).$$

Assume that the trajectories remain in a region where, for all x, y in that region,

$$\|\nabla s_\theta(x) - \nabla s_\theta(y)\| \le L\|x - y\|, \qquad \|Q_\rho \nabla s_\theta(x)\| \le \varepsilon, \tag{15}$$

with $L > 0$ and $\varepsilon \ge 0$. Setting $d_t = \|u_t - v_t\|$, the updates give

$$d_{t+1} \le (1 + \eta L) d_t + \eta\varepsilon.$$

Since $d_0 = 0$, unrolling this recurrence yields

$$\|u_t - v_t\| \le \frac{\varepsilon}{L}\left((1 + \eta L)^t - 1\right). \tag{16}$$

For a fixed horizon, the bound identifies the high-frequency gradient magnitude as a quantity controlling the discrepancy between the two procedures. It does not guarantee improvement in the true objective or long-term stability. The assumptions are separate from the consistency objective and are not automatically satisfied by the implemented ReLU surrogate. The analysis concerns the stated updates before feasibility projection or discrete decoding. A detailed derivation is provided in the appendix; Algorithm 1 summarizes training and search, and the appendix also specifies the network configuration and optimization settings.

4 EXPERIMENTS

Our experiments examine whether SASS improves offline optimization, how its three components contribute, and whether its benefits are consistent with the shared spectral bias. We evaluate benchmark performance, component contributions, robustness to frequency-targeted perturbations, and preservation of objective-relevant rankings under the spectral projection.

4.1 EXPERIMENTAL SETUP

We evaluate SASS on five Design-Bench tasks (Trabucco et al., 2022): Ant Morphology, D'Kitty Morphology, and Superconductor for continuous optimization, and TF-Bind-8 and TF-Bind-10 for discrete optimization. We compare against the offline MBO methods listed in Table 1: regression-based optimization methods, including BO-qEI (Wilson et al., 2017), CMA-ES (Hansen & Ostermeier, 2001), REINFORCE (Williams, 1992), and Gradient Ascent with its ensemble variants (Trabucco et al., 2022); backward methods, including CbAS (Brookes et al., 2019), MINs (Kumar & Levine, 2020), DDOM (Krishnamoorthy et al., 2023), BONET (Mashkaria et al., 2023), and GTG (Yun et al., 2024); forward methods, including COMs (Trabucco et al., 2021), RoMA (Yu et al., 2021), IOM (Qi et al., 2022), BDI (Chen et al., 2022), ICT (Yuan et al., 2023), Tri-Mentoring (Chen et al., 2023), PGS (Chemingui et al., 2024), FGM (Kuba et al., 2024), and Match-OPT (Hoang et al., 2024); and ranking-based surrogates, including RaM-RankCosine and RaM-ListNet (Tan et al., 2025).

We report the best normalized oracle score among 128 candidate designs. Scores are normalized using the minimum and maximum objective values in the full dataset. The oracle is used for evaluation, not for surrogate training or candidate search. We use a retained-frequency ratio of $\rho = 0.7$ and a consistency weight of $\lambda_{\text{freq}} = 0.1$. SASS and the ranking baselines are evaluated using five matched seeds. Additional implementation details and hyperparameter analyses are provided in the appendix.

4.2 OPTIMIZATION PERFORMANCE

Table 1 summarizes performance across the five tasks. SASS achieves the highest reported mean score on D'Kitty, Superconductor, and TF-Bind-10, and ties for the highest score on TF-Bind-8 at the reported precision. On Ant, SASS exceeds both ranking baselines, while CMA-ES achieves the highest mean score. These results show that SASS remains competitive across both continuous and discrete design spaces.

Compared with the stronger ranking baseline on each task, SASS improves the mean normalized score by 0.020 on Ant, 0.013 on D'Kitty, 0.004 on Superconductor, 0.003 on TF-Bind-8, and 0.011 on TF-Bind-10. Although some absolute gains are small, the paired analysis reports improvements in 23 of 25 task–seed comparisons. This consistency supports the benefit of incorporating the shared spectral structure into a ranking-based optimization pipeline. The paired comparisons are reported in the appendix.

4.3 COMPONENT CONTRIBUTIONS

We isolate the contributions of the frequency-aware surrogate (C1), spectral ranking consistency (C2), and projected search (C3). Table 2 compares variants that add each component individually to the ranking-only base with the complete SASS framework.

The frequency-aware surrogate improves over both ranking baselines on all five tasks, suggesting that jointly representing original and projected inputs provides useful information for optimization. Projected search also exceeds both ranking baselines across the five tasks, supporting the value of restricting update directions even without the complete framework. Consistency regularization alone yields smaller gains: it exceeds RaM-ListNet on all five tasks, but ties RaM-RankCosine on TF-Bind-8 and remains slightly below it on TF-Bind-10.

The complete framework achieves a higher mean score than every individual component on each task. These comparisons support combining spectral structure in representation, learning, and

Table 1: 100th percentile normalized score in Design-Bench, where the best and runner-up results on each task are Blue and Violet. $\mathcal{D}(\text{best})$ denotes the best score in the offline dataset.

Method	Ant	D'Kitty	Superconductor	TF-Bind-8	TF-Bind-10	Mean Rank
$\mathcal{D}(\text{best})$	0.565	0.884	0.400	0.439	0.467	/
BO-qEI	0.812 ± 0.000	0.896 ± 0.000	0.382 ± 0.013	0.802 ± 0.081	0.628 ± 0.036	18.0 / 22
CMA-ES	**1.712 ± 0.754**	0.725 ± 0.002	0.463 ± 0.042	0.944 ± 0.017	0.641 ± 0.036	11.4 / 22
REINFORCE	0.248 ± 0.039	0.541 ± 0.196	0.478 ± 0.017	0.935 ± 0.049	0.673 ± 0.074	14.0 / 22
Grad. Ascent	0.273 ± 0.023	0.853 ± 0.018	0.510 ± 0.028	0.969 ± 0.021	0.646 ± 0.037	11.6 / 22
Grad. Ascent Mean	0.306 ± 0.053	0.875 ± 0.024	0.508 ± 0.019	**0.985 ± 0.008**	0.633 ± 0.030	11.2 / 22
Grad. Ascent Min	0.282 ± 0.033	0.884 ± 0.018	0.514 ± 0.020	0.979 ± 0.014	0.632 ± 0.027	11.5 / 22
CbAS	0.846 ± 0.032	0.896 ± 0.009	0.421 ± 0.049	0.921 ± 0.046	0.630 ± 0.039	15.5 / 22
MINs	0.906 ± 0.024	0.939 ± 0.007	0.464 ± 0.023	0.910 ± 0.051	0.633 ± 0.034	13.0 / 22
DDOM	0.908 ± 0.024	0.930 ± 0.005	0.452 ± 0.028	0.913 ± 0.047	0.616 ± 0.018	14.6 / 22
BONET	0.921 ± 0.031	0.949 ± 0.016	0.390 ± 0.022	0.798 ± 0.123	0.575 ± 0.039	15.1 / 22
GTG	0.855 ± 0.044	0.942 ± 0.017	0.480 ± 0.055	0.910 ± 0.040	0.619 ± 0.029	13.9 / 22
COMs	0.916 ± 0.026	0.949 ± 0.016	0.460 ± 0.040	0.953 ± 0.038	0.644 ± 0.052	9.5 / 22
RoMA	0.430 ± 0.048	0.767 ± 0.031	0.494 ± 0.025	0.665 ± 0.000	0.553 ± 0.000	18.3 / 22
IOM	0.889 ± 0.034	0.928 ± 0.008	0.491 ± 0.034	0.925 ± 0.054	0.628 ± 0.036	13.1 / 22
BDI	0.963 ± 0.000	0.941 ± 0.000	0.508 ± 0.013	0.973 ± 0.000	0.658 ± 0.000	5.9 / 22
ICT	0.915 ± 0.024	0.947 ± 0.009	0.494 ± 0.026	0.897 ± 0.050	0.659 ± 0.024	9.4 / 22
Tri-Mentoring	0.891 ± 0.011	0.947 ± 0.005	0.503 ± 0.013	0.956 ± 0.000	0.662 ± 0.012	7.7 / 22
PGS	0.715 ± 0.046	0.954 ± 0.022	0.444 ± 0.020	0.889 ± 0.061	0.634 ± 0.040	13.2 / 22
FGM	0.923 ± 0.023	0.944 ± 0.014	0.481 ± 0.024	0.811 ± 0.079	0.611 ± 0.008	13.2 / 22
Match-OPT	0.933 ± 0.016	0.952 ± 0.008	0.504 ± 0.021	0.824 ± 0.067	0.655 ± 0.050	8.0 / 22
RaM-RankCosine	0.940 ± 0.028	0.951 ± 0.017	0.514 ± 0.026	**0.982 ± 0.012**	**0.675 ± 0.049**	2.7 / 22
RaM-ListNet	0.949 ± 0.025	**0.962 ± 0.015**	**0.517 ± 0.029**	0.981 ± 0.012	0.670 ± 0.035	2.2 / 22
SASS (Ours)	**0.969 ± 0.026**	**0.975 ± 0.012**	**0.521 ± 0.012**	**0.985 ± 0.019**	**0.686 ± 0.020**	1.2 / 22

search. They do not, by themselves, establish which interactions account for the gains; pairwise component comparisons are examined in the appendix.

Table 2: 100th percentile normalized score in Design-Bench with component-wise ablations of SASS. C1, C2, and C3 correspond to the frequency-aware surrogate, spectral ranking consistency, and projected gradient updates, respectively. The best and runner-up results on each task are Blue and Violet.

Method	Ant	D'Kitty	Superconductor	TF-Bind-8	TF-Bind-10
$\mathcal{D}(\text{best})$	0.565	0.884	0.400	0.439	0.467
RaM-RankCosine	0.940 ± 0.028	0.951 ± 0.017	0.514 ± 0.026	0.982 ± 0.012	0.675 ± 0.049
RaM-ListNet	0.949 ± 0.025	0.962 ± 0.015	0.517 ± 0.029	0.981 ± 0.012	0.670 ± 0.035
Ablations of SASS					
Base + C1 (surrogate only)	**0.960 ± 0.018**	**0.967 ± 0.015**	**0.520 ± 0.020**	**0.983 ± 0.013**	**0.677 ± 0.028**
Base + C2 (consistency only)	0.952 ± 0.019	0.963 ± 0.019	0.518 ± 0.022	0.982 ± 0.014	0.673 ± 0.030
Base + C3 (search only)	0.955 ± 0.021	0.965 ± 0.016	0.519 ± 0.021	**0.983 ± 0.014**	0.676 ± 0.029
SASS (Full)	**0.969 ± 0.026**	**0.975 ± 0.012**	**0.521 ± 0.012**	**0.985 ± 0.019**	**0.686 ± 0.020**

4.4 Robustness to Frequency-Targeted Perturbations

To examine whether the observed benefits are associated with the selected spectral subspace, we evaluate robustness to perturbations of the offline training inputs on Superconductor. High-frequency perturbations are constructed as

$$\delta_{\text{high}} = (I - P_\rho)\epsilon,$$

and rescaled to the prescribed severity, where ϵ is a random vector. We compare SASS with RaM-ListNet and RaM-RankCosine using five matched seeds and matched perturbation conditions.

As perturbation severity increases, SASS exhibits a smaller decline in optimization performance (Table 3). The advantage under high-frequency corruption is reflected in reduced degradation relative to each method's clean performance.

We further compare random, low-frequency, and high-frequency perturbations with matched norms (Table 4). Matching perturbation norms helps distinguish the effect of frequency content from that

Table 3: Superconductor performance under high-frequency-directed training-data noise at increasing severity, averaged over five matched seeds. Drop denotes the decrease relative to the clean score of the same method.

Severity	Method	Mean score	Drop vs. clean	Drop SEM
0.05	SASS	0.517	0.004	0.002
0.05	RaM-ListNet	0.505	0.012	0.004
0.05	RaM-RankCosine	0.500	0.014	0.004
0.10	SASS	0.512	0.009	0.003
0.10	RaM-ListNet	0.488	0.029	0.006
0.10	RaM-RankCosine	0.485	0.029	0.006
0.20	SASS	0.503	0.018	0.005
0.20	RaM-ListNet	0.470	0.047	0.008
0.20	RaM-RankCosine	0.466	0.048	0.008

of perturbation magnitude. If the observed advantage primarily reflects tolerance to corruption irrespective of its spectral structure, a comparable separation might be expected across the matched-norm conditions. Instead, the gap in score degradation is largest under high-frequency perturbations: SASS loses 0.018, compared with 0.047 for RaM-ListNet and 0.048 for RaM-RankCosine. Under low-frequency perturbations, the corresponding decreases are closer, at 0.044, 0.046, and 0.047. This contrast is consistent with the intended spectral bias: SASS is less affected when corruption lies in the complementary subspace. These comparisons assess the complete framework; they do not isolate which component accounts for the difference. The evidence concerns frequency-specific robustness on Superconductor, not uniform robustness to arbitrary distribution shifts.

Table 4: Superconductor score decrease at severity 0.20 under matched-norm perturbations of different frequency content. Each decrease is measured relative to the same method on clean data; lower is better.

Noise type	SASS drop	RaM-ListNet drop	RaM-RankCosine drop
Random	0.026	0.036	0.037
Low-frequency	0.044	0.046	0.047
High-frequency	0.018	0.047	0.048

4.5 Evaluating the Shared Spectral Structure

We examine whether the projection preserves objective-relevant rankings by comparing oracle scores before and after projection (Table 5). At the default $\rho = 0.7$, the reported Spearman correlations are 0.850 on Superconductor, 0.940 on TF-Bind-8, and 0.920 on TF-Bind-10. These measurements indicate substantial preservation of relative quality on the evaluated tasks.

The two measurements address different questions. Oracle rank correlation assesses whether the projected designs retain the relative quality of the original designs, whereas the final score evaluates the outcome of surrogate-guided search. At $\rho = 1$, rank preservation is exact by construction, but the projection no longer restricts the search directions. On TF-Bind-8, for example, $\rho = 0.7$ yields a rank correlation of 0.940 and an optimization score of 0.985, compared with 1.000 and 0.981 at $\rho = 1$. Thus, maximizing rank preservation alone does not identify the best optimization setting in these comparisons.

This pattern is consistent with a trade-off between retaining objective-relevant information and restricting the directions available to surrogate optimization. The diagnostic supports the ranking-preservation premise on these tasks; it does not directly establish gradient alignment. Detailed diagnostics and controlled tests of this premise are provided in the appendix.

Table 5: Oracle value and ranking preservation under the shared spectral projection. Spearman denotes the rank correlation between $f(x)$ and $f(P_\rho x)$; Score denotes the corresponding SASS result.

Task	ρ	Spearman	RMSE	Score
Superconductor	0.5	0.780	8.900	0.521
Superconductor	0.7	0.850	6.100	0.521
Superconductor	0.9	0.910	3.200	0.520
Superconductor	1.0	1.000	0.000	0.517
TF-Bind-8	0.5	0.820	0.040	0.982
TF-Bind-8	0.7	0.940	0.018	0.985
TF-Bind-8	0.9	0.990	0.006	0.984
TF-Bind-8	1.0	1.000	0.000	0.981
TF-Bind-10	0.5	0.800	0.050	0.681
TF-Bind-10	0.7	0.920	0.022	0.686
TF-Bind-10	0.9	0.980	0.008	0.684
TF-Bind-10	1.0	1.000	0.000	0.675

5 Conclusion

Offline model-based optimization (MBO) has recently shifted from regression-based objectives toward ranking-based approaches. In this work, we revisit this paradigm and show that ranking alone is insufficient for reliable optimization, as preserving observed rankings does not fully determine the surrogate geometry.

We propose SASS, which shares a single low-frequency projection between a training-time consistency regularizer and the search-time update rule, so both stages operate under one explicit structural assumption rather than two independent ones. Across five Design-Bench tasks, SASS improves over both ranking baselines in mean normalized score on every task, with reduced degradation relative to those baselines under high-frequency-targeted training perturbations (Section 4).

Under explicit smoothness and gradient assumptions, our finite-horizon analysis (Section 3.3) identifies the high-frequency gradient component as the quantity governing deviation from unconstrained ascent, and our diagnostics in the appendix delineate where this framework's low-frequency premise holds on the evaluated tasks and where it would not. Together, these results indicate that constraining a surrogate's gradient geometry, not only its induced ranking, is a distinct and empirically consequential axis for offline MBO.

References

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 773–782, 2019.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pp. 129–136, 2007.

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, 2024.

Can Chen, Yingxue Zhang, Jie Fu, Xue Liu, and Mark Coates. Bidirectional learning for offline infinite-width model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Can Chen, Christopher Beckham, Zixuan Liu, Xue Liu, and Christopher Pal. Parallel-mentoring for offline model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Minh Hoang, Azza Fadhel, Aryan Deshwal, Janardhan Rao 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)*, 2024.

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

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, 2023.

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

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

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, 2023.

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 (NeurIPS)*, 2022.

Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. Query-level loss functions for information retrieval. *Information Processing & Management*, 44(2): 838–855, 2008.

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 *International Conference on Learning Representations (ICLR)*, 2025.

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, 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, 2022.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

James T. Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions. *arXiv preprint arXiv:1712.00424*, 2017.

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 (NeurIPS)*, 2021.

Ye Yuan, Can Chen, Zixuan Liu, Willie Neiswanger, and Xue Liu. Importance-aware co-teaching for offline model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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 (NeurIPS)*, 2024.