WHEN HIGH SURROGATE SCORES MISLEAD: PAIRWISE RANKING RELIABILITY FOR OFFLINE MODEL-BASED OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

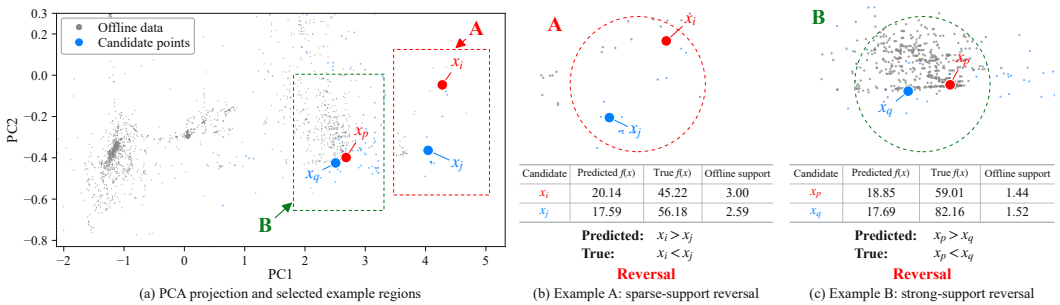
Many scientific and engineering design tasks require searching large design spaces for high-performing designs, yet costly evaluations leave only a fixed and sparsely distributed dataset. Offline model-based optimization (MBO) addresses this setting by learning surrogate models to guide candidate search. However, surrogate-guided optimization may produce misleading preferences when candidates lie beyond well-supported regions. Recent ranking-based methods improve candidate selection by learning relative candidate preferences, but a high surrogate score or rank does not indicate whether the induced pairwise ordering is reliable. Therefore, we propose TRACER, a two-stage framework for improving pairwise ranking reliability in offline MBO. Stage I improves surrogate order preservation through a Kolmogorov-Arnold Network (KAN) based surrogate, Top-ρ contrastive regularization, and a two-stage candidate search strategy. We theoretically characterize the relationship between surrogate approximation error and pairwise ordering preservation, and further analyze how KAN's local spline structure supports order preservation in offline-supported regions. Stage II reassesses candidate comparisons after search by integrating reliability evidence from offline support and local prediction stability, and performs reliability-aware reranking with support-aware candidate refinement. Across five Design-Bench tasks, TRACER achieves the lowest mean rank among 18 baselines. The anonymous link of our source codes is available at `https://anonymous.4open.science/status/TRACER-AF20`.

1 INTRODUCTION

Many scientific and engineering applications require searching large design spaces for high-performing candidates, yet each real evaluation can involve costly experiments and testing (Kim et al., 2026). For example, in superconducting material design, evaluating each candidate requires expensive physical experiments, making exhaustive exploration of the design space impractical. Therefore, only a limited set of historical design–performance pairs is typically available, rather than that data densely cover the entire design space. Offline model-based optimization (MBO) addresses this setting by learning surrogate models from existing data to guide candidate search without additional online evaluations (Kim et al., 2026). However, offline data are often sparse and unevenly distributed, making surrogate-guided candidate selection vulnerable to misleading preferences. In particular, a high surrogate score only reflects the preference produced by the learned model, but does not guarantee that the induced ordering between candidates agrees with their true objective values. This raises a fundamental question that, *when surrogate models assign higher preferences to candidates, can these pairwise rankings be reliably trusted?*

Existing offline MBO methods attempt to improve surrogate-guided candidate selection by learning better objective estimators (named score-based) or by constraining the search toward more reliable regions of the design space (typically support-aware based). Score-based methods (Yu et al., 2021; Trabucco et al., 2021) directly optimize surrogate predictions, while support-aware methods (Krishnamoorthy et al., 2023; Yun et al., 2024) reduce extrapolation risks by favoring candidates closer to previously evaluated designs. More recently, ranking-based methods shift the learning objective from absolute objective prediction to relative candidate preference (Tan et al., 2025b). This ranking perspective is better aligned with finite-budget Top-K selection because the final goal is to identify the best candidates rather than accurately estimate every objective value.

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.



Candidate	Predicted $f(x)$	True $f(x)$	Offline support
x_i	20.14	45.22	3.00
x_j	17.59	56.18	2.59

Predicted: $x_i > x_j$
True: $x_i < x_j$
Reversal

Candidate	Predicted $f(x)$	True $f(x)$	Offline support
x_p	18.85	59.01	1.44
x_q	17.69	82.16	1.52

Predicted: $x_p > x_q$
True: $x_p < x_q$
Reversal

(a) PCA projection and selected example regions (b) Example A: sparse-support reversal (c) Example B: strong-support reversal

Figure 1: High surrogate scores can lead to unreliable pairwise rankings. See Section 4.2 for details.

Learning candidate preference, however, does not guarantee reliable rankings. A surrogate may assign a higher score to one candidate while misordering it against another, and such pairwise misorderings directly affect the final candidate set selected for expensive evaluation. We illustrate this with RaM (Tan et al., 2025b) on the Superconductor task (Fig. 1). In the weak-support case (b), RaM ranks x_i above x_j, although x_j has a higher true objective value, showing that high surrogate scores can be unreliable when the comparison lacks sufficient offline support. More importantly, the same failure occurs under stronger support (c). That is, despite denser offline data around both candidates, RaM still prefers the inferior one. Thus, offline support is useful but insufficient for determining whether a surrogate-induced ordering is trustworthy, revealing the overlooked issue of **pairwise ranking reliability**.

Therefore, effective offline MBO requires not only learning better rankings, but also assessing which rankings should be trusted. This leads to two challenges. The first challenge is how to improve surrogate order preservation so that the initial candidate ranking contains fewer unreliable pairwise comparisons. The second challenge is how to identify and correct unreliable pairwise orderings after candidate search without discarding potentially valuable candidates under a fixed evaluation budget. To address these challenges, we propose **TRACER**, a two-stage framework for improving pairwise ranking reliability in offline MBO. Stage I focuses on improving surrogate-induced ordering quality before reliability assessment. Specifically, TRACER adopts a Kolmogorov-Arnold Network (KAN)-based surrogate, Top-ρ contrastive regularization, and a two-stage candidate search strategy to enhance local order preservation. Our theoretical analysis characterizes the relationship between surrogate approximation error and pairwise ordering preservation, and derives conditions under which improved local approximation leads to better Top-K candidate retention. Stage II focuses on identifying and correcting unreliable pairwise comparisons after candidate search. By integrating reliability evidence from offline support and local prediction stability, TRACER performs reliability-aware reranking and further repairs candidates with insufficient offline support while maintaining the fixed evaluation budget. Our contributions are summarized below.

- We identify **pairwise ranking reliability** as an overlooked issue in offline MBO and show that RaM errors increase as common offline support weakens, while non-negligible reversals remain even for strongly supported pairs.
- We propose **TRACER**, which combines local order preservation with reliability-aware reranking. Our analysis links local surrogate error to pairwise reversals and characterizes when reliability evidence can change the base ordering.
- Across five Design-Bench tasks, TRACER achieves a mean rank of $1.4/19$ against 18 baselines. Controlled experiments show that Stage I reduces reversals on identical candidate pairs and Stage II changes incorrect orderings to correct ones far more often than the reverse.

2 Preliminary

This section introduces the concepts underlying pairwise ranking reliability in offline MBO. We first define offline-supported neighborhoods and preference consistency, which characterize the available local evidence and the correctness of surrogate-induced orderings, respectively. Based on these concepts, we define pairwise ranking reliability as whether a surrogate-induced ordering can be trusted at decision time when the true objective values are unavailable. Pairwise ranking reliability does not represent the probability that an individual comparison is correct. Instead, TRACER uses

observable evidence, including offline support and local prediction consistency, to assess whether a surrogate-induced ordering should be revised.

Definition 1 (Offline-Supported Neighborhood). *Let $\mathcal{D}_X = \{x_i\}_{i=1}^N$ denote observed designs, and let $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a task-specific distance measure. Given a radius $\delta \geq 0$, the* δ-offline-supported neighborhood *is defined as the following.*

$$\Omega_\delta(\mathcal{D}_X) = \left\{ x \in \mathcal{X} : \min_{x_i \in \mathcal{D}_X} d(x, x_i) \leq \delta \right\}. \tag{1}$$

A candidate in $\Omega_\delta(\mathcal{D}_X)$ is called locally supported *at radius δ.*

This definition characterizes proximity to the finite offline observations rather than the unknown data-generating distribution. For continuous designs, $d(\cdot, \cdot)$ is computed in the normalized design space; for discrete designs, it is evaluated in the continuous relaxation used by the surrogate. Because our downstream objective is candidate selection, we next define the relative preference induced by the black-box function and by the learned surrogate.

Definition 2 (Preference Consistency). *For two candidates $x,\ x' \in \mathcal{X}$, the oracle preference is defined as $x \succ_f x' \iff f(x) > f(x')$, whereas the surrogate-induced preference is $x \succ_\theta x' \iff s_\theta(x) > s_\theta(x')$. The surrogate preference is* consistent *with the true preference for a non-tied candidate pair if* $\operatorname{sign}(s_\theta(x) - s_\theta(x')) = \operatorname{sign}(f(x) - f(x'))$.

The surrogate is defined over the entire design space and can therefore compare arbitrary candidates. Nevertheless, direct supervision is available only at observed designs. We are thus primarily interested in whether the surrogate preserves relative preferences for candidate pairs for which the offline observations provide local evidence. To formalize such pairs, let $\mathcal{P}_\delta(\mathcal{D}_X)$ contain all (x, x') for which an observed design $x_i \in \mathcal{D}_X$ satisfies $\max\{d(x, x_i), d(x', x_i)\} \leq \delta$. Requiring a common reference excludes pairs that are individually close to different observed clusters.

Problem 1 (Preference-Consistent Surrogate Learning). *Given $\mathcal{D}$, the objective is to learn a scalar-valued surrogate s_θ whose induced candidate preferences are as consistent as possible with the oracle preferences for candidate pairs in offline-supported neighborhoods. Let Π_δ be a candidate-pair distribution supported on $\mathcal{P}_\delta(\mathcal{D}_X)$, excluding oracle ties. The ideal learning objective is*

$$\theta^\star = \arg\min_\theta \mathcal{R}_{\text{pref}}(\theta; \Pi_\delta), \tag{2}$$

where $\mathcal{R}_{\text{pref}}(\theta; \Pi_\delta) = \mathbb{E}_{(x,x') \sim \Pi_\delta}[\mathbb{I}(\Delta_f(x, x')\Delta_\theta(x, x') < 0)]$, with $\Delta_f(x, x') = f(x) - f(x')$ and $\Delta_\theta(x, x') = s_\theta(x) - s_\theta(x')$. Ties in either the oracle or the surrogate difference are not counted as preference errors, consistent with Definition 2 and the reversal indicator in Eq. (15).

Problem 1 does not require the surrogate outputs to reproduce the exact numerical scale of the black-box objective. Instead, it treats the relative order induced by the surrogate as the primary learning target in regions where the offline observations provide local evidence.

Pairwise ranking reliability. Preference consistency describes whether a surrogate ordering agrees with the oracle ordering, but oracle values are unavailable for optimized candidates at decision time. We use *pairwise ranking reliability* to describe how strongly a surrogate-induced ordering is supported by offline evidence and local surrogate behavior.

3 Proposed Framework

TRACER addresses pairwise ranking reliability through two complementary stages. Stage I improves the reliability of the initial surrogate-induced ranking by enhancing local order preservation and constructing a high-potential candidate pool. Stage II reassesses the resulting candidates using reliability evidence, including predicted preference, offline support, and local prediction consistency, and corrects unreliable pairwise orderings through reliability-aware reranking and support repair. The detailed implementation is provided in Appendix C.1.

3.1 Stage I: Surrogate Learning for Local Order Preservation

Stage I aims to improve the reliability of the initial surrogate-induced ranking before candidate reassessment. It enriches ranking supervision from sparse offline data, improves local order preservation, and constructs a high-potential candidate pool for Stage II. Specifically, TRACER combines

ranking supervision, Top-ρ contrastive regularization, and a KAN-based surrogate to reduce pairwise ordering errors before reliability-aware reranking (Stage II).

3.1.1 Surrogate Training with Enriched Offline Supervision

RaM (Tan et al., 2025b) introduced a ranking-based perspective to offline MBO, a line since extended toward task-agnostic embeddings (Tan et al., 2025a). Rather than fitting a regression model, it trains a surrogate with listwise ranking losses. This perspective aligns naturally with candidate selection, but it also places a stronger requirement on the surrogate. If the model is expected to rank arbitrary candidate pairs, its relative predictions should remain dependable wherever the offline data provide local evidence. Sparse and uneven observations make this difficult. We therefore enrich the supervision extracted from the labels and use an architecture whose behavior in locally supported regions can be analyzed.

Enriching supervision from sparse labels. A pairwise ranking constraint requires the surrogate to preserve the observed order of two designs, i.e., $y_i > y_j \Rightarrow s_\theta(x_i) > s_\theta(x_j)$. Enumerating all such constraints from N offline samples produces $N(N-1)/2$ comparisons. Following RaM (Tan et al., 2025b), we avoid exhaustive pair construction by sampling B_{list} lists of length m. $\mathcal{L}_b = \{(x_{b,j}, y_{b,j})\}_{j=1}^{m}$. Let $\mathbf{s}_b$ and $\mathbf{y}_b$ denote the predicted-score and observed-value vectors of list b. After centering them, we optimize

$$\mathcal{L}_{\text{rank}} = \frac{1}{B_{\text{list}}} \sum_{b=1}^{B_{\text{list}}} \left(1 - \frac{\widetilde{\mathbf{s}}_b^\top \widetilde{\mathbf{y}}_b}{\|\widetilde{\mathbf{s}}_b\|_2 \|\widetilde{\mathbf{y}}_b\|_2 + \epsilon}\right), \tag{3}$$

which jointly learns the relative order within each sampled list.

We further perturb every candidate in each list. $\widetilde{x}_{b,j} = \Pi_{\mathcal{X}}(x_{b,j} + \xi_{b,j})$, where $\xi_{b,j} \sim \mathcal{N}(0, \sigma_v^2 I)$. The perturbed candidates inherit the rank group of their original samples and are not assigned new objective values. Let $\mathcal{A}_b$ contain the original and perturbed views of the Top-ρ candidates, while the remaining views serve as negatives. With normalized representations z_u, we use

$$\mathcal{L}_{\text{ctr}} = -\frac{1}{|\mathcal{B}_v|} \sum_{b \in \mathcal{B}_v} \frac{1}{|\mathcal{A}_b|} \sum_{u \in \mathcal{A}_b} \frac{1}{|\mathcal{A}_b| - 1} \sum_{p \in \mathcal{A}_b \setminus \{u\}} \log \frac{\exp\left(z_u^\top z_p / \tau\right)}{\sum_{a \in \mathcal{V}_b \setminus \{u\}} \exp\left(z_u^\top z_a / \tau\right)}. \tag{4}$$

This objective enriches supervision around the high-value region by pulling together Top-ρ views and separating them from the remaining candidates.

Joint Surrogate Objective. The complete surrogate objective combines the ranking and contrastive terms

$$\mathcal{L}_{\text{sur}} = \mathcal{L}_{\text{rank}} + \lambda_{\text{ctr}} \mathcal{L}_{\text{ctr}}, \tag{5}$$

where $\lambda_{\text{ctr}} > 0$ balances the two terms. $\mathcal{L}_{\text{rank}}$ (Eq. 3) provides the primary preference supervision from observed labels, while $\mathcal{L}_{\text{ctr}}$ (Eq. 4) regularizes the representation around high-value designs.

KAN-Based Surrogate Modeling. Data-side enrichment makes fuller use of the available labels, but it does not change how the surrogate extrapolates. When two candidates lie in an offline-supported neighborhood but no observed point falls exactly between them, the surrogate's relative prediction depends on its function class. The architecture choice therefore matters for preference preservation. Recent empirical studies have observed that Kolmogorov–Arnold Networks (KAN) and MLPs can achieve comparable pointwise errors while exhibiting different relative-order accuracy on several benchmark functions (Ma et al., 2025). This motivated us to examine KAN's structural properties from a preference-preservation perspective. A conditional analysis of why this structural property supports preference preservation in locally supported regions is provided in Section 3.1.2.

Candidate Search and Pool Construction. Starting from the top K_{out} observed designs $X^{(0)}$, we optimize each seed against a sampled offline reference set $\mathcal{R}_x$.

$$\mathcal{J}_{\text{search}}(x) = \frac{1}{|\mathcal{R}_x|} \sum_{r \in \mathcal{R}_x} \sigma(s(x) - s(r)), \tag{6}$$

where σ is the sigmoid function. This objective promotes candidates that outrank the observed references. We perform projected gradient ascent within a radius R of each seed.

$$x^{(t+1)} = \Pi_{\mathcal{B}_R(x^{(0)})}\left(x^{(t)} + \eta_t \nabla_x \mathcal{J}_{\text{search}}(x^{(t)})\right). \tag{7}$$

After T_1 iterations, we obtain $X^{(1)}$. We then select its top K_2 candidates according to $s(x)$ and apply T_2 steps of unconstrained refinement,

$$x^{(t+1)} = x^{(t)} + \eta_2 \nabla_x s(x^{(t)}),$$

yielding $X^{(2)}$. The final candidate pool is $\mathcal{C}^{\text{pool}} = X^{(1)} \cup X^{(2)}$, which is reassessed in Section 3.2.

3.1.2 Order-Preservation Analysis

Our analysis contains two levels of results. First, we establish an architecture-independent connection between surrogate approximation error and pairwise ordering preservation. This result explains why reducing local approximation error can improve ranking reliability. Second, we provide a conditional analysis of why KAN's local spline parameterization may reduce interference in offline-supported regions. The KAN-specific analysis serves as an explanation for the surrogate choice rather than an unconditional superiority claim.

Theorem 1 (Pointwise margin-controlled order preservation)**.** *Let $\mathcal{C} \subseteq \mathcal{X}$ be finite, and let $\widetilde{s}_\theta = a s_\theta + b$, $a > 0$, satisfy $|\widetilde{s}_\theta(x) - f(x)| \le \varepsilon(x)$ for every $x \in \mathcal{C}$. Then for any pair $x, x' \in \mathcal{C}$,*

$$|f(x) - f(x')| > \varepsilon(x) + \varepsilon(x') \Longrightarrow \operatorname{sign}(s_\theta(x) - s_\theta(x')) = \operatorname{sign}(f(x) - f(x')). \tag{8}$$

Consequently, the pairwise reversal rate on $\mathcal{C}$ is at most the fraction of pairs whose true margin does not exceed the sum of the two endpoint errors (Appendix B.1.1). The uniform margin condition is recovered with $\varepsilon(\cdot) \equiv e_\infty := \max_{x \in \mathcal{C}} |\widetilde{s}_\theta(x) - f(x)|$; Proposition 2 further bounds $\varepsilon(x)$ by the distance of x to the offline reference set,which is the same quantity that Stage II uses as the support signal F.

Theorem 1 is architecture independent. The KAN-specific locality result,which is an exact first-order decoupling of the coefficient-update channel under coordinate separation, together with a separation-independent rank-2 residual carried by the trainable scales (Theorem 3),and the associated scope conditions are given in Appendix B.1. Appendix D.3.1 provides the capacity-controlled empirical comparison with MLPs.

3.2 Stage II: Pairwise Ranking Reliability Assessment and Reranking

Stage II addresses pairwise orderings that remain unreliable after candidate search. Although the evidence signals are computed for individual candidates, they are used to reassess pairwise comparisons because the reliability of an ordering depends on the evidence associated with both candidates. Therefore, Stage II does not predict an absolute correctness probability for each pair. Instead, it adjusts the surrogate-induced ordering according to observable reliability evidence. A fixed evidence aggregation mechanism combines these signals into a reliability-aware priority. The resulting priority is used together with the surrogate preference to revise the initial ordering when additional support and consistency evidence justifies such changes.

3.2.1 Candidate Evidence for Pairwise Ranking Reliability

Three diagnostic signals. For each candidate $x_i \in \mathcal{C}^{\text{pool}}$, TRACER considers three complementary signals, namely surrogate preference, offline support, and local consistency. Since these quantities have different scales and directions, we first apply min-max normalization over the candidate pool. For any candidate-level scalar v_i, define

$$\text{Norm}_{\mathcal{C}}(v_i) = \frac{v_i - \min_{x_j \in \mathcal{C}^{\text{pool}}} v_j}{\max_{x_j \in \mathcal{C}^{\text{pool}}} v_j - \min_{x_j \in \mathcal{C}^{\text{pool}}} v_j + \epsilon}, \tag{9}$$

where $\epsilon > 0$ prevents numerical instability.

We use the surrogate score $s(x_i)$ to characterize the predicted preference of x_i. Its offline support is measured by the distance to the nearest reference design, $d_i = \min_{r \in \widetilde{\mathcal{R}}} \|x_i - r\|_2$, where $\widetilde{\mathcal{R}}$ denotes

the offline reference set. To assess local consistency, we perturb the candidate M times and measure the prediction variation $\varsigma_i = \mathrm{Std}_{m=1}^{M}[s(x_i + \xi_m)]$, where $\xi_m \sim \mathcal{N}(0, \sigma_n^2 I)$.

The three diagnostic signals are then defined as $Q_i = \mathrm{Norm}_{\mathcal{C}}(s(x_i))$, $F_i = 1 - \mathrm{Norm}_{\mathcal{C}}(d_i)$, $S_i = 1 - \mathrm{Norm}_{\mathcal{C}}(\varsigma_i)$. Here, Q_i measures the predicted preference of x_i relative to the other candidates in the pool, F_i measures its relative support from the offline reference set, and S_i measures the consistency of its surrogate prediction under local perturbations. All three signals are normalized within the candidate pool, with larger values indicating more favorable evidence.

Although Q_i, F_i, and S_i are computed candidate-wise, their role in Stage II is pairwise. For a comparison (x_i, x_j), Q_i and Q_j determine the relative predicted preference, whereas (F_i, S_i) and (F_j, S_j) provide support and stability evidence for the two endpoints of that ordering. A high-Q_i/low-F_i candidate may be favored through weakly supported extrapolation, while a high-Q_i/low-S_i candidate may lie near a locally sensitive surrogate response. We therefore aggregate the three signals through the fuzzy-rule-based mechanism introduced below.

To capture the state-dependent interaction among the three signals, we fuzzify Q_i, F_i, and S_i into coarse linguistic states and apply a fixed zero-order Sugeno policy. Each activated rule assigns a predefined candidate-priority consequent, and their weighted average produces the diagnostic score $T_i = \frac{\sum_r a_{r,i} w_r}{\sum_r a_{r,i} + \epsilon}$. Higher T_i indicates a more favorable combination of predicted preference, offline support, and local consistency. We interpret T_i as an evidence-conditioned diagnostic priority rather than a calibrated probability of candidate correctness. The complete membership functions, rule base, and consequents are provided in Appendix C.3.

3.2.2 Reranking with Reliability Evidence and Support Repair

The fuzzy diagnosis is combined with predicted preference through the base priority and a multiplicative reliability gate.

$$B_i = \frac{\alpha Q_i + \beta T_i}{\alpha + \beta}, \qquad G_i = f_0 + (1 - f_0)\sqrt{F_i S_i}, \qquad \mathrm{Final}_i = B_i G_i. \tag{10}$$

We first separate candidates by the hard-pass indicator $H_i = \mathbb{I}[F_i \geq \tau_F]\mathbb{I}[S_i \geq \tau_S]\mathbb{I}[T_i \geq \tau_T]$, then rank candidates within each tier by Final_i. The top K_{out} candidates form $\widehat{\mathcal{X}}$.

Theorem 2 (Pairwise Reordering under Reliability Gating)**.** *Consider two candidates x_i and x_j in the same hard-pass tier with $B_i > B_j > 0$. Let $\mathrm{Final}_k = B_k G_k$ and $G_k = f_0 + (1 - f_0)\sqrt{F_k S_k}$ for $k \in \{i, j\}$, where $f_0 \in (0, 1)$. Stage II reverses the base priority order, $\mathrm{Final}_j > \mathrm{Final}_i$, if and only if*

$$\frac{G_j}{G_i} > \frac{B_i}{B_j}. \tag{11}$$

because $G_k \in [f_0, 1]$, such a reversal is impossible when

$$\frac{B_i}{B_j} \geq \frac{1}{f_0}. \tag{12}$$

Theorem 2 shows that Stage II does not arbitrarily overwrite the base ordering. A lower-priority candidate can overtake a higher-priority one only when the gate provides enough evidence to offset the original priority advantage. The second condition shows that sufficiently separated base priorities cannot be reversed by the gate. The proof and an additional gate-suppression bound are provided in Appendix B.4.

Rather than deleting a selected candidate with weak support or unstable predictions, we move it toward its nearest offline reference. Let $R_i = \mathbb{I}[F_i < \tau_F \vee S_i < \tau_S]$ and $r_i^\star = \arg\min_{r \in \widetilde{\mathcal{R}}} \|x_i - r\|_2$. We deliver

$$x_i^\star = x_i + R_i \alpha_{\mathrm{pb}}(r_i^\star - x_i), \qquad \alpha_{\mathrm{pb}} \in [0, 1]. \tag{13}$$

Proposition 1 (Budget-preserving support repair)**.** *For every repaired candidate, $d_{\widetilde{\mathcal{R}}}(x_i^\star) \leq (1 - \alpha_{\mathrm{pb}}) d_{\widetilde{\mathcal{R}}}(x_i)$, while $|\mathcal{X}^\star| = |\widehat{\mathcal{X}}| = K_{\mathrm{out}}$.*

The proof is given in Appendix B.4.

4 EXPERIMENTS

We first compare TRACER with 18 representative offline MBO methods on the widely adopted testbed Design-Bench (Trabucco et al., 2022), which includes five real-world tasks covering both continuous optimization scenarios (Ant Morphology, D'Kitty Morphology, and Superconductor), and discrete sequence optimization scenarios (TF-Bind-8 and TF-Bind-10). We then analyze the contribution of each component and investigate how reliability evidence affects candidate selection.

4.1 EXPERIMENTAL SETUP

Evaluation Metrics. We use normalized score introduced in Design-Bench (Trabucco et al., 2022) as the primary metric for evaluating optimization performance. LPRA and Spearman correlation are used to evaluate surrogate ordering quality following previous ranking-based studies (Tan et al., 2025b). We further report directional and magnitude stability metrics to measure local surrogate stability (Sitawarin et al., 2022; Joo et al., 2023). Higher values indicate better performance for all metrics. Detailed metric definitions are provided in Appendix D.1.

Parameter Settings. TRACER uses fixed hyperparameters across all tasks without task-specific tuning. The KAN surrogate adopts the architecture $[d, [32, 0], [16, 0], 1]$, where d denotes the input dimension. The grid size and spline order are set to 3 and 2, respectively. The surrogate is optimized with Adam for 100 steps using a learning rate of 3×10^{-4} and the RankCosine objective. During Stage I training, Top-ρ contrastive regularization generates augmented views by perturbing the original candidates. The regularization weight, positive-pair ratio, temperature, and perturbation magnitude are set to $w_{\mathrm{ctr}} = 0.1$, $\rho = 0.2$, $\tau = 0.1$, and $\sigma_v = 0.01$, respectively. Candidate search starts from 128 initial candidates and returns 128 final candidates. The first-stage search uses 80 steps with a learning rate of 10^{-3} for continuous tasks and 100 steps with a learning rate of 10^{-1} for discrete tasks. Stage II further refines 8 candidates based on reliability evidence. Unless otherwise specified, all results are averaged over eight random seeds. Complete baseline descriptions and hyperparameter configurations are provided in Appendix D.1.

4.2 MAIN RESULTS

Overall results. Table 1 reports the 100th-percentile normalized scores on five Design-Bench tasks. TRACER achieves the highest score on four tasks, reaching 0.976 on Ant Morphology, 0.987 on D'Kitty Morphology, 0.544 on Superconductor, and 0.991 on TF-Bind-8. On TF-Bind-10, TRACER obtains 0.678, compared with 0.686 for DAR and 0.685 for ROOT. Overall, TRACER achieves the lowest mean rank of 1.4 among the 19 compared methods. Since TRACER is developed from the RaM codebase, we further compare it with the better result of the two RaM variants on each task. TRACER improves nScore_{100} by 0.027, 0.025, 0.027, 0.009, and 0.003 on Ant Morphology, D'Kitty Morphology, Superconductor, TF-Bind-8, and TF-Bind-10, respectively. These results show that the proposed modifications consistently improve the candidate quality obtained from the ranking-based baseline.

RQ1. ***Can High Surrogate Scores Imply Reliable Pairwise Rankings?***

To answer RQ1, we analyze whether the candidate preferences learned by a ranking-based optimizer provide reliable pairwise comparisons. We use RaM (Tan et al., 2025b), a representative ranking-based offline MBO method, as the basis for this analysis. Since TRACER is built upon the RaM framework, this analysis also reveals the limitation of ranking-based optimization addressed by our method. We run RaM on the Superconductor task and collect the candidates returned during search. For each candidate pair, we compare the ordering induced by RaM's surrogate scores with the ordering according to the true objective values from Design-Bench, which are used only for post-hoc evaluation. A pair is counted as a reversal when the two orderings disagree. We further analyze reversals under different levels of offline support. To avoid evaluating reliability only on optimizer-selected candidates, we augment this set with randomly sampled designs to cover a broader region of the design space. Candidate pairs are grouped according to the common offline support distance D_{ij}^{sup}, which measures the minimum radius required for a single offline neighborhood to jointly cover both candidates. A larger D_{ij}^{sup} indicates weaker offline support. Candidate pairs are sorted by D_{ij}^{sup} and divided into five groups with approximately equal numbers of pairs. The average support distances of Q_1 to Q_5 are 29.97, 16.74, 10.53, 7.06, and 4.45, respectively, where Q_1 represents the weakest offline support and Q_5 represents the strongest offline support.

High surrogate scores can be misleading under both weak and strong offline support. Figure 1 presents two representative ranking reversals produced by RaM. In the weak-support case, RaM assigns predicted scores of 20.14 and 17.59 to x_i and x_j, respectively, while their true objective values

Table 1: Normalized scores at the 100th percentile on Design-Bench. $\mathcal{D}(\text{best})$ denotes the best score in the offline dataset. Mean Rank is averaged over five tasks among 19 methods.

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	–
CbAS {ICML, (Brookes et al., 2019)}	0.846 ± 0.032	0.896 ± 0.009	0.421 ± 0.049	0.921 ± 0.046	0.630 ± 0.039	15.2 / 19
MINs {NeurIPS, (Kumar & Levine, 2020)}	0.906 ± 0.024	0.939 ± 0.007	0.464 ± 0.023	0.910 ± 0.051	0.633 ± 0.034	13.3 / 19
DDOM {ICML, (Krishnamoorthy et al., 2023)}	0.908 ± 0.024	0.930 ± 0.005	0.452 ± 0.028	0.913 ± 0.047	0.616 ± 0.018	14.2 / 19
BONET {ICML, (Mashkaria et al., 2023)}	0.921 ± 0.031	0.949 ± 0.016	0.390 ± 0.022	0.798 ± 0.123	0.575 ± 0.039	14.5 / 19
GTG {NeurIPS, (Yun et al., 2024)}	0.855 ± 0.044	0.942 ± 0.017	0.480 ± 0.055	0.910 ± 0.040	0.619 ± 0.029	13.9 / 19
COMs {ICML, (Trabucco et al., 2021)}	0.916 ± 0.026	0.949 ± 0.016	0.460 ± 0.040	0.953 ± 0.038	0.644 ± 0.052	10.3 / 19
RoMA {NeurIPS, (Yu et al., 2021)}	0.430 ± 0.048	0.767 ± 0.031	0.494 ± 0.025	0.665 ± 0.000	0.553 ± 0.000	17.1 / 19
IOM {NeurIPS, (Qi et al., 2022)}	0.889 ± 0.034	0.928 ± 0.008	0.491 ± 0.034	0.925 ± 0.054	0.628 ± 0.036	13.2 / 19
BDI {NeurIPS, (Chen et al., 2022)}	0.963 ± 0.000	0.941 ± 0.000	0.508 ± 0.013	0.973 ± 0.000	0.658 ± 0.000	7.2 / 19
ICT {NeurIPS, (Yuan et al., 2023)}	0.915 ± 0.024	0.947 ± 0.009	0.494 ± 0.026	0.897 ± 0.050	0.659 ± 0.024	10.4 / 19
Tri-Mentoring {NeurIPS, (Chen et al., 2023)}	0.891 ± 0.011	0.947 ± 0.005	0.503 ± 0.013	0.956 ± 0.000	0.662 ± 0.012	9.1 / 19
PGS {AAAI, (Chemingui et al., 2024)}	0.715 ± 0.046	0.954 ± 0.022	0.444 ± 0.020	0.889 ± 0.061	0.634 ± 0.040	13.2 / 19
FGM {AISTATS, (Grudzien et al., 2024)}	0.923 ± 0.023	0.944 ± 0.014	0.481 ± 0.024	0.811 ± 0.079	0.611 ± 0.008	13.2 / 19
Match-OPT {ICML, (Hoang et al., 2024)}	0.933 ± 0.016	0.952 ± 0.008	0.504 ± 0.021	0.824 ± 0.067	0.655 ± 0.050	9.0 / 19
RaM-RankCosine {ICLR, (Tan et al., 2025b)}	0.940 ± 0.028	0.951 ± 0.017	0.514 ± 0.026	0.982 ± 0.012	0.675 ± 0.049	5.0 / 19
RaM-ListNet {ICLR, (Tan et al., 2025b)}	0.949 ± 0.025	0.962 ± 0.015	0.517 ± 0.029	0.981 ± 0.012	0.670 ± 0.035	4.0 / 19
ROOT† {NeurIPS, (Dao et al., 2025)}	0.965 ± 0.014	0.972 ± 0.005	0.525 ± 0.012	0.986 ± 0.007	0.685 ± 0.053	2.2 / 19
DAR {arXiv, (Lyu et al., 2026)}	0.941 ± 0.013	0.959 ± 0.006	0.506 ± 0.008	0.988 ± 0.008	$\mathbf{0.686 \pm 0.056}$	3.6 / 19
TRACER (ours)	$\mathbf{0.976 \pm 0.018}$	$\mathbf{0.987 \pm 0.015}$	$\mathbf{0.544 \pm 0.017}$	$\mathbf{0.991 \pm 0.005}$	0.678 ± 0.025	**1.4 / 19**

† ROOT, the second-best performing algorithm in this table, was selected as a Spotlight paper at NeurIPS 2025.

are 45.22 and 56.18. Although RaM prefers x_i, x_j achieves a better true objective value. This case shows that candidate preferences derived from surrogate scores can become unreliable when offline support is limited. However, unreliable comparisons are not restricted to weakly supported regions. In a more strongly supported region, RaM still predicts x_p to be better than x_q with predicted scores of 18.85 and 17.69, while their true objective values of 59.01 and 82.16 indicate the opposite ordering. This example demonstrates that offline support alone does not guarantee correct fine-grained pairwise comparisons. These examples motivate a broader analysis of ranking reliability across different levels of offline support. Figure 2-(a) and Figure 3-(a) summarizes the reversal rates under different levels of offline support. Incorrect pairwise orderings exist across the entire design space rather than being restricted to weakly supported regions. Although weaker offline support is associated with a higher risk of unreliable comparisons, stronger offline support does not eliminate ranking reversals. Therefore, offline support provides useful evidence about pairwise ranking reliability, but it cannot replace explicit assessment of whether a surrogate-induced ordering is trustworthy.

RQ2. ***Does Stage I Improve Local Order Preservation?***

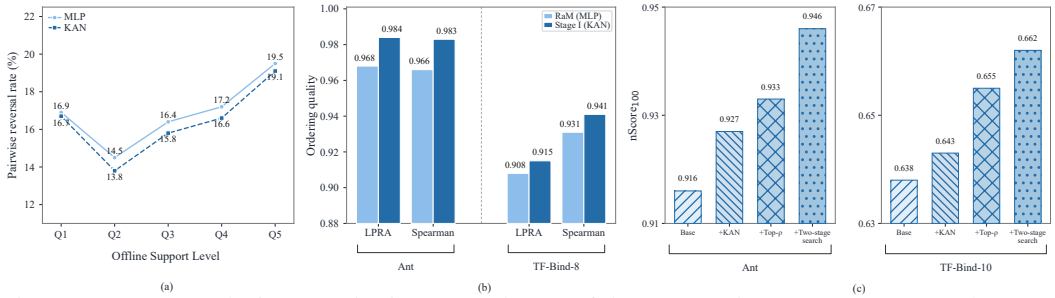
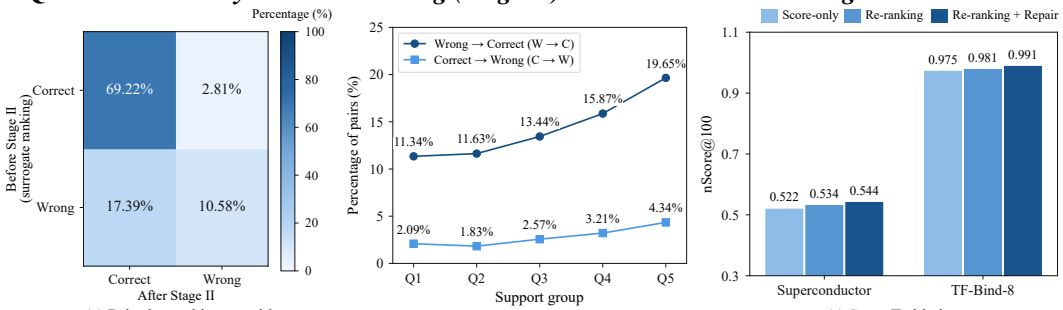


Figure 2: Stage I analysis. (a) Pairwise reversal rates of the MLP and KAN surrogates on the same candidate pairs from RQ1. (b) LPRA and Spearman correlation under matched parameter budgets. (c) Cumulative ablation of the Stage I components.

To answer RQ2, we investigate whether Stage I improves surrogate-induced ordering quality and provides a stronger ranking basis for reliability-aware reranking. On Superconductor, we construct a diagnostic candidate pool by combining 2^{13} randomly sampled designs from the offline dataset with 363 final candidates returned by RaM, resulting in 8,555 unique candidates after removing duplicates. Candidate pairs are grouped according to the common offline support distance D_{ij}^{sup}, where larger values indicate weaker common offline support. The five groups Q_1–Q_5 are ordered from weak to strong offline support and are used to compare the pairwise ordering reliability of the original RaM surrogate and the Stage I surrogate. Starting from the original RaM setting with an

MLP surrogate, we replace the surrogate with KAN while keeping the candidate pool and evaluation protocol unchanged. As shown in Figure 2(a), KAN consistently reduces pairwise reversal rates across all offline support levels, decreasing from 16.9% to 16.7%, 14.5% to 13.8%, 16.4% to 15.8%, 17.2% to 16.6%, and 19.5% to 19.1% from Q_1 to Q_5, respectively. Figure 2(b) further evaluates ordering quality under comparable trainable parameter counts. On Ant, KAN improves LPRA from 0.968 to 0.984 and Spearman correlation from 0.966 to 0.983. On TF-Bind-8, it improves LPRA from 0.908 to 0.915 and Spearman correlation from 0.931 to 0.941. These results show that the improved ordering quality is not caused by increased model capacity, but by better preservation of candidate relations. Figure 2(c) presents the cumulative contribution of Stage I components. Starting from the base surrogate, progressively adding KAN, Top-ρ contrastive regularization, and two-stage search consistently improves nScore_{100}. On Ant, the score increases from 0.916 to 0.927, 0.933, and 0.946, while on TF-Bind-10 it improves from 0.638 to 0.643, 0.655, and 0.662. These results verify that each Stage I component contributes to constructing a stronger initial ranking for Stage II.

RQ3. ***Does Reliability-Aware Reranking (Stage II) Correct Pairwise Ordering Errors?***



(a) Pairwise ranking transition after Stage II re-ranking

(b) Offline support level (increasing from weak to strong)

(c) Stage II ablation on optimization performance

Figure 3: Reliability-aware reranking (Stage II) for correcting pairwise misorderings.

To answer RQ3, we compare the pairwise rankings before and after Stage II using the same candidate pool generated by Stage I. Figure 3(a) shows that Stage II preserves 69.22% of the pairs that are already correctly ordered, while correcting 17.39% of previously incorrect pairs. In contrast, only 2.81% of originally correct pairs are changed into wrong orderings, and 10.58% remain incorrect after reranking. These results show that Stage II does not perturb the ranking arbitrarily. Instead, it corrects substantially more unreliable pairwise orderings than the new errors it introduces. Figure 3(b) further shows that, across all offline support groups, beneficial changes from wrong to correct consistently exceed harmful changes from correct to wrong. Specifically, the wrong-to-correct ratio increases from 11.34% in Q_1 to 19.65% in Q_5, whereas the correct-to-wrong ratio only changes from 2.09% to 4.34%. This result indicates that Stage II can effectively identify unreliable pairwise orderings under different offline support conditions. Offline support and local stability therefore provide complementary reliability evidence for reassessing surrogate-induced rankings rather than merely serving as heuristic signals. Figure 3(c) evaluates whether these pairwise corrections translate into improved final optimization performance on the same Stage I candidate pool. On Superconductor, the nScore_{100} improves from 0.522 with score-only selection to 0.534 after reranking, and further to 0.544 after adding repair. On TF-Bind-8, the score improves from 0.975 to 0.981 and then to 0.991. These gains demonstrate that the benefit of Stage II is not limited to pairwise ranking statistics. By correcting unreliable comparisons and refining candidates with insufficient offline support, Stage II further improves the quality of the final selected candidates.

5 Conclusion

Offline MBO relies on surrogate models to guide candidate search from limited offline data, yet our analysis reveals that surrogate preference does not necessarily correspond to reliable pairwise ranking. High surrogate scores can still lead to incorrect candidate orderings under different levels of offline support, motivating explicit modeling of pairwise ranking reliability. We propose TRACER, a two-stage framework that improves ranking reliability by combining surrogate order preservation with reliability-aware reranking. Extensive experiments on five Design-Bench tasks demonstrate that TRACER consistently outperforms representative offline MBO methods. Further analysis shows that Stage I improves surrogate ordering quality and reduces pairwise ranking errors, while Stage II effectively corrects unreliable comparisons and improves final candidate selection. Future work will explore more adaptive reliability estimation strategies for offline optimization.

AI USE STATEMENT

Generative AI tools were used to assist with writing and polishing the manuscript. No new experiments, empirical results, scientific claims, methods were generated. The technical content was derived from the authors' source manuscript. The authors reviewed the AI-assisted edits and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The model architecture, learning objectives, candidate-search procedure, diagnostic rules, and support-repair mechanism are specified in Section 3. Experimental metrics and settings are summarized in Section 4, with complete theoretical derivations, hyperparameters, evaluation protocols, and supplementary results provided in Appendix B through Appendix D. The anonymous source code is included with the submission materials.

REFERENCES

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

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

Can (Sam) Chen, Yingxue Zhang, Jie Fu, Xue (Steve) Liu, and Mark Coates. Bidirectional learning for offline infinite-width model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 29454–29467, 2022.

Can (Sam) Chen, Christopher Beckham, Zixuan Liu, Xue (Steve) Liu, and Chris Pal. Parallel-mentoring for offline model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 76619–76636, 2023.

Seungjin Choi. Conformal candidate certification for offline model-based optimization. *arXiv preprint arXiv:2606.15217*, 2026.

Manh Cuong Dao, The Hung Tran, Phi Le Nguyen, Thao Nguyen Truong, and Trong Nghia Hoang. ROOT: Rethinking offline optimization as distributional translation via probabilistic bridge. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 109458–109492, 2025.

Carl de Boor. *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences*. Springer, New York, revised edition, 2001.

Clara Fannjiang, Stephen Bates, Anastasios N. Angelopoulos, Jennifer Listgarten, and Michael I. Jordan. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022.

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

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 International Conference on Machine Learning (ICML)*, pp. 18374–18393. PMLR, 2024.

Sunghwan Joo, SeokHyeon Jeong, Juyeon Heo, Adrian Weller, and Taesup Moon. Towards more robust interpretation via local gradient alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8168–8176. AAAI Press, 2023.

Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Minsu Kim, Jiayao Gu, Ye Yuan, Taeyoung Yun, Zixuan Liu, Yoshua Bengio, and Can (Sam) Chen. Offline model-based optimization: Comprehensive review. *Transactions on Machine Learning Research*, 2026.

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

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

Pengqi Li, Lizhong Ding, Jiarun Fu, Chunhui Zhang, Guoren Wang, and Ye Yuan. Generalization bounds for Kolmogorov–Arnold networks (KANs) and enhanced KANs with lower Lipschitz complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 85761–85795, 2025.

Christoffer Löffler. Pairwise diverse and uncertain gradient-sampling for similarity retrieval. *Sensors*, 25(22):6899, 2025.

Shen-Huan Lyu, Rong-Xi Tan, Ke Xue, Yi-Xiao He, Yu Huang, Qingfu Zhang, and Chao Qian. On the learnability of offline model-based optimization: A ranking perspective. *arXiv preprint arXiv:2603.04000*, 2026.

Zeyuan Ma, Zhiyang Huang, Jiacheng Chen, Zhiguang Cao, and Yue-Jiao Gong. Surrogate learning in meta-black-box optimization: A preliminary study. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1137–1145. ACM, 2025.

Satvik Mehul Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative pretraining for black-box optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 24173–24197. PMLR, 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)*, volume 35, pp. 13226–13237, 2022.

Chawin Sitawarin, Zachary J. Golan-Strieb, and David Wagner. Demystifying the adversarial robustness of random transformation defenses. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 20232–20252. PMLR, 2022.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Rong-Xi Tan, Ming Chen, Ke Xue, Yao Wang, Yaoyuan Wang, Fu Sheng, and Chao Qian. Towards universal offline black-box optimization via learning language model embeddings. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 58499–58544. PMLR, 2025a.

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

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-Bench: Benchmarks for data-driven offline model-based optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 21658–21676. PMLR, 2022.

Yixuan Wang, Jonathan W. Siegel, Ziming Liu, and Thomas Y. Hou. On the expressiveness and spectral bias of KANs. In *International Conference on Learning Representations (ICLR)*, 2025.

Yonghan Yang, Ye Yuan, Zipeng Sun, Linfeng Du, Bowei He, Haolun Wu, Can Chen, and Xue Liu. Support-proximity augmented diffusion estimation for offline black-box optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 306. PMLR, 2026.

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)*, volume 34, pp. 4619–4631, 2021.

Ye Yuan, Can (Sam) Chen, Zixuan Liu, Willie Neiswanger, and Xue (Steve) Liu. Importance-aware co-teaching for offline model-based optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 55718–55733, 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)*, volume 37, pp. 83847–83876, 2024.

Xianyang Zhang and Huijuan Zhou. Generalization bounds and model complexity for Kolmogorov–Arnold networks. In *International Conference on Learning Representations (ICLR)*, 2025.

APPENDIX

A Related Work **14**

A.1 Score-driven selection. 14

A.2 Support-aware generation. 14

A.3 Ranking-based ordering. 14

A.4 Post-search delivery. 15

B Additional Theoretical Analysis **15**

B.1 Support Locality of the KAN Ranking Surrogate 15

B.2 Robustness of Rank-Defined Top-ρ Supervision 19

B.3 Top-ρ Contrastive Geometry and Stability 20

B.4 Reliability-Aware Reranking and Diagnosis-Guided Repair 21

C Additional Method Details **22**

C.1 Detailed Framework Overview 22

C.2 Context-Relative Candidate Search 23

C.3 Fuzzy Rules for Candidate Diagnosis 23

D Additional Experimental Details and Results **25**

D.1 Experimental Settings 26

D.2 Pairwise Ranking Reliability Diagnostics 27

D.3 Controlled Diagnostic Experiments 27

D.4 Additional Experiments 28

A Related Work

In this section, we categorize existing offline model-based optimization methods according to the decision quantities they primarily address, grouping prior work into four categories: candidate-wise score estimation, support-aware generation, ranking-based ordering, and post-search reliability assessment.

A.1 Score-driven selection.

Score-driven selection ranks candidate designs according to the predicted scores of a surrogate model and submits the highest-scoring ones. However, surrogate models tend to produce overly optimistic predictions in data-scarce regions, such that high predicted scores do not necessarily correspond to high-quality designs under the true objective. To mitigate this issue, existing methods improve optimization reliability by regularizing the surrogate, refining the training signal, or modifying the search procedure. For example, COMs (Trabucco et al., 2021) penalize overestimation on adversarially discovered designs, RoMA (Yu et al., 2021) enforces local smoothness around the current iterate, and IOM (Qi et al., 2022) aligns representations between offline data and optimized designs. Other methods focus on improving the training signal. BDI (Chen et al., 2022) performs bidirectional information distillation between the offline dataset and high-scoring designs, while ICT (Yuan et al., 2023) and Tri-Mentoring (Chen et al., 2023) leverage collaborative training among multiple proxies to suppress unreliable labels. In addition, Match-OPT (Hoang et al., 2024) aligns surrogate gradients with those of the true objective to improve the reliability of search directions, PGS (Chemingui et al., 2024) learns a search policy in place of a fixed gradient-ascent rule, and FGM (Grudzien et al., 2024) exploits functional graphical structure to guide optimization. Despite their different mechanisms, these methods ultimately rank candidate designs primarily by surrogate-predicted scores, leaving them susceptible to overly optimistic surrogate predictions.

A.2 Support-aware generation.

Support-aware generation constrains candidate generation to regions supported by the offline data, thereby avoiding the need to score arbitrary designs and reducing extrapolation risk in out-of-distribution regions. Representative approaches include CbAS (Brookes et al., 2019), which conditions an adaptive sampler on progressively stricter fitness thresholds, and MINs (Kumar & Levine, 2020), which invert the surrogate to map target values back to corresponding designs. More recent work further employs diffusion or sequence models to characterize the distribution of high-quality designs. DDOM (Krishnamoorthy et al., 2023) learns a conditional diffusion process over the design space; BONET (Mashkaria et al., 2023) formulates offline optimization as autoregressive trajectory generation; and GTG (Yun et al., 2024) uses a diffusion prior to guide trajectory synthesis. ROOT (Dao et al., 2025) recasts the problem as distributional translation through a probabilistic bridge, while SPADE (Yang et al., 2026) explicitly constrains candidate proximity to the data-supported region using kNN-based regularization. Although these methods alleviate severe out-of-distribution extrapolation by keeping generated designs close to the data manifold, support proximity alone does not guarantee that an individual candidate exhibits stable local behavior.

A.3 Ranking-based ordering.

Ranking-based ordering reformulates surrogate learning from absolute value prediction to relative preference modeling, as offline MBO ultimately selects only a Top-K set where preserving the ordering among candidates is more important than accurately calibrating objective values. RaM (Tan et al., 2025b) follows this idea by replacing pointwise regression with listwise ranking losses, and introduces the RaM-RankCosine and RaM-ListNet variants. DAR (Lyu et al., 2026) studies surrogate learnability from a ranking perspective and reduces the mismatch between the training distribution and near-optimal designs, while subsequent work further extends ranking surrogates toward task-agnostic representations (Tan et al., 2025a). Inspired by this perspective, we develop a KAN-based ranking surrogate that improves local order preservation through structured function modeling, providing a more reliable initial ordering for subsequent candidate selection.

A.4 POST-SEARCH DELIVERY.

Compared with the candidate search process, relatively little work has investigated the final screening and delivery of a finite candidate set after search completion. However, this stage is particularly relevant in machine-learning-guided biomolecular design, where uncertainty must be quantified before costly experimental validation (Fannjiang et al., 2022). A recent post-hoc approach further attaches a calibrated lower bound estimate to each candidate and advances only those satisfying a predefined target (Choi, 2026). To address this stage, we integrate surrogate preference, offline data support, and local prediction consistency as reliability evidence, and use these signals to reorder and repair candidate designs under a fixed evaluation budget.

B ADDITIONAL THEORETICAL ANALYSIS

This section presents a systematic theoretical analysis of the two stages of TRACER. For Stage I, we first establish pointwise order preservation and a support-to-error bridge (Theorem 1 and Proposition 2), and further characterize the locality and robustness of the ranking surrogate (Theorem 3 and Proposition 5). For Stage II, we then establish the theoretical properties of reliability-aware reranking and diagnosis-guided candidate repair (Theorem 2 and Propositions 1 and 8).

B.1 SUPPORT LOCALITY OF THE KAN RANKING SURROGATE

This subsection proves Theorems 1 and 3 and records two supporting properties of the spline parameterization. KANs and MLPs are both universal approximators (Wang et al., 2025; Zhang & Zhou, 2025), so no unconditional accuracy separation is available; the statements below concern support geometry, gradient interaction, and local estimation. Their premises are explicit, and Appendix B.1.4 records when they fail.

Setup. Consider a single edge of a KAN layer,

$$\phi(u) = w^{(b)} b(u) + w^{(s)} \sum_{k=1}^{G+p} c_k B_k(u) = w^{(b)} b(u) + w^{(s)} S(u), \qquad S(u) := \langle c, \mathbf{b}(u) \rangle, \tag{14}$$

where $\{B_k\}_{k=1}^{G+p}$ is the order-p B-spline basis on the fixed grid $g_0 < g_1 < \cdots < g_G$, $b(\cdot)$ is the residual base activation, and $c = (c_1, \ldots, c_{G+p})^\top$ collects the learnable spline coefficients. We write $\mathbf{b}(u) = [B_1(u), \ldots, B_{G+p}(u)]^\top$. All three parameter groups $(c, w^{(s)}, w^{(b)})$ are trainable unless stated otherwise.

B.1.1 PROOF OF THEOREM 1

For a finite candidate set $\mathcal{C} = \{x_j\}_{j=1}^m$ with oracle values $y_j = f(x_j)$, and a pointwise error bound $\varepsilon : \mathcal{C} \to \mathbb{R}_{\geq 0}$, define the pairwise reversal rate and the pair-dependent margin distribution

$$\begin{aligned} \mathcal{L}_{\text{pair}}(s_\theta; \mathcal{C}) &= \frac{2}{m(m-1)} \sum_{j<j'} \mathbb{1}\big[(s_\theta(x_j) - s_\theta(x_{j'}))(y_j - y_{j'}) < 0\big], \\ M_{\mathcal{C}}(\varepsilon) &= \frac{2}{m(m-1)} \sum_{j<j'} \mathbb{1}\big[|y_j - y_{j'}| \leq \varepsilon(x_j) + \varepsilon(x_{j'})\big]. \end{aligned} \tag{15}$$

Ties in either the surrogate difference or the oracle values are not counted as reversals, consistent with Definition 2 and the LPRA metric in Appendix D.1.

Proof of Theorem 1. Since $a > 0$, the calibrated score $\widetilde{s}_\theta = a s_\theta + b$ induces exactly the same ordering as s_θ, so it suffices to analyze $\widetilde{s}_\theta$. Take any pair $x, x' \in \mathcal{C}$ with, without loss of generality, $f(x) > f(x')$. If $f(x) - f(x') > \varepsilon(x) + \varepsilon(x')$, then by the triangle inequality

$$\begin{aligned} \widetilde{s}_\theta(x) - \widetilde{s}_\theta(x') \geq \big(f(x) - f(x')\big) &- \big|\widetilde{s}_\theta(x) - f(x)\big| \\ &- \big|\widetilde{s}_\theta(x') - f(x')\big| \;>\; \big(\varepsilon(x) + \varepsilon(x')\big) - \varepsilon(x) - \varepsilon(x') \;=\; 0, \end{aligned}$$

which gives Eq. (8). Hence a pair can be reversed only if its true margin satisfies $|f(x) - f(x')| \le \varepsilon(x) + \varepsilon(x')$. Averaging the corresponding indicator over all $m(m-1)/2$ pairs and comparing with Eq. (15) yields $\mathcal{L}_{\text{pair}}(s_\theta; \mathcal{C}) \le M_{\mathcal{C}}(\varepsilon)$. Note that no containment of $\mathcal{C}$ in $\Omega_\delta(\mathcal{D}_X)$ is required: leaving the supported region simply increases $\varepsilon(x)$ through Proposition 2, which is exactly the regime that Stage II is designed to flag. □

Proposition 2 (Support-to-error bridge). *Let $\widetilde{\mathcal{R}} \subseteq \mathcal{D}_X$ be the offline reference set used in Section 3.2, and write $d(x, \widetilde{\mathcal{R}}) := \min_{r \in \widetilde{\mathcal{R}}} d(x, r)$. Assume:* (i) *$f$ is L_f-Lipschitz with respect to d on $\mathcal{X}$;* (ii) *the calibrated surrogate has in-sample error $\max_{r \in \widetilde{\mathcal{R}}} |\widetilde{s}_\theta(r) - f(r)| \le \varepsilon_{\widetilde{\mathcal{R}}}$;* (iii) *$\widetilde{s}_\theta$ is L_s-Lipschitz with respect to d. Then for every $x \in \mathcal{X}$,*

$$|\widetilde{s}_\theta(x) - f(x)| \;\le\; \varepsilon(x) \;:=\; \varepsilon_{\widetilde{\mathcal{R}}} + (L_f + L_s)\, d(x, \widetilde{\mathcal{R}}). \tag{16}$$

In particular, on $\Omega_\delta(\mathcal{D}_X)$ with $\widetilde{\mathcal{R}} = \mathcal{D}_X$, $e_\infty \le \varepsilon_{\mathcal{D}} + (L_f + L_s)\,\delta$, and Theorem 1 gives $\mathcal{L}_{\text{pair}}(s_\theta; \mathcal{C}) \le M_{\mathcal{C}}\big(2\varepsilon_{\mathcal{D}} + 2(L_f + L_s)\delta\big)$. For discrete tasks, assumption (i) *is understood with respect to any extension of f to the continuous relaxation used by the surrogate, with L_f taken accordingly; only its existence is required, not its size.*

Proof. Fix $x \in \mathcal{X}$ and let $r^\star = \arg\min_{r \in \widetilde{\mathcal{R}}} d(x, r)$. Then

$$|\widetilde{s}_\theta(x) - f(x)| \le |\widetilde{s}_\theta(x) - \widetilde{s}_\theta(r^\star)| + |\widetilde{s}_\theta(r^\star) - f(r^\star)| + |f(r^\star) - f(x)| \le L_s\, d(x, r^\star) + \varepsilon_{\widetilde{\mathcal{R}}} + L_f\, d(x, r^\star),$$

which is Eq. (16). The Ω_δ statement follows because $d(x, \mathcal{D}_X) \le \delta$ there, and the rate bound substitutes the uniform $\varepsilon(\cdot)$ into $M_{\mathcal{C}}$. □

Remark 1 (Why Stage II uses endpoint-wise support). *Under Eq.* (16)*, the reversal-safe margin of a pair (x, x') is $\varepsilon(x) + \varepsilon(x') = 2\varepsilon_{\widetilde{\mathcal{R}}} + (L_f + L_s)\big(d(x, \widetilde{\mathcal{R}}) + d(x', \widetilde{\mathcal{R}})\big)$: the safety of a comparison degrades with the support distance of the* worse-supported *endpoint. In particular, for the common-support distance D_{ij}^{sup} of Eq.* (49)*, $d(x_i, \widetilde{\mathcal{R}}) \le D_{ij}^{\text{sup}}$ whenever $\widetilde{\mathcal{R}} = \mathcal{D}_X$, so $\varepsilon(x_i) + \varepsilon(x_j) \le 2\varepsilon_{\mathcal{D}} + 2(L_f + L_s) D_{ij}^{\text{sup}}$: the RQ1 support grouping is the empirical counterpart of the endpoint-wise safe margin, and the endpoint-wise evidence (F_i, F_j) in Section 3.2 plays the same role at delivery time.*

Remark 2 (Where the KAN analysis enters). *The constant L_s in Eq.* (16) *is controlled by the local spline coefficients: for the edge of Eq.* (14)*, at most $p + 1$ basis functions are active at any input (Lemma 1), so*

$$|S'(u)| \le (p+1) \max_{k \in \mathcal{A}(u)} |c_k| \cdot \max_k \|B_k'\|_\infty, \qquad |b(u) - b(u')| \le L_b\, |u - u'|,$$

and the Lipschitz constant of ϕ on a supported patch is governed by the coefficients active there. By Propositions 3 and 4, those coefficients are estimated from local samples only, and by Corollary 2, first-layer coefficient updates driven by non-local data leave them unchanged to first order. The constant entering $\varepsilon(x)$ on a supported patch is therefore itself locally determined. Two caveats keep the claim honest: L_f is unknown at decision time, and $\varepsilon_{\widetilde{\mathcal{R}}}$ is only estimable on held-out offline points. The result links support and error structurally rather than certifying individual pairs, consistent with the conditional framing of Section 3.1.2.

Theorem 3 (Three-channel first-order coupling). *Let $x, x' \in \mathcal{X}$ and consider one gradient step of size η on a loss L evaluated at x'. For edge (i, q) of one KAN layer, write $\ell'_{iq} := \partial L / \partial \phi_{iq}(x'_i)$. Holding all other parameters fixed, the first-order change of the edge response at x_i decomposes into three channels:*

$$\Delta\phi_{iq}(x_i) = -\eta\, \ell'_{iq} \Big[\underbrace{(w_{iq}^{(s)})^2 \big\langle \mathbf{b}(x_i), \mathbf{b}(x'_i) \big\rangle}_{\text{coefficient channel}} + \underbrace{S_{iq}(x_i)\, S_{iq}(x'_i)}_{w^{(s)}\text{ channel}} + \underbrace{b(x_i)\, b(x'_i)}_{w^{(b)}\text{ channel}} \Big]. \tag{17}$$

Summing over the edges of a layer, with $y_q(x) = \sum_{i=1}^d \phi_{iq}(x_i)$, $g_q := \partial L / \partial y_q(x')$, a common grid per input dimension, and a common spline scale $w^{(s)} := w_{iq}^{(s)}$,

$$\Delta y_q(x) = -\eta\, g_q \Big[(w^{(s)})^2 \sum_{i \notin \text{Sep}(x, x')} \big\langle \mathbf{b}(x_i), \mathbf{b}(x'_i) \big\rangle + \sum_{i=1}^d \rho_{iq}(x_i, x'_i) \Big], \tag{18}$$

where $\rho_{iq}(u,u') := S_{iq}(u)S_{iq}(u') + b(u)b(u')$ *and* $\mathrm{Sep}(x,x')$ *collects the coordinates whose grid cells are separated by more than* p *cells. Three properties follow. (i)* Exact locality of the coefficient channel. *By Lemma 1, every separated coordinate contributes exactly zero to the first sum; if* $\mathrm{Sep}(x,x') = \{1,\dots,d\}$*, the coefficient channel vanishes entirely. (ii)* The residual is rank-2 per edge and separation-independent. *Each* ρ_{iq} *is the Gram kernel of the two-dimensional feature map* $u \mapsto (S_{iq}(u), b(u))$*; it has the same functional form for every pair and is not governed by grid separation, since neither* S_{iq} *nor* b *has compact support tied to the grid. (iii)* Recovery of exact decoupling. *If* $w^{(s)}$ *and* $w^{(b)}$ *are frozen (or absorbed into* c*), the residual channels carry no update; under full separation the response at* x *is then unchanged to first order. The common-scale assumption is only cosmetic: with edge-specific scales, the coefficient channel reads* $\sum_i (w_{iq}^{(s)})^2 \langle \mathbf{b}(x_i), \mathbf{b}(x_i')\rangle$*, which still vanishes on every separated coordinate.*

B.1.2 Proof of Theorem 3

The proof rests on the support structure of the B-spline basis.

Lemma 1 (Support locality)**.** *Each* B_k *is supported on at most* $p+1$ *consecutive grid cells. For any input* u*, the active set* $\mathcal{A}(u) = \{k : B_k(u) \neq 0\}$ *satisfies* $|\mathcal{A}(u)| = p+1$*. If* u *and* u' *lie in cells whose indices differ by more than* p*, then* $\mathcal{A}(u) \cap \mathcal{A}(u') = \varnothing$ *and hence* $\langle \mathbf{b}(u), \mathbf{b}(u')\rangle = 0$*.*

Proof. Standard properties of the B-spline basis (de Boor, 2001): the order-p basis function B_k is supported on the knot interval $[g_{k-p-1}, g_k]$, which spans $p+1$ cells, and exactly $p+1$ basis functions are nonzero at any point of the domain. Two supports each spanning $p+1$ cells cannot intersect when the containing cells differ by more than p indices, so every product $B_k(u)B_k(u')$ vanishes and the inner product is zero. □

Proof of Theorem 3. Differentiating Eq. (14) with respect to each parameter group of edge (i,q) gives

$$\frac{\partial \phi_{iq}(u)}{\partial c_k} = w_{iq}^{(s)} B_k(u), \qquad \frac{\partial \phi_{iq}(u)}{\partial w_{iq}^{(s)}} = S_{iq}(u), \qquad \frac{\partial \phi_{iq}(u)}{\partial w_{iq}^{(b)}} = b(u).$$

A gradient step of size η driven by the loss at x' therefore induces

$$\Delta c_k = -\eta\, \ell'_{iq}\, w_{iq}^{(s)} B_k(x_i'), \qquad \Delta w_{iq}^{(s)} = -\eta\, \ell'_{iq}\, S_{iq}(x_i'), \qquad \Delta w_{iq}^{(b)} = -\eta\, \ell'_{iq}\, b(x_i').$$

The first-order change of the edge response at x_i is

$$\Delta \phi_{iq}(x_i) = \sum_k w_{iq}^{(s)} B_k(x_i)\, \Delta c_k + S_{iq}(x_i)\, \Delta w_{iq}^{(s)} + b(x_i)\, \Delta w_{iq}^{(b)},$$

and substituting the three updates yields Eq. (17). For the layer statement, note that $\ell'_{iq} = g_q$ is shared across the input coordinates of output unit q; summing Eq. (17) over i with a common $w^{(s)}$, splitting the coefficient sum according to $\mathrm{Sep}(x,x')$, and applying Lemma 1 to every $i \in \mathrm{Sep}(x,x')$ gives Eq. (18), since every summand $\langle \mathbf{b}(x_i), \mathbf{b}(x_i')\rangle$ with $i \in \mathrm{Sep}(x,x')$ is zero. Property (ii) follows from $\rho_{iq}(u,u') = \langle (S_{iq}(u), b(u)), (S_{iq}(u'), b(u'))\rangle$ being an inner product in $\mathbb{R}^2$. Property (iii) is immediate: frozen scales give $\Delta w_{iq}^{(s)} = \Delta w_{iq}^{(b)} = 0$. □

Remark 3 (Optimizer)**.** *The first-order reading isolates the component of the update driven by* x'*. Momentum or Adam-style preconditioning rescales each parameter component but does not create* x'*-dependence in the coefficient channel on separated coordinates: the* x'*-driven part of* Δc_k *remains proportional to* $B_k(x_i') = 0$*.*

For a globally supported layer, $h(x) = \sum_j v_j \sigma(w_j^\top x)$ gives $\partial h(x)/\partial w_j = v_j \sigma'(w_j^\top x)\, x$, and summing the inner products over j yields a single dense cross-input coupling term: there is no channel decomposition, and neither $\langle x, x'\rangle$ nor σ' is supported on a bounded region of the input space, so no coordinate can force a zero contribution.

Corollary 1 (Separation strictly reduces coupling)**.** *Because the B-spline basis is nonnegative,* $(w^{(s)})^2 \langle \mathbf{b}(x_i), \mathbf{b}(x_i')\rangle \geq 0$ *for every coordinate. Hence each additional separated coordinate removes exactly this quantified nonnegative amount from the layer coupling in Eq.* (18)*, while the*

residual channel $\sum_i \rho_{iq}(x_i, x'_i)$ is unchanged. Cross-region coupling is therefore a monotone decreasing function of the number of separated coordinates, up to a separation-independent rank-2 floor carried by the trainable scales.

Corollary 2 (Bulk updates reach the high-value fit only through the rank-2 channel)**.** *Let U be a high-value region and V a bulk region such that every pair $(x, x') \in U \times V$ is separated on all coordinates. Then for a loss restricted to V, $\partial L_V / \partial c_k = 0$ for every coefficient c_k active on U, and the first-order change of the response at $x \in U$ reduces to*

$$\Delta\phi(x) = -\eta\, \ell' \left[S(x)S(x') + b(x)b(x')\right] \qquad \textit{per edge},$$

which is independent of the grid distance between U and V. If $w^{(s)}$ and $w^{(b)}$ are frozen, bulk updates leave the fitted response on U unchanged to first order.

Proof. For $x' \in V$ and any coefficient c_k active on U, Lemma 1 gives $B_k(x'_i) = 0$ on every (separated) coordinate, so each per-sample coefficient gradient $\ell'\, w^{(s)} B_k(x'_i)$ vanishes. The restricted-loss gradient is a *linear* combination of these per-sample gradients and therefore also vanishes on every such c_k. The remaining update channels are $\Delta w^{(s)}$ and $\Delta w^{(b)}$, and substituting them into Eq. (17) gives the displayed expression. The frozen-scale case is property (iii) of Theorem 3. □

B.1.3 Supporting Properties

Two further consequences of compact support concern the constants entering a local error rather than its optimization dynamics.

Proposition 3 (Localized approximation error)**.** *Let $f \in C^{p+1}$ and let $I_t = [g_t, g_{t+1}]$ with $h_t = g_{t+1} - g_t$. Then*

$$\inf_c \left\| f - \sum_k c_k B_k \right\|_{L^\infty(I_t)} \le C_p\, h_t^{p+1} \left\| f^{(p+1)} \right\|_{L^\infty(\tilde{I}_t)}, \tag{19}$$

where $\tilde{I}_t$ is the union of cells covered by the basis functions active on I_t, and C_p depends only on p.

Proof. This is the standard local approximation bound for spline spaces (de Boor, 2001). By Lemma 1, only the coefficients active on I_t affect the restriction of the spline to I_t, so the bound involves the derivative norm on the local patch $\tilde{I}_t$ rather than on the whole domain. □

The error constant in Eq. (19) involves only a local derivative norm, whereas classical rates for globally supported approximators carry norms over the entire domain. When the objective varies sharply overall but is comparatively smooth inside a high-value neighborhood, the two constants differ.

Proposition 4 (Banded local estimation)**.** *Fix the outer weights of Eq.* (14) *and estimate the spline coefficients by least squares on samples $\{u_j\}$ with design matrix $\Phi_{jk} = B_k(u_j)$. Then $(\Phi^\top \Phi)_{kk'} = \sum_j B_k(u_j) B_{k'}(u_j) = 0$ whenever $|k - k'| > p$, so the normal equations are banded with bandwidth p and the coefficients active on a region are determined by the samples falling in the corresponding support patch.*

Proof. By Lemma 1, B_k and $B_{k'}$ have disjoint supports when $|k - k'| > p$, so every summand vanishes. □

Propositions 3 and 4 are two sides of the same structure. Local coefficients are *estimated from* local data and, by Corollary 2, coefficient updates driven by non-local data leave them unchanged to first order; only the shared rank-2 scale channel can perturb the fitted response, independently of grid distance.

B.1.4 Scope and Failure Modes

The results above give sufficient conditions, not an unconditional advantage. Six boundaries are relevant to the reported experiments.

(i) Expressivity. KANs and MLPs are both universal approximators. Established generalization bounds for KANs are governed by coefficient norms and activation Lipschitz constants rather than by width (Li et al., 2025). The analysis concerns inductive bias, optimization, and estimation, and makes no claim about representational capacity.

(ii) Grid resolution. Our configuration uses $G = 3$ and $p = 2$, giving $G + p = 5$ coefficients per edge with $p + 1 = 3$ active at any input. Exact per-coordinate separation is therefore rare at this setting. The operative statement is Corollary 1: the coefficient channel is exactly zero on separated coordinates, and each additional separated coordinate removes a quantified nonnegative amount $(w^{(s)})^2\langle \mathbf{b}(x_i), \mathbf{b}(x_i')\rangle$ from the total coupling, up to the separation-independent rank-2 floor of Theorem 3(ii). A finer grid increases the number of separated coordinates and should strengthen the effect, until local sample density becomes the binding constraint.

(iii) Discrete one-hot relaxation. Under one-hot relaxation the input is sparse. At each position a single coordinate is near one and the remainder are near zero. Two candidates therefore agree, near zero, on most coordinates, and those coordinates are not separated at any grid resolution. The residual coefficient-channel term in Eq. (18) consequently dominates, and refining the grid does not remedy it. This is a structural limitation of support locality on nearly discrete input distributions rather than an incidental result, and it is consistent with the smaller local-order differences observed on the sequence tasks in Figure 5.

(iv) Local sample sparsity. Proposition 4 cuts both ways. Localized estimation isolates a region from the bulk, but it also estimates the corresponding coefficients from fewer samples. When a high-value region contains few samples, variance dominates and globally shared parameters may be preferable.

(v) Trainable scales. Exact locality lives in the coefficient channel only. The scale parameters $w^{(s)}, w^{(b)}$ contribute a per-edge rank-2 residual that is identical in form for all pairs and does not decay with grid distance (Theorem 3(ii)). Freezing $w^{(s)}, w^{(b)}$ or absorbing them into c removes this channel and recovers exact first-order decoupling (Theorem 3(iii)); we keep them trainable, so the claims in this paper concern the coefficient channel and the monotone reduction of Corollary 1, not an exact zero of the full coupling.

(vi) Depth. Theorem 3 is a single-layer statement. For $L > 1$ layers, exact decoupling still holds for first-layer coefficient updates, since $\partial L/\partial c_k^{(1)} = \ell'\, w^{(s)} B_k(x_i') = 0$ for every $c_k^{(1)}$ active on a separated region, regardless of the downstream computation. Updates to deeper-layer coefficients and to all scale parameters, however, propagate through mixed activations and are generally nonzero. The result should therefore be read as characterizing the input layer's inductive bias; Appendix D.3.1 provides the capacity-controlled empirical counterpart.

B.2 Robustness of Rank-Defined Top-ρ Supervision

This subsection develops the robustness of rank-defined Top-ρ supervision to label-outlier magnitude. Labels determine the Top-ρ member set within each list, while their absolute numeric values do not enter the contrastive loss directly.

Proposition 5 (Rank-defined Top-ρ signals have bounded sensitivity to outlier magnitude)**.** *Fix a length-m list and its normalized representations, and let $k_\rho = |\mathcal{G}_b|$. If a label perturbation $y \mapsto y'$ does not change the Top-ρ set, then the within-list contrastive loss in Eq.* (4) *is unchanged. If the perturbation replaces only q Top-ρ members, then under $\|\mathbf{z}_i\|_2 = 1$, the loss change is controlled by q/k_ρ, τ, and m, and does not grow unboundedly with the outlier magnitude $|y_i' - y_i|$. By contrast, the MSE change under the same label perturbation contains the quadratic term $(y_i' - y_i)^2$.*

Proof of Proposition 5. First consider MSE.

$$\mathcal{L}_{\mathrm{mse}}(\mathbf{u}, \mathbf{y}) = \frac{1}{m}\sum_i (u_i - y_i)^2.$$

If the r-th label is perturbed to $y'_r = y_r + \Delta$ while all other labels are unchanged, then

$$\mathcal{L}_{\mathrm{mse}}(\mathbf{u}, \mathbf{y}') - \mathcal{L}_{\mathrm{mse}}(\mathbf{u}, \mathbf{y}) = \frac{1}{m}\left[\Delta^2 - 2\Delta(u_r - y_r)\right]. \tag{20}$$

Thus, as the outlier magnitude $|\Delta|$ increases, the MSE change contains a quadratic term Δ^2/m.

Next consider the Top-ρ contrastive regularizer. Fix the two-view normalized representations $\{\mathbf{z}_i\}_{i=1}^{2m}$ within a list. In the contrastive loss, labels $\mathbf{y}$ determine only the Top-ρ original-index set $\mathcal{G}$, which in turn determines the valid anchor set $\mathcal{A}$ and the positive set $\mathcal{P}(i)$. If the perturbation $\mathbf{y} \mapsto \mathbf{y}'$ does not change $\mathcal{G}$, then $\mathcal{A}$, all $\mathcal{P}(i)$, and the similarity matrix remain unchanged, so the within-list contrastive loss is exactly the same.

Now consider the case where the perturbation replaces q Top-ρ members. Let $k = |\mathcal{G}| = |\mathcal{G}'|$, and let the corresponding two-view anchor sets be $\mathcal{A}$ and $\mathcal{A}'$. Then $|\mathcal{A}| = |\mathcal{A}'| = 2k$ and $|\mathcal{A} \setminus \mathcal{A}'| = |\mathcal{A}' \setminus \mathcal{A}| = 2q$. Since $\|\mathbf{z}_i\|_2 = 1$, the similarity terms are bounded in $[-1/\tau, 1/\tau]$, and each anchor loss is bounded by $C_{\tau,m} := \log(2m-1) + 2/\tau$. For anchors in $\mathcal{A} \cap \mathcal{A}'$, the change comes only from the positive-set average and is at most $4q/((2k-1)\tau)$. For anchors that appear only in $\mathcal{A}$ or only in $\mathcal{A}'$, their contribution is controlled by $C_{\tau,m}$. Therefore, the within-list contrastive losses induced by the two Top-ρ sets satisfy

$$|\mathcal{L}_{\mathrm{ctr}}(\mathcal{G}) - \mathcal{L}_{\mathrm{ctr}}(\mathcal{G}')| \le \frac{4q}{(2k-1)\tau} + \frac{2q}{k} C_{\tau,m}. \tag{21}$$

The right-hand side depends only on the fraction of replaced Top-ρ members, the temperature, and the list length; it does not contain the outlier magnitude $|\Delta|$. This shows that the rank-defined Top-ρ signal has bounded sensitivity to label-outlier magnitude. □

B.3 Top-ρ Contrastive Geometry and Stability

This subsection characterizes the group-level constraint imposed by the Top-ρ contrastive regularizer on high-value representation geometry within a listwise batch. It then transfers this average geometry to average prediction consistency under explicit conditions on the prediction head and representation norms.

Anchor loss and mean positive geometry. Fix a list b and a valid anchor $i \in \mathcal{A}_b$. To simplify notation, we omit the subscript b, write the positive set as $\mathcal{P}(i)$, and write the set of all representation indices except the anchor as $\mathcal{N}(i) = \{j : j \neq i\}$. Let $P_i = |\mathcal{P}(i)|$ and

$$Z_i = \sum_{j \in \mathcal{N}(i)} \exp\left(\frac{\mathbf{z}_i^\top \mathbf{z}_j}{\tau}\right). \tag{22}$$

Following Eq. (4), the anchor loss is

$$\ell(i) = -\frac{1}{P_i} \sum_{p \in \mathcal{P}(i)} \left[\frac{\mathbf{z}_i^\top \mathbf{z}_p}{\tau} - \log Z_i\right]. \tag{23}$$

Proposition 6 (Small anchor loss controls mean positive geometry)**.** *Assume $\|\mathbf{z}_j\|_2 = 1$ for the anchor and all comparison representations. If $\ell(i) \le \epsilon$, then*

$$\frac{1}{P_i} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(\mathbf{z}_i^\top \mathbf{z}_p/\tau)}{Z_i} \ge -\epsilon, \tag{24}$$

and the mean squared distance to the positive set satisfies

$$D_i^+ := \frac{1}{P_i} \sum_{p \in \mathcal{P}(i)} \|\mathbf{z}_i - \mathbf{z}_p\|_2^2 \le \underbrace{2 - 2\tau \log Z_i + 2\tau\epsilon}_{\Delta_i(\epsilon)}. \tag{25}$$

Proof. Equation (23) is the negative average log-softmax probability assigned to the positives, so $\ell(i) \le \epsilon$ immediately gives Eq. (24). The same equation also gives $P_i^{-1} \sum_{p \in \mathcal{P}(i)} \mathbf{z}_i^\top \mathbf{z}_p = \tau(\log Z_i - \ell(i)) \ge \tau(\log Z_i - \epsilon)$. For normalized representations, $\|\mathbf{z}_i - \mathbf{z}_p\|_2^2 = 2 - 2\mathbf{z}_i^\top \mathbf{z}_p$. Averaging this identity over the positive set yields Eq. (25). □

Gradient decomposition of Top-ρ SupCon. Let p_{ij} be the softmax weight on $\mathcal{N}(i)$ induced by $\exp(\mathbf{z}_i^\top \mathbf{z}_j/\tau)$, and define $\bar{\mathbf{z}}_i^+ = P_i^{-1} \sum_{p\in\mathcal{P}(i)} \mathbf{z}_p$ and $\bar{\mathbf{z}}_i^{\text{all}} = \sum_{j\in\mathcal{N}(i)} p_{ij}\mathbf{z}_j$.

Proposition 7 (Attraction and repulsion gradient of Top-ρ SupCon)**.** *For the normalized anchor representation $\mathbf{z}_i$, the gradient of the anchor loss above is*

$$\frac{\partial \ell(i)}{\partial \mathbf{z}_i} = \frac{1}{\tau}\left(\bar{\mathbf{z}}_i^{\text{all}} - \bar{\mathbf{z}}_i^+\right). \tag{26}$$

Proof. Decompose the anchor loss into the positive-average term and the log-sum-exp normalization term. The gradient of the first term with respect to $\mathbf{z}_i$ is $-\bar{\mathbf{z}}_i^+/\tau$, while standard differentiation of the log-sum-exp term gives $\bar{\mathbf{z}}_i^{\text{all}}/\tau$. Adding the two terms proves the proposition. □

Proposition 7 shows that the negative-gradient direction is $\tau^{-1}(\bar{\mathbf{z}}_i^+ - \bar{\mathbf{z}}_i^{\text{all}})$. Thus optimization pulls valid anchors toward the Top-ρ positive center while reducing their relative attraction to non-positive samples, especially high-similarity negatives emphasized by the softmax weights. This decomposition explains why the Top-ρ contrastive regularizer can tighten local high-value representation geometry without changing the RankCosine main objective.

From mean positive geometry to average prediction consistency. Let v_j denote the input view associated with representation index j, and write $\mathbf{h}_j = \Phi_\theta(v_j) = r_j\mathbf{z}_j$, where $r_j = \|\mathbf{h}_j\|_2 > 0$. Suppose the surrogate decomposes as $F_\theta = g \circ \Phi_\theta$.

Corollary 3 (Conditional average prediction stability)**.** *Assume g is L_g-Lipschitz in the relevant high-value neighborhood and $r_j \le R$ for the anchor and its positives. Define the mean norm mismatch $\delta_{r,i} = P_i^{-1}\sum_{p\in\mathcal{P}(i)} |r_i - r_p|$. If $\ell(i) \le \epsilon$, then*

$$\frac{1}{P_i}\sum_{p\in\mathcal{P}(i)} |F_\theta(v_i) - F_\theta(v_p)| \le L_g\left[R\sqrt{\Delta_i(\epsilon)} + \delta_{r,i}\right]. \tag{27}$$

Proof. For each positive p, $\|\mathbf{h}_i - \mathbf{h}_p\|_2 \le r_i\|\mathbf{z}_i - \mathbf{z}_p\|_2 + |r_i - r_p| \le R\|\mathbf{z}_i - \mathbf{z}_p\|_2 + |r_i - r_p|$. Apply the Lipschitz property of g, average over p, and use Cauchy–Schwarz to bound the mean normalized distance by $\sqrt{D_i^+}$. Proposition 6 then gives $D_i^+ \le \Delta_i(\epsilon)$. □

Proposition 6 and Corollary 3 control averages over the Top-ρ positive set. They do not provide an individual similarity lower bound for every clean and perturbed pair. One low-similarity paired view may be compensated by other high-similarity positives. The result therefore supports group-level representation consistency rather than a certificate for every perturbation.

B.4 Reliability-Aware Reranking and Diagnosis-Guided Repair

This subsection proves Theorem 2 and gives an additional suppression bound for the multiplicative gate.

Proof of Theorem 2. For two candidates in the same hard-pass tier with $B_i > B_j > 0$,

$$\text{Final}_j > \text{Final}_i \iff B_jG_j > B_iG_i \iff \frac{G_j}{G_i} > \frac{B_i}{B_j},$$

because $G_i, G_j > 0$ for $f_0 \in (0,1)$. This proves Eq. (11). In addition, $G_k = f_0 + (1-f_0)\sqrt{F_kS_k}$ and $F_k, S_k \in [0,1]$ imply $G_k \in [f_0, 1]$. Hence

$$\frac{G_j}{G_i} \le \frac{1}{f_0}.$$

If $B_i/B_j \ge 1/f_0$, the strict inequality required by Eq. (11) cannot hold, proving Eq. (12). □

Proposition 8 (Support and consistency gate suppression)**.** *For $f_0 \in (0,1)$,*

$$B_i - \mathrm{Final}_i = B_i(1-f_0)\big(1-\sqrt{F_iS_i}\big). \tag{28}$$

If $F_iS_i \le \eta$ for some $\eta \in [0,1]$, then

$$\mathrm{Final}_i \le B_i\big[f_0 + (1-f_0)\sqrt{\eta}\big]. \tag{29}$$

Proof. By Eq. (10), $\mathrm{Final}_i = B_i[f_0 + (1-f_0)\sqrt{F_iS_i}]$. Subtracting this quantity from B_i gives Eq. (28). If $F_iS_i \le \eta$, monotonicity of the square root and $B_i \ge 0$ give Eq. (29). □

Proof of Proposition 1. Since $r_i^\star$ is nearest to x_i, $d_{\widetilde{\mathcal{R}}}(x_i) = \|x_i - r_i^\star\|_2$. The linear update gives

$$x_i^\star - r_i^\star = (1-\alpha_{\mathrm{pb}})(x_i - r_i^\star).$$

Although another reference point may become nearer after the update, $r_i^\star$ remains an admissible reference. Hence

$$d_{\widetilde{\mathcal{R}}}(x_i^\star) \le \|x_i^\star - r_i^\star\|_2 = (1-\alpha_{\mathrm{pb}})d_{\widetilde{\mathcal{R}}}(x_i).$$

For the budget, Eq. (13) maps each selected candidate to exactly one delivered design and adds or removes none, so $|\mathcal{X}^\star| = |\widehat{\mathcal{X}}| = K_{\mathrm{out}}$. □

Proposition 9 (Rule consequents respect the stated prototype order)**.** *For the explicitly comparable fuzzy-rule prototypes in Table 2, the fixed consequents satisfy*

$$\begin{aligned} &w_1 > w_2, \quad w_1 > w_3, \qquad\quad w_2 > w_6, \\ &w_3 > w_7, \quad w_4 > w_5 > w_8. \end{aligned} \tag{30}$$

Thus the listed prototype comparisons do not assign a lower consequent when support or local consistency is improved while the compared quality level is fixed.

Proof. The fixed consequent vector in Appendix C.3 gives $w_1 = 0.95$, $w_2 = 0.86$, $w_3 = 0.82$, $w_4 = 0.74$, $w_5 = 0.58$, $w_6 = 0.46$, $w_7 = 0.38$, and $w_8 = 0.34$. Direct substitution verifies every inequality in Eq. (30). These comparisons correspond respectively to the explicit high-quality and mid-quality rule prototypes whose support or stability antecedent is weakened. □

Proposition 9 is a finite audit of the prespecified prototype consequents. It does not imply that the continuous defuzzified map $T(Q,F,S)$ is globally coordinatewise monotone: overlapping memberships, wildcard antecedents, and the fallback branch can change which rules contribute. We therefore use the rule table only as a fixed qualitative action policy and make no end-to-end monotonicity or calibration claim for T.

C Additional Method Details

In this section, we present the methodological details of TRACER. We first introduce the overall framework and then elaborate on its two main components: Context-Relative Candidate Search and Fuzzy Rules for Candidate Diagnosis.

C.1 Detailed Framework Overview

Figure 4 provides the complete implementation view of TRACER. The main paper describes the two stages in prose because the detailed training, search, diagnosis, reranking, and repair operations are described separately in Sections 3.1 and 3.2.

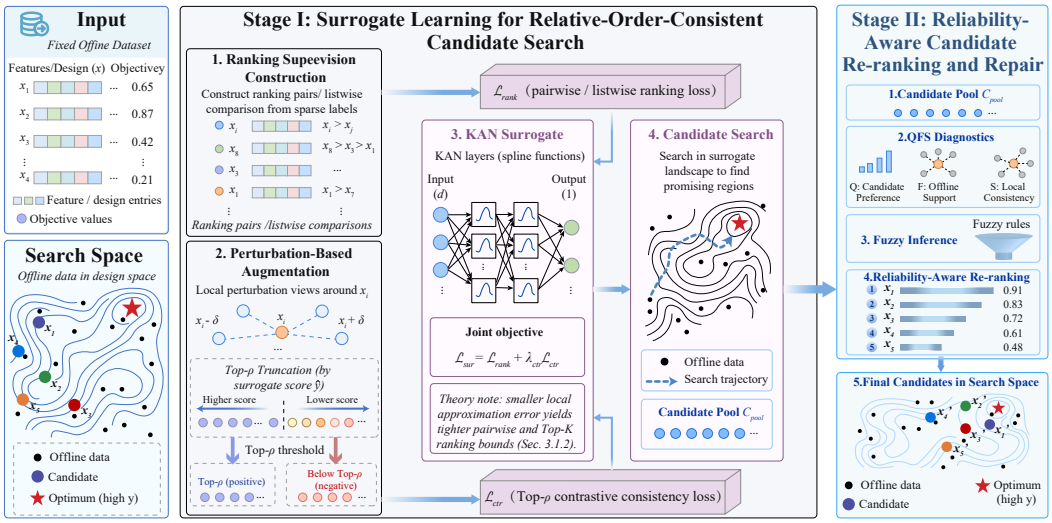


Figure 4: Detailed framework of TRACER. Stage I improves local order preservation through ranking supervision, Top-ρ contrastive regularization, and KAN-guided search. Stage II uses predicted preference Q, pool-relative offline support F, and local consistency S as candidate evidence for pairwise ranking reliability, followed by fuzzy diagnosis, reliability-aware reranking, and support repair.

C.2 Context-Relative Candidate Search

The initial set $X^{(0)} = \{x_i^{(0)}\}_{i=1}^{B_{\text{seed}}}$ contains the $B_{\text{seed}} = K_{\text{out}} = 128$ highest-label offline designs. At iteration t, candidate i is paired with n_c independently sampled context sets $\mathcal{R}_{i,t,c} = \{r_{i,t,c,j}\}_{j=1}^{m_c-1}$ from $\tilde{\mathcal{R}}$. Here $m_c = 12$ counts the candidate plus 11 reference samples in each small list. The implementation minimizes the negative mean sigmoid win probability over all current candidates and their sampled contexts,

$$\mathcal{L}_{\text{win}}^{(t)} = -\frac{1}{B_{\text{seed}} n_c (m_c - 1)} \sum_{i=1}^{B_{\text{seed}}} \sum_{c=1}^{n_c} \sum_{j=1}^{m_c-1} \sigma\Big(s(x_i^{(t)}) - s(r_{i,t,c,j})\Big), \tag{31}$$

where $\sigma(\cdot)$ is the sigmoid function. Thus $n_c = 5$ means five independently sampled context sets per candidate and iteration, not five restarts. Instead of maximizing a single-point score, Eq. (31) asks whether each candidate wins on average against sampled reference contexts. Stage I search uses Adam with the linear learning-rate curriculum $\eta_t = \eta_1[0.1 + 0.9t/(T_1 - 1)]$, followed by radius projection,

$$x_i^{(t+1)} = \Pi_{\mathcal{X} \cap \mathcal{B}_R(x_i^{(0)})}\Big(\text{AdamStep}\Big[x_i^{(t)}, \nabla_{x_i} \mathcal{L}_{\text{win}}^{(t)}, \eta_t\Big]\Big), \tag{32}$$

where $\mathcal{B}_R(x_i^{(0)})$ is the radius-R ball centered at the seed. The refinement stage selects the top $K_2 = 8$ candidates from $X^{(1)}$ by $s(x)$ and performs $T_2 = T_1$ direct Adam-ascent steps with $\eta_2 = \eta_1$, without the Stage I context loss or radius constraint.

$$x^{(t+1)} = \text{AdamAscent}\Big[x^{(t)}, \nabla_x s(x^{(t)}), \eta_2\Big]. \tag{33}$$

The implementation then concatenates $X^{(1)}$ and $X^{(2)}$ without duplicate removal. Candidate-range clipping is applied after the Stage II repair step.

C.3 Fuzzy Rules for Candidate Diagnosis

This subsection specifies the singleton-consequent policy used for soft candidate-state diagnosis in Section 3.2.1. Equation (45) converts rule activation into a delivery-readiness score. The score encodes a fixed action preference rather than a calibrated probability of candidate correctness. For each candidate $x_i \in \mathcal{C}^{\text{pool}}$, we characterize its reliability from three complementary perspectives: predicted potential, offline support, and local consistency.

Table 2: Fixed fuzzy rules for candidate diagnosis. A dash marks an unused antecedent, and each consequent encodes the delivery priority associated with that evidence pattern.

r	Q	F	S	w_r
1	high	good	stable	w_1
2	high	ok	stable	w_2
3	high	good	normal	w_3
4	mid	good	stable	w_4
5	mid	ok	normal	w_5
6	high	poor	–	w_6
7	high	–	unstable	w_7
8	mid	poor	–	w_8
9	low	good	–	w_9
10	low	–	unstable	w_{10}

Definition 3 (Candidate Preference)**.** *For each candidate $x_i \in \mathcal{C}^{\mathrm{pool}}$, we define the candidate-preference score as.*

$$Q_i = \frac{s(x_i) - \min_j s(x_j)}{\max_j s(x_j) - \min_j s(x_j) + \varepsilon}, \tag{34}$$

where $s(x_i)$ denotes the surrogate ranking score. A larger Q_i indicates stronger predicted potential relative to other candidates in the pool.

Definition 4 (Offline Support)**.** *Let $d_i = \min_{r \in \tilde{\mathcal{R}}} \|x_i - r\|_2$ denote the nearest-neighbor distance from x_i to the offline reference set $\tilde{\mathcal{R}}$. We define the offline-support score as.*

$$F_i = 1 - \frac{d_i - \min_j d_j}{\max_j d_j - \min_j d_j + \varepsilon}. \tag{35}$$

A larger F_i indicates stronger support from the observed offline data, whereas a smaller value suggests weaker offline support.

Definition 5 (Local Consistency)**.** *For each candidate x_i, we apply M local perturbations and measure the variation of the resulting surrogate predictions.*

$$\varsigma_i = \mathrm{std}_m \left[s(x_i + \xi_m) \right], \qquad \xi_m \sim \mathcal{N}(0, \sigma_n^2 I). \tag{36}$$

We define the local-consistency score as

$$S_i = 1 - \frac{\varsigma_i - \min_j \varsigma_j}{\max_j \varsigma_j - \min_j \varsigma_j + \varepsilon}. \tag{37}$$

A larger S_i indicates more stable surrogate predictions under local perturbations, whereas a smaller value reflects higher local sensitivity.

Membership functions. For any normalized $X \in \{Q, F, S\}$, the membership functions for the three linguistic values (low, mid, and high) are

$$\mu_{X,\mathrm{low}}(x) = \mathrm{trapmf}(x;\, 0,\, 0,\, 0.20,\, 0.45), \tag{38}$$

$$\mu_{X,\mathrm{mid}}(x) = \mathrm{trimf}(x;\, 0.20,\, 0.50,\, 0.80), \tag{39}$$

$$\mu_{X,\mathrm{high}}(x) = \mathrm{trapmf}(x;\, 0.55,\, 0.75,\, 1,\, 1), \tag{40}$$

where $\mathrm{trimf}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$ and $\mathrm{trapmf}(x; a, b, c, d) = \max(\min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0)$ denote triangular and trapezoidal membership functions, with the usual left/right shoulder convention when adjacent parameters are equal. For F, low/mid/high correspond to poor/ok/good; for S, they correspond to unstable/normal/stable. The three dimensions share the same set to keep scales consistent.

Rule consequents. The consequents $\{w_r\}$ of the $K_{\mathrm{rule}} = 10$ rules in Table 2 are fixed across tasks as $(w_1, \ldots, w_{10}) = (0.95,\, 0.86,\, 0.82,\, 0.74,\, 0.58, 0.46,\, 0.38,\, 0.34,\, 0.32,\, 0.16)$. Rules $r = 1$ to 3 represent delivery-ready patterns. Rules $r = 4$ to 5 represent borderline combinations with intermediate evidence. Rules $r = 6$ to 7 diagnose potentially overestimated candidates whose

high quality conflicts with poor support or unstable behavior. Rule $r = 8$ represents an evidence-limited borderline case, while rules $r = 9$ to 10 represent low-priority patterns. The rule table is intentionally sparse, so a no-rule region can occur. The implemented fallback is the deterministic quality-only branch

$$T_i^{\mathrm{fb}} = 0.15\,\mu_{Q,\mathrm{low}}(Q_i) + 0.50\,\mu_{Q,\mathrm{mid}}(Q_i) + 0.85\,\mu_{Q,\mathrm{high}}(Q_i), \tag{41}$$

followed by clipping to $[0, 1]$. It depends only on the current candidate's already-defined quality memberships and therefore introduces neither a circular pool median nor learned labels.

The three signals provide complementary evidence for candidate diagnosis: Q_i measures predicted potential, F_i characterizes offline support, and S_i reflects local prediction stability. However, none of them alone is sufficient for reliable candidate selection. For example, a candidate with high Q_i but low F_i may result from unsupported extrapolation, while a high-Q_i/low-S_i candidate may correspond to a locally unstable surrogate peak. Conversely, strong support or stability alone does not imply high optimization potential.

To jointly model these complementary evidence patterns, we introduce a fuzzy rule-based candidate diagnosis mechanism. Specifically, we describe Q_i by $\{\mathrm{low}, \mathrm{mid}, \mathrm{high}\}$, F_i by $\{\mathrm{poor}, \mathrm{ok}, \mathrm{good}\}$, and S_i by $\{\mathrm{unstable}, \mathrm{normal}, \mathrm{stable}\}$. Different combinations of these linguistic states characterize different candidate conditions.

Definition 6 (Fuzzy Rule-based Candidate Diagnosis)**.** *Given the complementary evidence (Q_i, F_i, S_i) of candidate x_i, its delivery priority is determined by a set of fuzzy rules*

$$\mathcal{R}_r : (Q_i, F_i, S_i) \to w_r, \tag{42}$$

where $w_r \in [0, 1]$ is a rule-specific hyperparameter representing the delivery priority associated with the corresponding evidence pattern.

As shown in Tab. 2, representative rules include.

$$\begin{aligned}
&(\mathrm{high}, \mathrm{good}, \mathrm{stable}) \to w_{\mathrm{HGS}},\\
&(\mathrm{high}, \mathrm{ok}, \mathrm{stable}) \to w_{\mathrm{HOS}},\\
&(\mathrm{high}, \mathrm{poor}, -) \to w_{\mathrm{HP}},\\
&(\mathrm{high}, -, \mathrm{unstable}) \to w_{\mathrm{HU}},\\
&(\mathrm{low}, -, \mathrm{unstable}) \to w_{\mathrm{LU}},
\end{aligned} \tag{43}$$

where a blank entry indicates that the corresponding signal is not considered by the rule. The rule priorities follow the reliability implied by their evidence patterns, such that$w_{\mathrm{HGS}} > w_{\mathrm{HOS}} > w_{\mathrm{HP}}, w_{\mathrm{HGS}} > w_{\mathrm{HU}} > w_{\mathrm{LU}}$. Since Q_i, F_i, and S_i are continuous, a candidate may simultaneously activate multiple fuzzy rules. Let $\mu_{X,\ell}(\cdot)$ denote the membership function of linguistic state ℓ for signal $X \in \{Q, F, S\}$. For each rule r, its activation strength is defined as

$$a_{r,i} = \min_{X \in \mathcal{P}_r} \mu_{X,\ell_r^X}(X_i), \tag{44}$$

where $\mathcal{P}_r$ denotes the set of signals involved in rule r, and ℓ_r^X is the linguistic state assigned to signal X by that rule. We then aggregate all activated rules through a zero-order Sugeno-style weighted average.

$$T_i = \frac{\sum_{r=1}^{K_{\mathrm{rule}}} a_{r,i} w_r}{\sum_{r=1}^{K_{\mathrm{rule}}} a_{r,i} + \varepsilon}. \tag{45}$$

Here, K_{rule} denotes the number of fuzzy rules and w_r is the delivery-priority hyperparameter associated with rule r. The resulting T_i summarizes the joint evidence from Q_i, F_i, and S_i into a continuous delivery-priority score.

D Additional Experimental Details and Results

This section provides supplementary experimental details and results for TRACER. We first describe the experimental settings, then analyze pairwise ranking reliability through diagnostic and controlled experiments, and finally report additional results that complement the main-text evaluation.

D.1 Experimental Settings

D.1.1 Evaluation Metrics.

The normalized score nScore_q measures percentile candidate quality on a task-independent scale, with higher values indicating better solutions (Trabucco et al., 2022). Let Score_q be the qth percentile of candidate oracle values and $y_{\min}, y_{\max}$ be the task-specific Design-Bench normalization constants.

$$\mathrm{nScore}_q = \frac{\mathrm{Score}_q - y_{\min}}{y_{\max} - y_{\min}}, \quad q \in \{50, 100\} \tag{46}$$

LPRA measures the fraction of local candidate pairs without order reversal (Kendall, 1938), while the Spearman correlation measures agreement between the complete surrogate and oracle rank vectors (Spearman, 1904). For a K-item neighborhood, let $\widehat{y}_i, y_i$ denote surrogate and oracle scores and let $\widehat{r}_i, r_i$ denote their ranks.

$$\begin{aligned} \mathrm{LPRA} &= 1 - \frac{2}{K(K-1)} \sum_{i<j} \mathbf{1}[(\widehat{y}_i - \widehat{y}_j)(y_i - y_j) < 0], \\ \mathrm{Spearman} &= \frac{\sum_i (\widehat{r}_i - \bar{\widehat{r}})(r_i - \bar{r})}{\sqrt{\sum_i (\widehat{r}_i - \bar{\widehat{r}})^2} \sqrt{\sum_i (r_i - \bar{r})^2}}. \end{aligned} \tag{47}$$

Directional stability measures the alignment of locally perturbed gradients (Sitawarin et al., 2022; Joo et al., 2023), while magnitude stability is the complement of gradient-norm CV and measures consistency in gradient strength (Löffler, 2025); both measures are higher-is-better. Given $g_j = \nabla_x F_\theta(x + \xi_j)$ and $\bar{g} = J^{-1} \sum_{j=1}^{J} g_j$, we compute

$$\begin{aligned} S_{\mathrm{dir}} &= \frac{1}{J} \sum_{j=1}^{J} \frac{g_j^\top \bar{g}}{\|g_j\|_2 \|\bar{g}\|_2 + \epsilon}, \\ S_{\mathrm{mag}} &= 1 - \frac{\mathrm{Std}_j(\|g_j\|_2)}{\mathrm{Mean}_j(\|g_j\|_2) + \epsilon}. \end{aligned} \tag{48}$$

D.1.2 Hyperparameter Configuration

As shown in Table 3, we report the complete hyperparameter configurations used for each dataset.

Table 3: Fixed TRACER hyperparameters used across Design-Bench tasks. All settings are shared unless explicitly marked by task type, and only the input dimension d changes across datasets.

Component	Hyperparameter	Value / Setting
Listwise data	list length m; number of lists S; lists/minibatch B_{list}	$m = 1000$; $S = 5000$; $B_{\mathrm{list}} = 128$
KAN surrogate	width; grid; B-spline order	$[d, [32, 0], [16, 0], 1]$; grid $= 3$; order $p = 2$
Training	optimizer; learning rate; training steps	Adam; 3×10^{-4}; 100 steps
Ranking objective	main loss	RankCosine
Contrastive regularizer	w_{ctr}; ρ; τ; view noise σ_v	0.1; 0.2; 0.1; 0.01 (unprojected view)
Candidate initialization	seeds B_{seed}; pool assembly	top-label $B_{\mathrm{seed}} = 128$; concatenate $128 + 8 = 136$ without deduplication
Stage 1 search	T_1; base η_1; contexts n_c; small-list size m_c; radius R	continuous: $T_1 = 80$, $\eta_1 = 10^{-3}$; discrete: $T_1 = 100$, $\eta_1 = 10^{-1}$; 5 contexts; 12 items (candidate + 11); 0.5
Stage 2 refinement	refined candidates K_2; step/lr scale	8; $T_2 = T_1$, $\eta_2 = \eta_1$; no radius constraint
Output budget	submitted candidates K_{out}	128
Support audit	reference subset size N_{ref}; distance batch size	4096; 256
Stability audit	noise samples M; noise scale σ_n	8; 0.03 (unprojected relaxation-space perturbation)
Reranking priority	α, β, f_0	0.40, 0.60, 0.25
Hard-pass thresholds	τ_F, τ_S, τ_T	0.55, 0.60, 0.28
Support-domain repair	clip margin η_c; pullback ratio α_{pb}	0.02; 0.35

D.2 Pairwise Ranking Reliability Diagnostics

RQ1 and RQ3 use oracle values only for post-hoc evaluation.

$$D_{ij}^{\mathrm{sup}} := \min_{x_r \in \mathcal{D}_X} \max\{d(x_i, x_r),\, d(x_j, x_r)\}, \tag{49}$$

For RQ1, candidate pairs are grouped by the common-support distance in Eq. (49). A reversal occurs when $(s_i - s_j)(f_i - f_j) < 0$. For RQ3, the Stage I candidate pool is fixed and each pair is assigned one of four transitions according to whether its ordering is correct before and after Stage II. These diagnostics do not alter training, search, reranking, or repair.

Margin-controlled analysis. Pairs with small surrogate margins may be harder to order. We therefore define $M_{ij} = |s(x_i) - s(x_j)|$ and repeat the RQ1 analysis within small, medium, and large margin groups formed by task-wise quantiles. This separates the effect of common offline support from the magnitude of the predicted preference.

Support-conditioned Stage II analysis. The RQ3 transition analysis uses the same Q_1 to Q_5 support groups as RQ1. We report the proportions of Wrong→Correct and Correct→Wrong transitions in each group. The comparison measures whether Stage II corrects more orderings than it damages and whether its intervention is concentrated where RQ1 identifies greater ranking risk.

D.3 Controlled Diagnostic Experiments

D.3.1 Capacity-Controlled Local-Order Diagnostic

The comparison in Figure 5 uses a KAN with input dimension d, additive hidden widths 32 and 16, and one scalar output. Its grid size is $G = 3$, and its B-spline order is $p = 2$. The symbolic edge masks remain zero, so the symbolic affine tensors do not participate in the numerical forward pass. We therefore match the MLP to the active numerical branch. Each active KAN edge contains $G + p = 5$ spline coefficients and two learned scales, giving

$$P_{\mathrm{KAN}}^{\mathrm{num}} = (G + p + 2)(32d + 32 \times 16 + 16). \tag{50}$$

The resulting KAN counts are 17,136, 22,960, 9,072, and 10,416 for Ant, Superconductor, TF-Bind-8, and TF-Bind-10, respectively. The two-hidden-layer MLP widths are $[103, 103]$, $[113, 113]$, $[83, 83]$, and $[87, 87]$. Their trainable parameter counts are 17,099, 22,827, 9,131, and 10,441, so every relative mismatch is below 0.7%.

Both surrogates use the same training data, optimizer, objective, and random seeds. Evaluation uses the same oracle-score top-20% anchors and $K_{\mathrm{NN}} = 16$ neighborhoods. Continuous tasks use Euclidean neighbors, whereas discrete tasks use Hamming neighbors. Each reported bar is the mean over eight seeds. Tied surrogate differences or oracle values are not counted as pairwise reversals.

D.3.2 Diagnostic-Policy Ablation Protocol

For each task and seed, a single trained surrogate and one two-stage search produce the shared candidate pool. We compute the normalized diagnostic signals (Q_i, F_i, S_i) once on that pool and then replace only the delivery map. Score-only ranks candidates by the raw surrogate score, which is order-equivalent to Q_i. The linear and product controls use

$$A_i^{\mathrm{lin}} = 0.40Q_i + 0.30F_i + 0.30S_i, \qquad A_i^{\mathrm{prod}} = Q_i F_i S_i, \tag{51}$$

respectively, and select the top $K_{\mathrm{out}} = 128$ candidates by the corresponding score.

Hard filter first forms $H_i = \mathbb{I}[F_i \geq 0.55]\mathbb{I}[S_i \geq 0.60]\mathbb{I}[T_i \geq 0.28]$. Passing candidates are placed before failing candidates, and each tier is ordered by the surrogate score. If fewer than 128 candidates pass, the remaining slots are filled by the highest-scoring failing candidates, so the output budget remains fixed. Fuzzy applies the complete reliability-gated reranking in Section 3.2.1: it computes T_i, combines Q_i and T_i with weights 0.40 and 0.60, applies the $0.25 + 0.75\sqrt{F_i S_i}$ gate, and orders candidates lexicographically by the hard-pass state and the gated score.

Fuzzy + PB uses exactly the same selected set and order as Fuzzy before repair. A selected candidate is flagged when $F_i < 0.55$ or $S_i < 0.60$ and is moved 35% toward its nearest offline reference

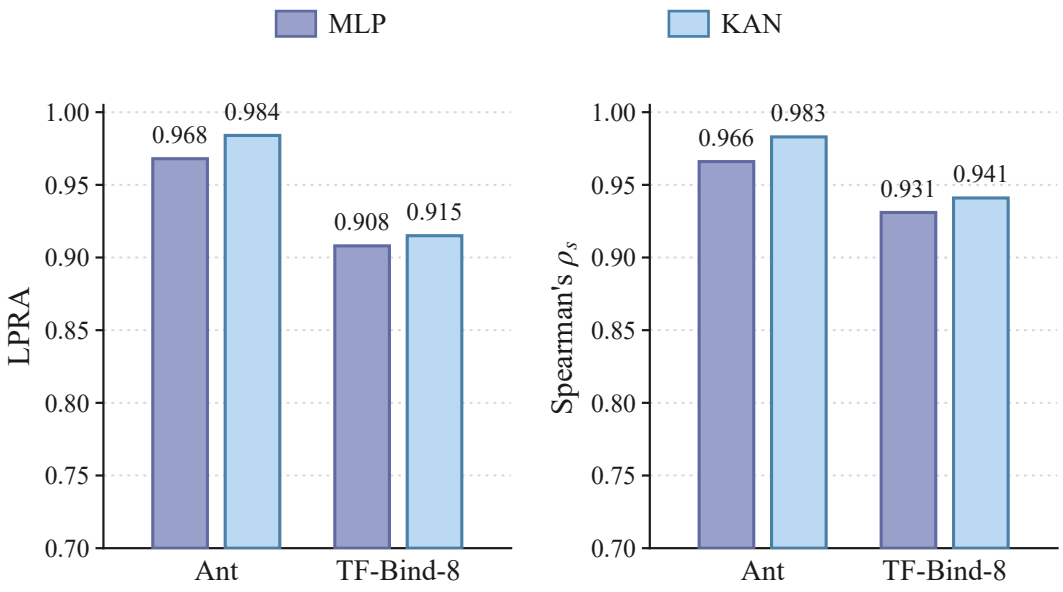


Figure 5: Comparison of KAN and MLP on Ant and TF-Bind-8 under comparable trainable parameter budgets.

according to Eq. (13). We apply the 0.02 reference-range margin before and after pullback, and do not recompute $Q/F/S/T$ or reorder the set after repair. Continuous designs are evaluated directly, whereas discrete designs remain in the continuous relaxation until argmax decoding immediately before oracle evaluation. Figure 7 reports mean nScore_{100} over eight seeds. Because pullback is paired only with the fuzzy policy, this comparison evaluates the complete pipeline rather than a full factorial interaction between pullback and every fusion rule.

D.4 Additional Experiments

D.4.1 Capacity-Controlled Local-Order Results

Under matched active parameter budgets, KAN exhibits higher local ranking stability on both tasks. LPRA improves by 0.016 on Ant and 0.007 on TF-Bind-8.

D.4.2 More Ablations.

As shown in Fig. 6, replacing MLP with KAN raises nScore_{100} from 0.916 to 0.927 on Ant and from 0.638 to 0.643 on TF-Bind-10. Top-ρ regularization then raises the scores to 0.933 and 0.655, while two-stage search reaches 0.946 and 0.662. Reliability-aware delivery produces the largest cumulative increment, yielding 0.976 and 0.678.

As shown in Fig. 7, removing the full reranking and correction module reduces performance by 0.022 on Superconductor and 0.016 on TF-Bind-8. Linear and product fusion both score 0.522 on Superconductor, matching score-only ranking, while decreasing TF-Bind-8 by about 0.03; hard filtering also degrades performance. Fuzzy reranking improves Superconductor from 0.522 to 0.534, and support pullback further raises it to 0.544, showing that the two mechanisms are complementary.

D.4.3 Hyperparameter Sensitivity

As shown in Fig. 8, we evaluate α_{pb}, (τ_F, τ_S, τ_T), ρ, and w_{ctr} on Superconductor. A moderate pullback is beneficial, with $\alpha_{\mathrm{pb}} = 0.35$ performing best. The default thresholds $(0.55, 0.60, 0.28)$ balance retaining unreliable candidates against discarding promising ones. Performance is highest at $\rho = 0.2$, supporting a compact high-potential region, and at $w_{\mathrm{ctr}} = 0.1$; larger contrastive weights increasingly interfere with the primary ranking objective.

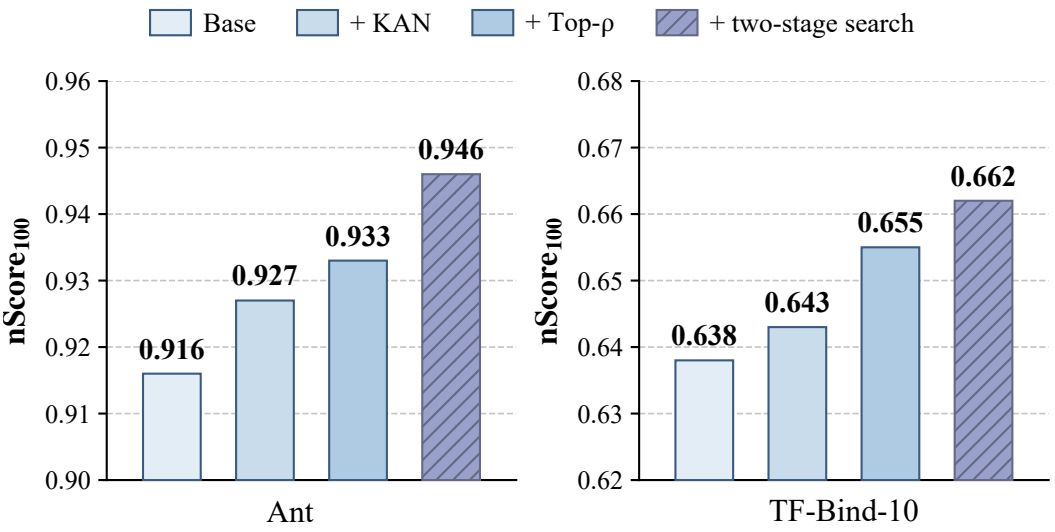


Figure 6: Stage I ablation on Ant and TF-Bind-10. *Base* is the MLP surrogate; "+KAN" replaces MLP with KAN; "+Top-ρ" adds contrastive regularization; and "+Two-stage Search" adds the candidate-search mechanism.

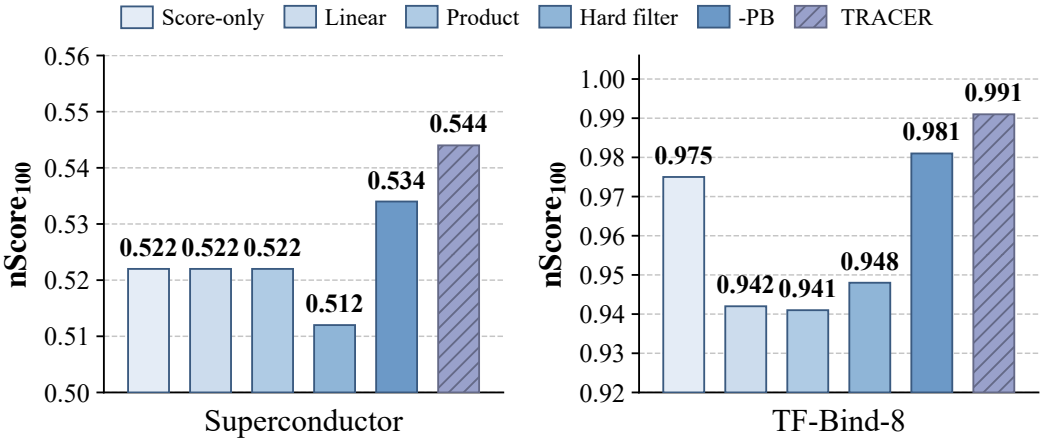


Figure 7: Stage II ablation. *Score-only* uses Stage I output; *Linear* and *Product* aggregate Q, F, and S by weighted sum and multiplication; *Hard Filter* requires all thresholds; and *-PB* removes final support pullback.

D.4.4 Additional 50th-Percentile Results

Table 4 reports compiled 50th-percentile quality for 17 methods; ROOT and DAR report only the 100th percentile. Although TRACER targets the high-value tail, it records the lowest descriptive mean rank ($4.6/17$), followed by RaM-ListNet ($5.7/17$), Match-OPT ($6.2/17$), and Tri-Mentoring ($6.8/17$). These entries retain their source protocols and do not constitute a common-seed statistical comparison.

D.4.5 Gradient Stability Analysis of Top-ρ.

To analyze the effect of Top-ρ on local search stability, we apply Gaussian perturbations of different scales to Ant, Superconductor, and TF-Bind-8, as shown in Fig. 9. The figure reports the stability gains of Top-ρ regularization over RankCosine in terms of gradient directional stability S_{dir} and

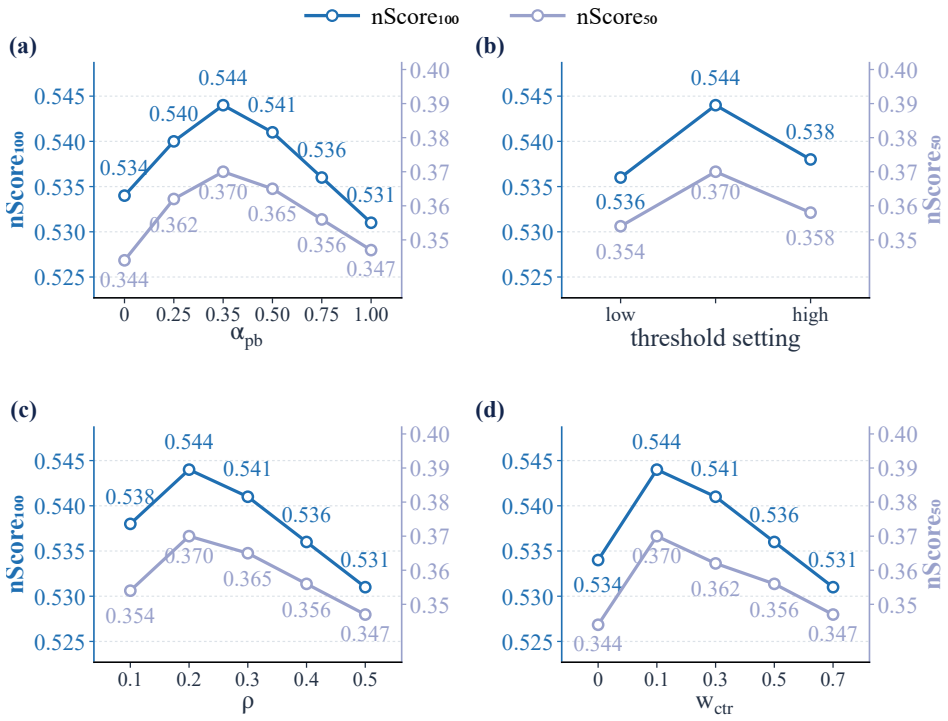


Figure 8: Hyperparameter sensitivity on Superconductor. Panels examine the support-pullback ratio α_{pb}, hard-pass thresholds, Top-ρ fraction, and contrastive-loss weight w_{ctr}.

Table 4: Compiled 50th-percentile normalized scores on Design-Bench. TRACER records the lowest descriptive mean rank, 4.6/17, among methods with available median results. External entries retain their source protocols, so the cross-method ranking is descriptive. Mean ranks are recomputed over the 17 listed methods. The best and runner-up results on each task are shown in Blue and Violet, respectively, and $\mathcal{D}(\text{best})$ 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	/
CbAS (ICML, 2019)	0.385 ± 0.027	0.740 ± 0.023	0.121 ± 0.014	0.422 ± 0.022	0.457 ± 0.006	15.4 / 17
MINs (NeurIPS, 2020)	0.640 ± 0.029	0.886 ± 0.006	0.332 ± 0.014	0.407 ± 0.014	0.465 ± 0.006	9.4 / 17
DDOM (ICML, 2023)	0.598 ± 0.030	0.829 ± 0.050	0.313 ± 0.017	0.416 ± 0.023	0.464 ± 0.006	11.9 / 17
BONET (ICML, 2023)	0.795 ± 0.039	0.906 ± 0.008	0.334 ± 0.032	0.476 ± 0.149	0.452 ± 0.050	8.2 / 17
GTG (NeurIPS, 2024)	0.593 ± 0.022	0.889 ± 0.002	0.350 ± 0.023	0.542 ± 0.038	0.458 ± 0.008	7.4 / 17
COMs (ICML, 2021)	0.532 ± 0.020	0.882 ± 0.002	0.376 ± 0.067	0.513 ± 0.019	0.474 ± 0.014	7.5 / 17
RoMA (NeurIPS, 2021)	0.193 ± 0.017	0.344 ± 0.097	0.368 ± 0.012	0.520 ± 0.074	0.516 ± 0.004	9.6 / 17
IOM (NeurIPS, 2022)	0.459 ± 0.024	0.829 ± 0.022	0.291 ± 0.059	0.490 ± 0.055	0.467 ± 0.000	12.6 / 17
BDI (NeurIPS, 2022)	0.569 ± 0.000	0.876 ± 0.000	0.389 ± 0.022	0.595 ± 0.000	0.429 ± 0.000	7.4 / 17
ICT (NeurIPS, 2023)	0.550 ± 0.028	0.875 ± 0.006	0.333 ± 0.018	0.547 ± 0.041	0.499 ± 0.012	8.0 / 17
Tri-Mentoring(NeurIPS, 2023)	0.548 ± 0.013	0.870 ± 0.002	0.363 ± 0.019	0.619 ± 0.009	0.491 ± 0.001	6.8 / 17
PGS (AISTATS, 2024)	0.190 ± 0.030	0.885 ± 0.001	0.233 ± 0.033	0.503 ± 0.041	0.386 ± 0.177	13.4 / 17
FGM(AISTATS, 2024)	0.532 ± 0.039	0.871 ± 0.017	0.353 ± 0.058	0.540 ± 0.117	0.466 ± 0.004	10.1 / 17
Match-OPT((Hoang et al., 2024))	0.587 ± 0.008	0.887 ± 0.001	0.381 ± 0.038	0.435 ± 0.017	0.471 ± 0.013	6.2 / 17
RaM-RankCosine (ICLR, 2025)	0.566 ± 0.012	0.881 ± 0.003	0.356 ± 0.013	0.544 ± 0.043	0.462 ± 0.006	8.8 / 17
RaM-ListNet (ICLR, 2025)	0.579 ± 0.014	0.888 ± 0.003	0.359 ± 0.013	0.552 ± 0.032	0.467 ± 0.009	5.7 / 17
TRACER	0.605 ± 0.021	0.883 ± 0.017	0.370 ± 0.015	0.561 ± 0.027	0.469 ± 0.008	4.6 / 17

magnitude stability S_{mag}, with the corresponding absolute values provided in Table 7. We observe that Top-ρ consistently improves both S_{dir} and S_{mag} across all tasks and perturbation scales, resulting in positive gains for all 18 comparisons. For example, on Ant at $\sigma = 0.05$, S_{dir} increases from 0.724 to 0.783, corresponding to a gain of 0.059, while S_{mag} increases from 0.633 to 0.706, yielding a gain of 0.073. the stability advantage becomes more pronounced as the perturbation scale increases: the gains range from 0.030 to 0.034 at $\sigma = 0.01$, and increase to 0.055 to 0.073 at $\sigma = 0.05$. These results indicate that Top-ρ effectively reduces the sensitivity of gradient direc-

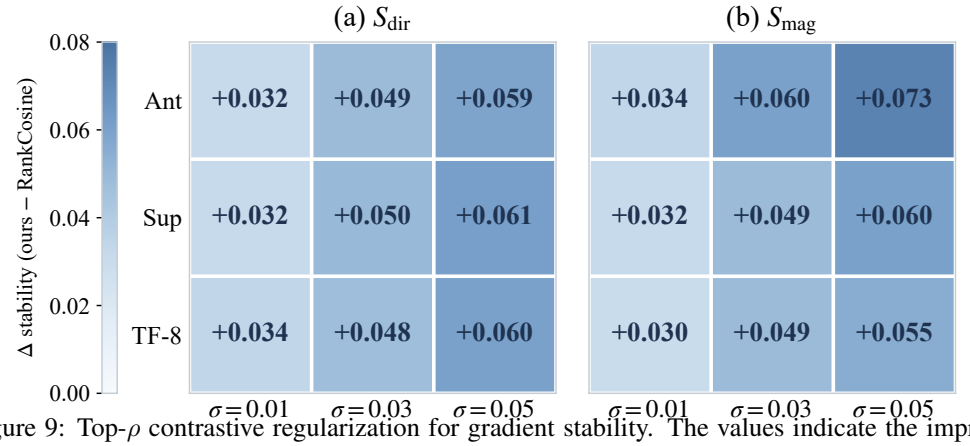


Figure 9: Top-ρ contrastive regularization for gradient stability. The values indicate the improvements in directional stability S_{dir} and magnitude stability S_{mag} achieved by adding Top-ρ contrastive regularization over RankCosine alone under Gaussian perturbations of different scales σ.

Table 5: Tight-budget comparison, macro-averaged over the five Design-Bench tasks after task-wise normalization. Entries are mean ± standard deviation over eight seeds while $K_{\text{out}} = 128$. S3 removes candidates outside the delivery-ready tier, whereas S4 repairs candidates with weak F or S toward offline support. The quantity $|\mathcal{X}^\star|$ is the submitted count.

Budget (pool : submit)	Method	100th	$\lvert\mathcal{X}^\star\rvert$
B-loose (256:128)	S3	0.824 ± 0.028	128
	S4	0.830 ± 0.024	128
B-tight (136:128)	S3	0.815 ± 0.030	119
	S4	0.829 ± 0.025	128
B-critical (128:128)	S3	0.798 ± 0.034	109
	S4	0.827 ± 0.026	128

tion and magnitude to local perturbations, thereby providing a more stable optimization signal for candidate search. This benefit becomes particularly pronounced under stronger local perturbations.

D.4.6 Tight-Budget Sensitivity

Table 5 compares deletion and pullback as the pool-to-budget ratio tightens from 256:128 to 128:128. Deletion (S3) reduces the submitted count $|\mathcal{X}^\star|$ from 128 to 109 and loses effectiveness as the pool tightens. Diagnosis-guided pullback (S4) preserves 128 submissions and achieves higher 100th scores under all three budgets. Thus, the delivery gain reflects both changed candidate priority and budget-preserving repair under tight pools.

D.4.7 Stage-Wise Runtime and Memory Overhead

To evaluate the computational overhead of TRACER, we decompose the framework into three stages: Stage I (KAN surrogate training with Top-ρ contrastive regularization and candidate search), Stage II (candidate-level reliability diagnosis), and Stage III (support pullback), as shown in Table 6. All experiments are conducted on a single NVIDIA A40 GPU with $K_{\text{out}} = 128$. We compare against score-only selection, which selects the Top-K_{out} candidates from the same candidate pool solely according to surrogate scores. Runtime and memory consumption are averaged over two runs, where memory denotes the incremental GPU footprint of each stage. The one-time training cost of Stage I is obtained from checkpoint metadata and amortized over subsequent candidate deliveries. We make two observations:

1) The reliability-aware delivery modules introduced by TRACER incur only limited additional overhead. Stage II and Stage III together require 27.3 ms and 43.8 ms for delivering 128 candidates on Superconductor and TF-Bind-8, respectively, corresponding to approximately 0.2–0.3 ms per candidate. Although this introduces a relative increase of $4.0\times$ and $7.5\times$ compared with score-only selection, the absolute overhead accounts for only 0.15% and 0.11% of the post-training runtime.

Table 6: Stage-wise runtime and incremental GPU memory of TRACER against score-only selection over the same candidate pool, measured on one NVIDIA A40 with $K_{\text{out}} = 128$. Entries are mean $\pm$ standard deviation over two runs. The multiplier in parentheses is runtime relative to score-only selection on the same task. The total excludes the one-time Stage I training row; it sums stage runtimes and takes the peak of the stage memories.

	Superconductor		TF-Bind-8	
Stage	Runtime (s)	ΔMem (MiB)	Runtime (s)	ΔMem (MiB)
Score-only selection	0.0068 ± 0.0016	7.3	0.0059 ± 0.0003	2.2
S-I: KAN + Top-ρ training	1109.8 ± 3.2	3186.5	809.7 ± 3.4	1423.9
S-I: candidate search	18.441 ± 0.448	224.2	39.617 ± 0.065	66.9
S-II: candidate diagnosis	0.0219 ± 0.0012 (3.2×)	61.7	0.0311 ± 0.0038 (5.3×)	18.0
S-III: support pullback	0.0054 ± 0.0011 (0.8×)	11.6	0.0126 ± 0.0009 (2.2×)	3.3
Delivery layer (S-II + S-III)	0.0273 (4.0×)	61.7	0.0438 (7.5×)	18.0
Total per run (excl. training)	18.468 ± 0.450	224.2	39.660 ± 0.061	66.9

Table 7: Source values for Figure 9: directional stability S_{dir} and magnitude stability S_{mag} under Gaussian perturbations of scale σ, averaged over eight random seeds. Higher is better; the figure encodes the row-wise margin.

		S_{dir}			S_{mag}		
Task	Objective	σ=0.01	0.03	0.05	σ=0.01	0.03	0.05
Ant Morphology	RankCosine	0.914	0.812	0.724	0.872	0.746	0.633
	+ Top-ρ	0.946	0.861	0.783	0.906	0.806	0.706
Superconductor	RankCosine	0.908	0.804	0.715	0.916	0.811	0.721
	+ Top-ρ	0.940	0.854	0.776	0.948	0.860	0.781
TF-Bind-8	RankCosine	0.912	0.818	0.728	0.920	0.815	0.732
	+ Top-ρ	0.946	0.866	0.788	0.950	0.864	0.787

2) The reliability assessment and correction stages introduce only minor memory overhead. Stage II increases GPU memory usage by 61.7 MiB and 18.0 MiB on Superconductor and TF-Bind-8, respectively, while Stage III adds 11.6 MiB and 3.3 MiB. Both remain below the peak memory consumption of candidate search (224.2 MiB and 66.9 MiB). The dominant resource consumption comes from the one-time surrogate training, which requires 1109.8 s and 3186.5 MiB on Superconductor and 809.7 s and 1423.9 MiB on TF-Bind-8.

D.4.8 Gradient-Stability Source Values

To further assess the effect of Top-ρ regularization on local ranking stability, we report in Table 7 the absolute stability values corresponding to Figure 9. As the perturbation scale increases, stability degrades under both objectives; however, the Top-ρ-regularized surrogate consistently achieves higher stability than RankCosine across all settings. The improvement becomes more pronounced at larger perturbation scales, increasing from $+0.030$–$+0.034$ at $\sigma = 0.01$ to $+0.055$–$+0.073$ at $\sigma = 0.05$.