MITIGATING OVER-OPTIMIZATION IN PRM-GUIDED SEARCH IN MATHEMATICAL REASONING BY OPTIMIZING THE GUIDE

Anonymous authors
Paper under double-blind review

ABSTRACT

Process reward models (PRMs) provide dense step-level guidance for search-based reasoning, enabling inference-time compute to be allocated toward promising partial solutions. However, recent evidence suggests that PRM-guided search can over-optimize imperfect process rewards, pruning viable trajectories while expanding spurious ones. In this work, we theoretically show that directly leveraging PRM score is vulnerable to verifier noise through an extreme-value effect: non-viable prefixes become more likely to receive spuriously high scores as reasoning depth increase. Therefore, we formulate the PRM-guided search as a robust optimization problem over plausible reward perturbations, termed maximin PRM-guided search, leading to a training-free robust process supervision method that preserves promising alternatives when step-level scores are noisy. Maximin PRM-guided search mitigates this failure mode by reducing sensitivity to over-optimized PRM outliers. Without fine-tuning or online adaptation, maximin search consistently improves the PRM-guided search by 17–35% on average, outperforming outcome- and step-level baselines in 14 out of 16 settings.

1 INTRODUCTION

Large language models (LLMs) perform strongly on tasks that benefit from explicit intermediate reasoning, including mathematical problem solving and code generation, and chain-of-thought prompting is a simple way to elicit this behavior (Wei et al., 2022; Kojima et al., 2022). Performance depends not only on base-model capability but also on inference-time computation through sampling, search, and verification (Wang et al., 2022; Yao et al., 2023; Besta et al., 2024; Cobbe et al., 2021).

Search-based inference must decide which partial trajectories merit further computation. Process reward models (PRMs) provide step-level values that can guide pruning and allocate compute more selectively than outcome-only scoring (Uesato et al., 2022; Lightman et al., 2023; Feng et al., 2023; Chen et al., 2024; Wang et al., 2024). This benefit, however, depends critically on verifier reliability. PRMs are learned proxies for latent reasoning quality and can degrade under limited supervision, model misspecification, or distribution shift (Lightman et al., 2023; Casper et al., 2023). In sequential search, false negatives can permanently prune promising branches, while false positives can waste substantial budget. Thus, optimizing imperfect PRM scores can induce *over-optimization* or *reward hacking* (Coste et al., 2023; Gao et al., 2023; Skalse et al., 2022).

A natural response is to improve the reward model online, for example through test-time training or related adaptation mechanisms (Sun et al., 2020; Shinn et al., 2023; Hosseini et al., 2024). However, this elevates the computational and algorithmic burden rather than fundamentally resolving the vulnerability. Such approaches introduce additional optimization into the critical path of inference, increasing computational cost and implementation complexity (Snell et al., 2024). Furthermore, their stability relies on precisely the low-signal, out-of-distribution regime that already makes reward estimation difficult (Gao et al., 2023; Casper et al., 2023). More importantly, adaptation aims to repair the scorer itself, whereas the core decision problem is how to act when the scorer is inherently uncertain. From this perspective, the central challenge is not solely accurate reward learning, but robust decision-making under reward uncertainty (Iyengar, 2005; Kumar et al., 2020).

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.

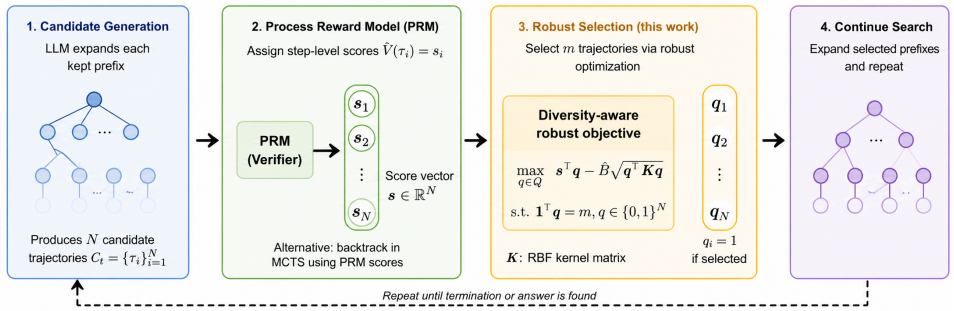


Figure 1: **Maximin PRM-guided search.** At each pruning step, the LLM generates candidate trajectories and a PRM assigns step-level scores. Rather than retaining only the highest-scoring candidates, maximin keeps m prefixes that balance high PRM score with diversity under an RBF kernel K, reducing sensitivity to correlated or noisy verifier errors.

In this work, we address PRM imperfection from that perspective (cf. Figure 1). We formulate PRM-guided search as a robust optimization problem in which reward estimates are treated as uncertain rather than exact. The key idea is as follows: when reward estimates are unreliable, the search should reflect plausible reward perturbations. This formulation leads to inference procedures that are better aligned with the true objective, mitigating over-optimization of imperfect process rewards and yielding strong robustness to noisy reward estimates. More broadly, our results suggest that scaling inference-time compute in PRM-guided search requires not only more search, but also better algorithmic robustness to the noisy score evaluations to make search effective in the first place.

Our contributions are threefold: **1)** we characterize a concrete over-optimization failure mode in PRM-guided search: maximization over noisy process scores can select non-viable prefixes even when viable continuations are present, with risk determined by verifier uncertainty, generator coverage, search width, and depth; **2)** we formulate step-level guidance as robust optimization under verifier misspecification, yielding a kernel regularizer that controls worst-case aggregate score error; **3)**, we derive a training-free plug-in algorithm that improves PRM-guided search by 17–35% on average and attains the best or tied-best result in 14 of 16 evaluated settings.

2 Related Work

PRM-guided inference allocates search compute using learned step-level evaluation, including self-evaluation-guided beam search, AlphaZero-style search, step-level beam search, and actor–critic decoding (Xie et al., 2023; Feng et al., 2023; Chen et al., 2024; Dong et al., 2025). Recent self-improvement methods such as ReST-MCTS*, V-STaR, and PRIME address reward misspecification by adapting the policy, verifier, or reward model, but require additional training or outcome supervision (Zhang et al., 2024; Hosseini et al., 2024; Cui et al., 2025). In contrast, we keep the generator and PRM fixed and make the inference rule itself robust to verifier error.

Over-optimization has been studied mainly for outcome-level inference-time optimization such as Best-of-N, where increasing optimization pressure against a misspecified reward model can eventually reduce true performance (Jinnai et al., 2025; Ichihara et al., 2025; Khalaf et al., 2025; Huang et al., 2025). PRM-guided search is less understood theoretically despite evidence that partial-state evaluations become unreliable under long horizons and distribution shift (Zhang et al., 2025; Khalifa et al., 2025; Cinquin et al., 2025). We study how verifier misspecification interacts with sequential pruning; an extended discussion is provided in Appendix E.1.

3 Background

Reasoning models. A reasoning model generates a structured output for an input problem x whose ground-truth answer is y, which typically consists of a sequence of intermediate reasoning steps followed by a final answer. We analyze inference on a single input x and suppress the dependence on x in the notation when it is clear from context. Let $s \in \mathcal{S} \triangleq \mathcal{V}^*$ denote a *thought* as the Kleene closure of a set of tokens $\mathcal{V}$. Typically, it corresponds to a semantically coherent reasoning segment

separated from adjacent segments by a fixed delimiter "\n\n" (Lightman et al., 2023; Zhang et al., 2025; Cinquin et al., 2025). A *reasoning trajectory* of length t is an ordered sequence of thoughts $\tau^{(t)} = (x, s_1, \ldots, s_t)$, with $s_i \in \mathcal{S}$ for $i \in [t] \triangleq \{1, 2, \cdots, t\}$. We let $T_{\max}$ denote a maximum reasoning horizon and let $\mathsf{T} = \inf\{t \le T_{\max} : s_t = \texttt{EOS}\}$ denote the random termination time, with the convention $\mathsf{T} = T_{\max}$ if no EOS token is generated earlier. The distribution of T is induced by the generator and the search procedure, and may depend on the search history. For simplicity, we pad positions after T with EOS, so that all trajectories can be represented on the common horizon $T_{\max}$. When we consider multiple trajectories with the same length, we use $\tau_i^{(t)}$ to denote the i-th trajectory of length t for $i \in [N]$ with N being the maximum number of trajectories. For a prefix trajectory $\tau^{(t)} = (x, s_1, \ldots, s_t)$ and a new thought $s \in \mathcal{S}$, we write $\tau^{(t)} \oplus s \triangleq (x, s_1, \ldots, s_t, s)$ as a prefix extension. We write $\tau^{(t)} \preceq \tau^{(t')}$ if a trajectory $\tau^{(t)}$ is a prefix of $\tau^{(t')}$.

A base reasoning model's thought-level policy π defines a conditional distribution over the next thought given a partial trajectory: $\pi(s \mid \tau^{(t)}) \in \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ denotes the probability simplex over $\mathcal{S}$. This induces a distribution over complete trajectories by autoregressive factorization, $\pi(\tau^{(T)}) = \prod_{t=1}^{T} \pi(s_t \mid \tau^{(t-1)})$, with $\tau^{(0)} \triangleq x$. When a trajectory is terminated, a deterministic answer extraction map $\mathrm{ans} : \bigcup_{t=0}^{T_{\max}} \mathcal{S}^t \to \mathcal{Y}$ returns the final output $\hat{y} = \mathrm{ans}(\tau)$ in answer space $\mathcal{Y}$.

Step-level guidance. To guide search over partial trajectories, we consider a process reward model (PRM) $\widehat{V}$ that assigns a scalar value to each intermediate trajectory, $\widehat{V}(\tau^{(t)}) \in \mathbb{R}$, interpreted as an estimate of the future utility of continuing from $\tau^{(t)}$. We let the oracle value function $V(\tau^{(t)})$ output a utility of $\tau^{(t)}$, such as the logit of the probability of yielding the ground-truth answer (Zhang et al., 2025) or whether all intermediate thoughts in $\tau^{(t)}$ are correct (Uesato et al., 2022). We assume V is bounded as $\underline{V}_t \le V(\tau^{(t)}) \le \overline{V}_t$ for all $\tau^{(t)} \in \mathcal{S}^t$.

We use step-level beam search as the canonical PRM-guided search procedure in our analysis. At depth t, let $\mathcal{B}_t = \{\tau_i^{(t)}\}_{i=1}^{w_t}$ denote the active beam, where w_t is the beam width. Each trajectory $\tau_i^{(t)} \in \mathcal{B}_t$ is expanded by sampling or enumerating candidate next thoughts from the base generator $\pi(\cdot \mid \tau_i^{(t)})$. Writing $\mathcal{C}_{t+1} = \bigcup_{i=1}^{w_t} \left\{\tau_i^{(t)} \oplus s : s \sim \pi(\cdot \mid \tau_i^{(t)})\right\}$ for the resulting set of candidate continuations, the PRM $\hat{V}$ is used to score each candidate prefix $\tau \in \mathcal{C}_{t+1}$. The size of $\mathcal{C}_{t+1}$, $|\mathcal{C}_{t+1}| \le N$, is the total number of trajectories at step t and $b_t = \lfloor N/w_t \rfloor$ is the branching factor at t. We note that the procedure reduces to outcome-level selection (Best-of-N) when $b_t = 1$ and reduces to the step-level greedy decoding when $w_t = 1$. The next beam is then formed by selecting the top-w_{t+1} trajectories under this score (i.e., $\mathcal{B}_{t+1} \subseteq \mathcal{C}_{t+1}$ with $|\mathcal{B}_{t+1}| = w_{t+1}$) and then repeating the procedure until termination (i.e., collecting N answers). Thus, π determines how candidate reasoning steps are proposed, while $\widehat{V}$ determines how computation is allocated across them.

4 Closer look at an over-optimization in PRM-guided search

The goal of PRM-guided search can be understood as maximizing the average score of the selected trajectories. Formally, it can be represented as solving the optimization problem of $\mathcal{B}_t \in \arg\max_{\mathcal{B}'_t \subseteq \mathcal{C}_t : |\mathcal{B}'_t| = w_t} \frac{1}{|\mathcal{B}'_t|} \sum_{\tau \in \mathcal{B}'_t} V(\tau)$. Ideally, this guidance filters out unpromising trajectories, thereby compensating for an imperfect generator. In practice, however, off-the-shelf PRMs can suffer from reward over-optimization and even underperform outcome-level methods that naively extend previous thought prefixes such as Best-of-N (Cinquin et al., 2025; Zhang et al., 2025; Hu et al., 2025). We therefore quantify the failure induced by optimizing an imperfect verifier.

Setup. We consider the beam-style search over partial solutions described in §3. Let $\mathcal{T}^\star \subseteq \mathcal{S}^T$ denote the set of successful full trajectories satisfying $\mathrm{ans}(\tau^\star) = y$. Define the viable-prefix set

$$\widetilde{\mathcal{P}}_t^\star = \left\{\tau^{(t)} \in \mathcal{S}^t : \exists \tau^\star \in \mathcal{T}^\star \text{ such that } \tau^{(t)} \preceq \tau^\star\right\},$$

and its complement $\widetilde{\mathcal{P}}_t^- = \mathcal{S}^t \setminus \widetilde{\mathcal{P}}_t^\star$. For a candidate set $\mathcal{C}_t$, let $\mathcal{C}_t^\star = \mathcal{C}_t \cap \widetilde{\mathcal{P}}_t^\star$ and $\mathcal{C}_t^- = \mathcal{C}_t \cap \widetilde{\mathcal{P}}_t^-$, with $N_t^\star = |\mathcal{C}_t^\star|$ and $N_t^- = |\mathcal{C}_t^-|$.

Assumption 1 (Verifier regularity)**.** *The learned verifier satisfies* $\widehat{V}(\tau^{(t)}) = V(\tau^{(t)}) + \beta_t(\tau^{(t)}) + \varepsilon_t(\tau^{(t)})$ *for all* $t \in [T_{\max}]$*, where* $\beta_t : \mathcal{S}^t \to \mathbb{R}$ *is a deterministic bias function satisfying* $|\beta_t(\tau^{(t)})| \le$

B_t for all $\tau^{(t)} \in \mathcal{S}^t$. Conditional on the search history $\mathcal{H}_t$ and realized candidate set $\mathcal{C}_t$, the noise variables $\{\varepsilon_t(\tau) : \tau \in \mathcal{C}_t\}$ are independent and identically distributed as $\mathcal{N}(0, \sigma_t^2)$. Further, we assume $B_1 \geq \cdots \geq B_{T_{\max}}$ and $\sigma_1 \geq \cdots \geq \sigma_{T_{\max}}$.

Assumption 1 captures the observation that early-stage evaluations are more uncertain because the final answer is distant, whereas later steps provide clearer outcome signals (Hu et al., 2025; Cinquin et al., 2025). The Gaussian assumption is used only for the extreme-value analysis below; structured and correlated verifier misspecification is addressed separately in §5. Additional discussion of the modeling assumptions is deferred to Appendix A.1.

Main result. We first isolate the pure reward-maximization regime by considering greedy selection, $w_t = 1$. The general beam-search extension is given in Appendix D.1. We assume throughout that $\mathcal{C}_t^\star \neq \emptyset$ and $\mathcal{C}_t^- \neq \emptyset$. At step t, greedy PRM-guided search selects

$$\widehat{\tau}_{\text{greedy}}^{(t)} \in \arg\max_{\tau^{(t)} \in \mathcal{C}_t} \widehat{V}(\tau^{(t)}). \tag{1}$$

We now lower-bound the probability that this maximization selects a non-viable prefix even though viable candidates are present. Proofs of all formal claims are provided in Appendix C.

Theorem 1. (Greedy reward hacking via deterministic cutoff). *Under Assumption 1 and any fixed $\lambda \in \mathbb{R}$, let $\Gamma_t(\mathcal{C}_t) = \sup_{\tau_\star \in \mathcal{C}_t^\star, \tau_- \in \mathcal{C}_t^-} \{V(\tau_\star) + \beta_t(\tau_\star) - V(\tau_-) - \beta_t(\tau_-)\}$, $q_t(\lambda) = \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star}$, and $r_t(\lambda) = 1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t)+\lambda}{\sigma_t}\right)$. Then,*

$$\Pr\left(\widehat{\tau}_{\text{greedy}}^{(t)} \notin \widetilde{\mathcal{P}}_t^\star \middle| \mathcal{C}_t\right) \geq q_t(\lambda)\left[1 - (1 - r_t(\lambda))^{N_t^-}\right] \triangleq \ell_t(\lambda; \mathcal{C}_t). \tag{2}$$

Consequently, $\Pr\left(\widehat{\tau}_{\text{greedy}}^{(t)} \notin \widetilde{\mathcal{P}}_t^\star \middle| \mathcal{C}_t\right) \geq \sup_{\lambda \in \mathbb{R}} \ell_t(\lambda; \mathcal{C}_t)$.

Theorem 1 identifies a certified reward-hacking event analogous to over-optimization in Best-of-N sampling (Gao et al., 2023): all viable prefixes fall below a cutoff while at least one non-viable prefix exceeds it. The bound can be written as

$$\ell_t(\lambda; \mathcal{C}_t) = \underbrace{\Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star}}_{\text{viable-frontier term}} \cdot \underbrace{\left[1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t) + \lambda}{\sigma_t}\right)^{N_t^-}\right]}_{\text{non-viable outlier term}}. \tag{3}$$

Thus, failure is governed jointly by the strength of the viable frontier and the extreme upper tail generated by the non-viable population. A weak generator makes both effects less favorable: it decreases $N_t^\star$ while increasing N_t^-, leaving fewer viable prefixes to compete against more opportunities for an extreme positive verifier error.

On the effective gap and verifier noise. The quantity $\Gamma_t(\mathcal{C}_t)$ in (2) is a conservative separation scale for the cutoff argument: it upper-bounds the deterministic advantage that viable prefixes may have over non-viable ones. Hence, a larger $\Gamma_t(\mathcal{C}_t)$ requires a non-viable prefix to receive a larger positive noise realization before it can overtake the viable frontier, whereas a smaller gap makes the selection more sensitive to verifier noise. Because the cutoff λ is only an auxiliary proof parameter and is optimized in the final certificate, the effect of σ_t should not be interpreted while holding λ fixed. Instead, writing $\lambda = a\sigma_t$ gives $\sup_{\lambda \in \mathbb{R}} \ell_t(\lambda; \mathcal{C}_t) = \sup_{a \in \mathbb{R}} \Phi(a)^{N_t^\star}\left[1 - \Phi\left(a + \frac{\Gamma_t(\mathcal{C}_t)}{\sigma_t}\right)^{N_t^-}\right]$. Thus, after optimizing over the cutoff, the deterministic separation and verifier uncertainty enter through the normalized gap $\gamma_t = \Gamma_t(\mathcal{C}_t)/\sigma_t$. For fixed $\Gamma_t(\mathcal{C}_t)$, increasing σ_t decreases this normalized separation and increases the certified failure risk. In other words, verifier noise is most harmful when the deterministic distinction between viable and non-viable prefixes is small relative to the noise scale.

Extensions across generation, depth, and beam width. The same mechanism persists in the full sequential search. Accounting for generator randomness reveals a tradeoff: increasing the candidate width improves the probability of sampling viable continuations, but also increases the number of non-viable prefixes over which verifier errors are maximized. Moreover, because leaving the viable-prefix set is irreversible, these local failure risks accumulate across reasoning depth. For beam width $w_t > 1$, failure requires at least w_t non-viable verifier outliers rather than one, so beam search mitigates but does not eliminate the effect. We formalize the generator-averaged failure probability, its depth-wise accumulation, and the beam-search extension in Appendix A.2.

5 MITIGATING OVER-OPTIMIZATION IN STEP-LEVEL GUIDED DECODING

In this section, we address the failure of imperfect PRM-guided search by replacing pure score maximization with a reward-misspecification-robust objective that explicitly accounts for correlated verifier errors.

5.1 A ROBUST OBJECTIVE FOR STEP-LEVEL SELECTION

At a pruning stage t, let $\mathcal{C}_t = \{\tau_1, \dots, \tau_N\}$ denote the candidate prefixes, with estimated scores $s_i = \widehat{V}(\tau_i)$ and oracle scores $r_i = V(\tau_i)$, where we suppress the depth superscript for readability. Then, we write the signed verifier error as $\nu_i = r_i - s_i$. A binary vector $q \in \{0,1\}^N$ represents the retained beam, where $q_i = 1$ indicates that τ_i is selected and $\mathbf{1}^\top q = m$ fixes the beam size.

The central question is how to model uncertainty in ν_i's. A pointwise uncertainty model such as $|\nu_i| \le B$ assumes bounded verifier errors and treats the errors across prefixes independently. This misses an important failure mode of learned verifiers that semantically similar reasoning states can share systematic misspecification. If a PRM overvalues one local reasoning pattern, several nearby prefixes may therefore receive optimistic scores simultaneously, so selecting many of them compounds the same verifier error rather than diversifying exposure to uncertainty.

We encode this structure through a kernel on trajectory representations:

$$k(\tau_i, \tau_j) = \exp\big(-\gamma d^2(\tau_i, \tau_j)\big)\,, \tag{4}$$

where $d(\tau_i, \tau_j) = \sqrt{2 - 2\frac{\mathrm{emb}(\tau_i)^\top \mathrm{emb}(\tau_j)}{\|\mathrm{emb}(\tau_i)\|\,\|\mathrm{emb}(\tau_j)\|}}$ and $\mathrm{emb}(\cdot)$ is a trajectory representation, e.g., the last hidden state of the PRM. Let $K_{ij} = k(\tau_i, \tau_j)$ denote the corresponding Gram matrix, with $K_{ii} = 1$ and $0 \le K_{ij} \le 1$. We set γ using the median pairwise-distance heuristic (Gretton et al., 2012; Garreau et al., 2017), as described in Appendix F.

Let Φ_k denote the RKHS associated with k. For the robustness derivation below, we model the structured component of verifier misspecification by a function $e \in \Phi_k$ satisfying $\nu_i = e(\tau_i)$ and $\|e\|_{\Phi_k} \le B$. For common kernels such as the RBF kernel, this is a smoothness model: nearby prefixes may exhibit similar verifier errors. This complements Assumption 1, whose Gaussian component is used to isolate idiosyncratic noise in the extreme-value failure analysis; more generally, verifier error may contain both a smooth structured component and residual stochastic noise. Further details connecting the two models are given in Appendix B.1.

For a fixed beam selection q, the true aggregate value of the retained beam is $r^\top q = s^\top q + \sum_{i=1}^N q_i e(\tau_i)$. Thus, relative to the nominal PRM score $s^\top q$, the effect of verifier misspecification is entirely captured by the aggregate perturbation $\sum_i q_i e(\tau_i)$. The next result characterizes the largest magnitude that this perturbation can attain over all error functions in the RKHS uncertainty set.

Theorem 2. (Worst-case aggregate error under RKHS smoothness). *If the estimation error is induced by $e \in \Phi_k$ with $\|e\|_{\Phi_k} \le B$, then for every $q \in \{0,1\}^N$,*

$$\sup_{\|e\|_{\Phi_k} \le B} \left| \sum_{i=1}^N q_i e(\tau_i) \right| = B\sqrt{q^\top K q}. \tag{5}$$

Theorem 2 gives the worst-case magnitude of the aggregate perturbation for any fixed beam selection q. Since the RKHS ball is symmetric, the worst-case aggregate score of a fixed selection q is $\inf_{\|e\|_{\Phi_k} \le B} \sum_{i=1}^N q_i\big(s_i + e(\tau_i)\big) = s^\top q - B\sqrt{q^\top K q}$. Hence, $s^\top q$ is the nominal value assigned by the PRM, while $B\sqrt{q^\top K q}$ is exactly the amount by which that aggregate value can deteriorate under the worst structured verifier error in the uncertainty set.

A robust decision should therefore select the beam with the largest value after accounting for this worst-case downward perturbation. The radius B in Theorem 2 is unknown in practice, so we use an inference-time radius $\widetilde{B}$, which we calibrate as described below, and solve

$$\max_{q \in \{0,1\}^N} \quad J(q) \triangleq s^\top q - \widetilde{B}\sqrt{q^\top K q} \qquad \text{s.t.} \qquad \mathbf{1}^\top q = m. \tag{6}$$

For the feasible family $\mathcal{Q} \triangleq \{q \in \{0,1\}^N : \mathbf{1}^\top q = m\}$, whenever $\|e\|_{\Phi_k} \le \widetilde{B}$, $J(q)$ is a certified lower bound on the true aggregate value $r^\top q$ for every $q \in \mathcal{Q}$. When $\widetilde{B} = 0$, the robustness correction vanishes and (6) reduces exactly to standard PRM-guided top-m selection.

Connection to the over-optimization failure. The analysis in § 4 shows that score maximization can prefer prefixes receiving unusually favorable verifier errors. The robust objective targets the structured version of this mechanism. If a local region of trajectory space is systematically overestimated, ordinary top-m selection may devote several beam slots to nearby prefixes that all inherit the same optimistic bias. Under (6), however, repeating that correlated error also increases $q^\top Kq$. Hence, a cluster of non-viable prefixes with only a marginal nominal score advantage cannot dominate the beam merely by reproducing the same misspecification across many similar candidates; its score advantage must be large enough to overcome the corresponding increase in worst-case uncertainty.

Figure 4 in Appendix B.2 illustrates that the standard score maximization can concentrate both selections inside a spuriously overestimated local region, whereas the robust objective can retain a separated high-value candidate by accounting for the shared uncertainty of the cluster. A formal clustered-failure condition under which the robust selector preserves a viable alternative discarded by standard PRM beam selection is given in Appendix **??**.

Empirical choice of the RKHS radius B. The RKHS radius B is unknown, and estimating such a radius from data is known to be delicate (Maddalena et al., 2021; Scharnhorst et al., 2022; Tokmak et al., 2025). When calibration data containing both predicted and reference scores is available, let $\nu = (\nu_1, \dots, \nu_N)$ denote the observed residual vector. We use the sample-based radius

$$\widetilde{B} \triangleq \sqrt{\nu^\top K^\dagger \nu}, \tag{7}$$

where $K^\dagger$ is the Moore–Penrose pseudoinverse. When the interpolation constraints are feasible, this quantity is the RKHS norm of the minimum-norm function that matches the observed residuals. Thus, (7) provides a data-dependent scale for the uncertainty penalty rather than treating $\widetilde{B}$ as an arbitrary diversity coefficient. We give the corresponding variational characterization in Appendix B.3. When oracle step rewards are unavailable, we instead use the maximum within-trajectory variation in estimated process rewards as a proxy for the residual magnitude. This heuristic is not intended to consistently estimate the true RKHS norm; its role is to adapt the conservatism of the selector to the observed instability of the verifier.

Connection to diversity-based regularized decoding. Although the regularizer in (6) superficially resembles diversity-promoting decoding such as diverse beam search, its role is different: it arises as a robustness penalty against smooth PRM misspecification rather than as a generic lexical or semantic diversity heuristic. In particular, $\sqrt{q^\top Kq}$ controls the worst-case aggregate verifier error under the RKHS error model, whereas minimum Bayes risk (MBR) uses candidate geometry for consensus or centrality (Freitag et al., 2022; Eikema & Aziz, 2022; Jinnai et al., 2025), determinantal point process-style selection promotes volume or low redundancy (Kulesza et al., 2012; Meister et al., 2021), and diverse beam search encourages lexical or semantic mode coverage during decoding (Vijayakumar et al., 2016; Shi et al., 2025). Further, this perspective provides a principled mechanism to select $\tilde{B}$ from the calibration set, unlike the previous regularized decoding methods. We provide a detailed comparison in Appendix E.2.

5.2 Method

The main computational difficulty in (6) is the non-separable term $\sqrt{q^\top Kq}$. We handle it through an exact variational reformulation that converts (6) into a family of cardinality-constrained submodular maximization problems, yielding a principled approximate solution.

Variational reformulation. We use the identity $\sqrt{z} = \min_{\eta>0} \left(\frac{z}{2\eta} + \frac{\eta}{2}\right)$ for $z \ge 0$. Applying this identity to $z = q^\top Kq$ yields the exact representation

$$J(q) = \max_{\eta>0} J_\eta(q), \qquad J_\eta(q) \triangleq s^\top q - \frac{\tilde{B}}{2\eta} q^\top Kq - \frac{\tilde{B}}{2}\eta, \tag{8}$$

yielding $\max_{q\in\mathcal{Q}} J(q) = \max_{q\in\mathcal{Q}} \max_{\eta>0} J_\eta(q) = \max_{\eta>0} \max_{q\in\mathcal{Q}} J_\eta(q)$.

Algorithm 1 Grid-based submodular surrogate maximization for maximin PRM-guided search

Require: scores $s \in \mathbb{R}^N$, kernel matrix $K \in \mathbb{R}^{n\times n}$, penalty $\tilde{B} > 0$, cardinality m, grid parameter $\xi > 0$, inner solver $\mathcal{A}$

1: Construct the grid $\mathcal{G}_\xi = \left\{\eta_t = \sqrt{m}(1+\xi)^h : h = 0, 1, \ldots, H\right\},\ H \triangleq \left\lceil \log_{1+\xi} \sqrt{m} \right\rceil$
2: **for** each $\eta_i \in \mathcal{G}_\xi$ **do**
3: Compute $q_i^\dagger \leftarrow \mathcal{A}(\eta_i)$, where $\mathcal{A}(\eta_i)$ approximately solves $\max_{q\in\mathcal{Q}} \left[s^\top q - \frac{\tilde{B}}{2\eta_i} q^\top K q\right]$
4: **end for**
5: Return $q^\dagger \in \arg\max_{i\in\{0,\ldots,H\}} \left[s^\top q_i^\dagger - \tilde{B}\sqrt{{q_i^\dagger}^\top K q_i^\dagger}\right]$

Outer optimization. For every feasible q, the maximizing auxiliary variable is $\eta = \sqrt{q^\top K q} \in [\sqrt{m}, m]$. We therefore discretize this interval with the multiplicative grid

$$\mathcal{G}_\xi = \left\{\eta_h = \sqrt{m}(1+\xi)^h : h = 0, 1, \ldots, H\right\}, \qquad H \triangleq \left\lceil \log_{1+\xi} \sqrt{m} \right\rceil. \tag{9}$$

For each $\eta \in \mathcal{G}_\xi$, we approximately maximize the corresponding surrogate J_η, and then select among the resulting subsets using the original square-root objective J. Thus, the surrogates are used only to generate candidate solutions; final selection is performed according to the robust objective in (6).

Inner optimization. For fixed η, the surrogate J_η is a cardinality-constrained non-monotone submodular objective. We approximately maximize it using randomized greedy (Buchbinder et al., 2014), followed by objective-improving 1-swap local search (Lee et al., 2009; Fadaei et al., 2011). The randomized-greedy stage provides the approximation factor used below, while the swap stage monotonically improves the surrogate objective. The exact marginal-gain updates, swap rule, runtime, and implementation cap are given in Appendix B.4.

Error bound. We finally quantify the approximation error relative to the original square-root objective. The result separates the loss due to discretizing η from the approximation factor of the inner submodular solver. For simplicity, we state the result when $J_\eta(q) \geq 0$ for all $q \in \mathcal{Q}$ and $\eta \in [\sqrt{m}, m]$; the general shifted-objective result is given in Corollary 3 in Appendix D.3.

Theorem 3. (End-to-end approximation guarantee). *Let $q^\star \in \arg\max_{q\in\mathcal{Q}} J(q)$, and let $q^\dagger$ be the output of Algorithm 1. Let us assume that for every $\eta \in \mathcal{G}_\xi$, the randomized greedy warm start returns a set q_η^{rg} satisfying $\mathbb{E}\left[J_\eta\left(q_\eta^{\mathrm{rg}}\right)\right] \geq \rho \max_{q\in\mathcal{Q}} J_\eta(q)$ and that the subsequent pairwise-swap local search returns $q_\eta^\dagger$ with $J_\eta(q_\eta^\dagger) \geq J_\eta\left(q_\eta^{\mathrm{rg}}\right)$. Then, it holds that*

$$\mathbb{E}[J(q^\dagger)] \geq \rho\left(J(q^\star) - \frac{\tilde{B} m \xi^2}{2(1+\xi)}\right). \tag{10}$$

Together, Theorem 2 and Theorem 3 connect robustness and computation: the former shows that $J(q)$ is a certified lower bound on the true aggregate score under RKHS-bounded misspecification, while the latter shows that Algorithm 1 approximately maximizes this certificate, up to the inner-solver factor ρ and the grid error $O(\tilde{B}m\xi^2/(1+\xi))$. Thus, the method retains a set only when its nominal PRM advantage remains competitive after accounting for its exposure to structured verifier error.

6 Experiments

We evaluate maximin search on GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), AIME'24, and AIME'25, covering grade-school through competition-level mathematical reasoning. We report accuracy on all benchmarks. Because each AIME benchmark contains only 30 problems, we average over three random seeds.

Base models and configuration. We use Qwen2.5-Math-1.5B (Yang et al., 2024) and Phi-3.5-mini-instruct (Abdin et al., 2024) as base generators together with Skywork o1-PRM-1.5B (He et al., 2024). The two generator families test whether the method depends on a particular policy while remaining weak enough that step-level pruning decisions are non-degenerate. Following He et al. (2024), we average the PRM's step rewards to obtain the trajectory-level value estimate $\widehat{V}(\tau)$. We use zero-shot

Table 1: **Benchmark results under different generators and inference budgets.** Best results within each generator–dataset–budget block are bolded; second-best results are underlined. Greedy uses $N = 1$. We report the standard deviations in Table 2 in Appendix F.

Generator	Method	GSM8K		MATH-500		AIME'24		AIME'25	
		$N = 16$	$N = 64$	$N = 16$	$N = 64$	$N = 16$	$N = 64$	$N = 16$	$N = 64$
Qwen2.5-Math-1.5B	Greedy	82.3	82.3	70.4	70.4	10.0	10.0	10.0	10.0
	Beam search	84.3	84.9	72.2	75.2	**16.7**	**16.7**	10.0	10.0
	BoN	87.3	88.4	<u>75.2</u>	76.8	**16.7**	**16.7**	**15.6**	**17.8**
	BoN-MBR	<u>88.0</u>	<u>88.8</u>	<u>75.2</u>	<u>77.4</u>	<u>14.4</u>	**16.7**	**15.6**	<u>12.2</u>
	SBS	87.4	87.5	72.4	76.6	8.9	<u>15.6</u>	<u>13.3</u>	**17.8**
	Maximin	**88.9**	**89.3**	**76.8**	**78.8**	**16.7**	**16.7**	**15.6**	**17.8**
Phi-3.5-mini-instruct	Greedy	84.1	84.1	45.8	45.8	3.3	3.3	0.0	0.0
	Beam search	85.8	84.7	53.6	53.0	3.3	3.3	**6.7**	0.0
	BoN	89.3	89.6	<u>63.4</u>	65.8	<u>6.7</u>	5.6	<u>2.2</u>	3.3
	BoN-MBR	<u>89.7</u>	89.7	62.2	65.0	<u>6.7</u>	6.7	<u>2.2</u>	3.3
	SBS	88.9	**91.4**	63.2	<u>66.4</u>	4.4	<u>7.8</u>	<u>2.2</u>	<u>6.7</u>
	Maximin	**89.8**	<u>91.2</u>	**63.8**	**70.2**	**8.9**	**8.9**	<u>2.2</u>	**7.8**

chain-of-thought prompting; prompts, generation settings, and optimization hyperparameters are provided in Appendix F.

Baselines and inference budgets. All methods are evaluated under matched budgets $N \in \{16, 64\}$. We compare against (i) greedy decoding and likelihood beam search without external supervision; (ii) Best-of-N (BoN) and MBR-BoN (Jinnai et al., 2025) using outcome-level selection; and (iii) SBS (Chen et al., 2024), a PRM-guided step-level beam search that retains the top-m prefixes by PRM score. For PRM-guided methods we use branch factor 4 and $m = \lfloor N/4 \rfloor$. Detailed baseline definitions are given in Appendix F.

6.1 Benchmark results

Table 1 shows that naively using PRM scores as step-level guidance is not sufficient. As expected, external guidance substantially improves over unguided decoding; for example, on MATH-500 with Phi-3.5-mini-instruct at $N = 64$, beam search obtains 53.0%, while BoN/SBS/maximin reach 65.8%/66.4%/70.2%. However, SBS does not consistently outperform outcome-level selection methods such as BoN or BoN-MBR. For instance, with Qwen2.5-Math-1.5B, SBS is below BoN-MBR on GSM8K at $N = 64$ (87.5 vs. 88.8), on MATH-500 at $N = 16$ (72.4 vs. 75.2), and on AIME'24 at $N = 16$ (10.0 vs. 13.3). This is consistent with prior work showing that direct step-level aggregation can be brittle (Cinquin et al., 2025).

However, adding the quadratic kernel penalty through maximin search makes the same step-level PRM signal more effective. Maximin achieves the best or tied-best result in 14 of the 16 generator–dataset–budget blocks, and improves over SBS by roughly 17%–35% on average across generator and search-budget settings, with only a 0.27% increase in runtime overhead when $N = 16$. The gains are more pronounced on harder benchmarks; for example, on Phi-3.5-mini-instruct, maximin improves over SBS from 66.4% to 70.2% on MATH-500 at $N = 64$, and from 6.7% to 13.3% on AIME'24 at $N = 64$. Consistent with this trend, Figure 5 shows that on MATH-500 the methods perform similarly on easier levels, while maximin obtains the highest Level-5 accuracy for both $N = 16$ and $N = 64$. These results suggest that PRMs contain useful process-level information, but that robust aggregation is important for exploiting it. This improvement is obtained without online adaptation or fine-tuning, using the same off-the-shelf generators and PRMs.

6.2 Impacts of reward noise

We test the robustness of SBS and maximin search to noisy reward estimates by adding independent Gaussian noise $\mathcal{N}(0, \epsilon^2)$ to the process reward, with $\epsilon \in \{0.1, 0.3, 1.0, 3.0\}$. All experiments use Qwen2.5-Math-1.5B guided by Skywork o1-PRM-1.5B on 100 random samples from MATH-500. On this subset, maximin and SBS achieve 79% and 76% accuracies, respectively. Although these numbers differ slightly from the full evaluation due to subsampling, the relative trends are consistent.

Figure 2 shows that both methods degrade as the noise level increases, but maximin search is generally less sensitive to score perturbations. In particular, at $\epsilon = 1.0$, SBS shows a 12.7% relative

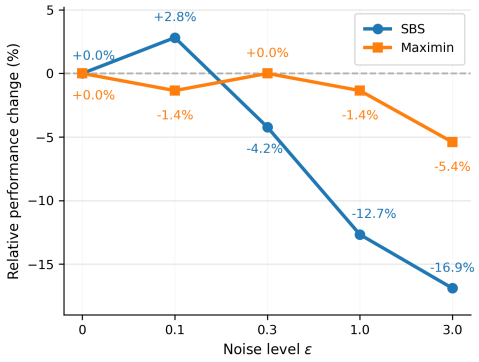


Figure 2: **Robustness to injected verifier noise.** Relative performance change from the noise-free baseline as the score-noise level increases.

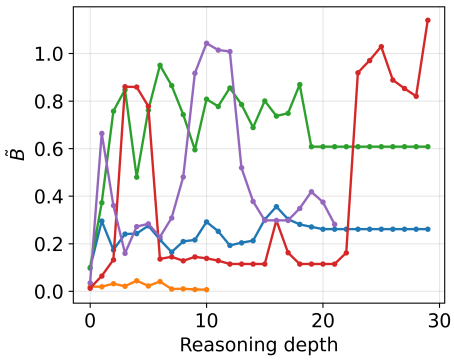


Figure 3: **Estimated $\widetilde{B}$ across reasoning depth.** Each curve is one example; the inferred uncertainty scale varies across instances and steps.

degradation from its noise-free baseline, while maximin search changes by only 1.4%. Even for extreme noise level at $\epsilon = 3.0$, maximin search degrades by only 5.4% as opposed to 16.9% in SBS. Interestingly, a small amount of noise improves SBS, increasing performance from 71% to 73% at $\epsilon = 0.1$. This suggests that mild perturbations may regularize SBS by preventing over-reliance on small score differences. However, this benefit disappears once the perturbation becomes large enough to substantially corrupt the score ordering. Overall, maximin search remains more robust because it does not depend solely on the noisy score estimation, but also preserves a less concentrated reasoning trajectories.

6.3 Algorithmic Design Choices & Sensitivity Analysis

With the setup same as in §6.2, we examine three major algorithmic choices in maximin PRM-guided search: the local improvement step, the adaptive RKHS radius, and the representation geometry used by the RBF kernel.

Local refinement validates the robust objective. The 1-swap stage further optimizes the robust objective after randomized-greedy initialization. Removing it reduces accuracy from 79% to 77%, only slightly above SBS. This distinction is important: additional optimization of the *raw* imperfect PRM can exacerbate reward hacking, while additional optimization of the robust lower bound improves performance. The result therefore supports the objective itself rather than merely the randomized initialization.

The uncertainty scale should adapt to the local problem. We set $\widetilde{B}$ using the sample-dependent proxy in (7) rather than treating it as a global diversity coefficient. Fixed choices $\widetilde{B} \in [0.03, 0.1, 0.3, 1]$ give accuracies $[71\%, 78\%, 77\%, 74\%]$, respectively. Although a tuned constant can approach the adaptive rule on this small subset, Figure 3 shows substantial variation in the inferred radius across both examples and reasoning depth. This heterogeneity is expected because the appropriate robust penalty depends on the local verifier-error geometry of the sampled candidates, and it motivates instance- and step-dependent calibration rather than a universal coefficient.

Kernel geometry matters. Our RBF kernel uses angular distance after ℓ_2-normalizing PRM representations. Replacing it with Euclidean distance in the unnormalized representation space reduces accuracy from 79% to 74%. This supports the use of representation geometry that is insensitive to raw norm artifacts and aligns with common practice for comparing language representations (Li et al., 2020; Su et al., 2021). More importantly for our formulation, it shows that robustness depends on a meaningful notion of which prefixes are likely to share verifier misspecification.

Hyperparameter sensitivity Maximin remains effective across the tested sampling temperatures and is stable over a broad range of grid resolutions. The default $\xi = 0.05$ achieves a favorable accuracy–cost trade-off with only 0.26% runtime overhead relative to SBS; detailed sensitivity curves and runtime comparisons are reported in Appendix F.

AI USE STATEMENT

In this work, we used generative AI tools for grammar checking and language editing, as well as for improving the visual presentation of Figures 1 and 4. For these figures, the authors first designed and sketched the figure content and layout, and generative AI tools were used only to refine the visual style and clarity of the author-created figures. We did not use generative AI tools to generate research ideas, formulate hypotheses, design experiments, conduct analyses, interpret results, or draw scientific conclusions. All AI-assisted text and figures were reviewed and approved by the authors. We take full responsibility for the final content of this work, including all text, claims, and figures produced with the aid of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI Conference on Artificial Intelligence*, 2024.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *ACM-SIAM Symposium on Discrete Algorithms*, 2014.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: process supervision without process. In *Advances in Neural Information Processing Systems*, 2024.

Tristan Cinquin, Geoff Pleiss, and Agustinus Kristiadi. Limits of prm-guided tree search for mathematical reasoning with llms. *arXiv preprint arXiv:2510.20272*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Heng Dong, Kefei Duan, and Chongjie Zhang. Enhancing decision-making of large language models via actor-critic. *arXiv preprint arXiv:2506.06376*, 2025.

Bryan Eikema and Wilker Aziz. Sampling-based approximations to minimum bayes risk decoding for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Salman Fadaei, MohammadAmin Fazli, and MohammadAli Safari. Maximizing non-monotone submodular set functions subject to different constraints: Combined algorithms. *Operations Research Letters*, 39(6):447–451, 2011.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825, 2022.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2023.

Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu, Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 open series, November 2024. URL `https://doi.org/10.5281/zenodo.16998085`.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

Pengfei Hu, Zhenrong Zhang, Qikai Chang, Shuhang Liu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, Feng Ma, et al. Prm-bas: Enhancing multimodal reasoning through prm-guided beam annealing search. *arXiv preprint arXiv:2504.10222*, 2025.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*, 2025.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, Kenshi Abe, Mitsuki Sakamoto, and Eiji Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *arXiv preprint arXiv:2502.12668*, 2025.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. Regularized best-of-n sampling with minimum bayes risk objective for language model alignment. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio du Pin Calmon. Inference-time reward hacking in large language models. *arXiv preprint arXiv:2506.19248*, 2025.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Annual ACM Symposium on Theory of Computing*, 2009.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Conference on Empirical Methods in Natural Language processing*, 2020.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2023.

Emilio Tanowe Maddalena, Paul Scharnhorst, and Colin N Jones. Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134:109896, 2021.

Clara Meister, Martina Forster, and Ryan Cotterell. Determinantal beam search. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

Preetum Nakkiran. More data can hurt for linear regression: Sample-wise double descent. *arXiv preprint arXiv:1912.07242*, 2019.

Paul Scharnhorst, Emilio T Maddalena, Yuning Jiang, and Colin N Jones. Robust uncertainty bounds in reproducing kernel hilbert spaces: A convex optimization approach. *IEEE Transactions on Automatic Control*, 68(5):2848–2861, 2022.

Weijie Shi, Yue Cui, Yaguang Wu, Jingzhi Fang, Shibo Zhang, Mengze Li, Sirui Han, Jia Zhu, Jiajie Xu, and Xiaofang Zhou. Semantic-guided diverse decoding for large language model. *arXiv preprint arXiv:2506.23601*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, 2022.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Connectionist Models Summer School*. Psychology Press, 1993.

Abdullah Tokmak, Kiran G Krishnan, Thomas B Schön, and Dominik Baumann. Safe exploration in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2503.10352*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems*, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems*, 2024.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics*, 2025.

A Additional technical details for section 4

A.1 Discussion of the verifier regularity assumption

The bounded-bias condition in Assumption 1 imposes a uniform error bound over trajectories. This condition can be replaced by a probabilistic bound, in which case the corresponding analysis becomes probabilistic as well; we use the uniform formulation for simplicity.

The Gaussian noise assumption is adopted for analytical tractability and yields closed-form extreme-value expressions, following related uses of Gaussian models in high-dimensional interpolation and double-descent analyses (Advani et al., 2020; Belkin et al., 2020; Nakkiran, 2019). It is used only for the Gaussian extreme-value analysis and is not intended to describe all forms of verifier error. In particular, systematic misspecification may be correlated across semantically similar prefixes. We model this structured component separately through the RKHS formulation in §5.

A.2 Generator, depth, and beam-search extensions

The main-text analysis conditions on a realized candidate set and isolates the one-step verifier-selection failure. Here, we incorporate randomness in candidate generation, characterize how the resulting local risks accumulate across the reasoning horizon, and extend the argument from greedy selection to beam search.

Theorem 4. (Generator-averaged greedy failure). *Let us suppose the current greedy prefix at depth $t-1$ is viable, i.e., $\widehat{\tau}^{(t-1)}_{\mathrm{greedy}} \in \widetilde{\mathcal{P}}^{\star}_{t-1}$. Let $\mathcal{H}_t$ denote the search history before generating candidates at depth t. Let us also assume a Markovian candidate generator: conditional on $\mathcal{H}_t$, the N_t candidate continuations are sampled independently, and each continuation remains viable with probability*

$$p_t(\mathcal{H}_t) = \Pr_{s\sim\pi(\cdot|\widehat{\tau}^{(t-1)}_{\mathrm{greedy}})} \left(\widehat{\tau}^{(t-1)}_{\mathrm{greedy}} \oplus s \in \widetilde{\mathcal{P}}^{\star}_t \;\middle|\; \mathcal{H}_t \right).$$

Finally, assume that for every mixed candidate set with $\mathcal{C}^{\star}_t \neq \emptyset$ and $\mathcal{C}^{-}_t \neq \emptyset$, there exists an $\mathcal{H}_t$-measurable $\overline{\Gamma}_t(\mathcal{H}_t)$ such that $\Gamma_t(\mathcal{C}_t) \leq \overline{\Gamma}_t(\mathcal{H}_t)$. Given Assumption 1 and any fixed $\lambda \in \mathbb{R}$, define

$$a_t(\lambda) = \Phi\left(\frac{\lambda}{\sigma_t}\right), \qquad b_t(\lambda;\mathcal{H}_t) = \Phi\left(\frac{\overline{\Gamma}_t(\mathcal{H}_t)+\lambda}{\sigma_t}\right).$$

Then,

$$\Pr\left(\widehat{\tau}^{(t)}_{\mathrm{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_t \;\middle|\; \mathcal{H}_t\right) \geq \delta_t(\lambda;\mathcal{H}_t),$$

where

$$\delta_t(\lambda;\mathcal{H}_t) = (1-p_t(\mathcal{H}_t))^{N_t} + \sum_{k=1}^{N_t} \binom{N_t}{k} p_t(\mathcal{H}_t)^k \left(1-p_t(\mathcal{H}_t)\right)^{N_t-k} a_t(\lambda)^k \left[1-b_t(\lambda;\mathcal{H}_t)^{N_t-k}\right].$$

For the generator-averaged results, we further assume that, conditional on $\mathcal{H}_t$, candidate continuations are sampled independently from the generator and that verifier noises are conditionally independent of the candidate-sampling randomness given the realized candidate set.

Corollary 1. (Depth-wise accumulation under Markovian generation). *Under the assumptions of Theorem 4 for all $t \in [T_{\max}]$, let*

$$\mathcal{S}_t = \left\{\widehat{\tau}^{(t)}_{\mathrm{greedy}} \in \widetilde{\mathcal{P}}^{\star}_t\right\}$$

denote the event that greedy search remains viable through depth t. Define $\delta^{\star}_t(\mathcal{H}_t) = \sup_{\lambda\in\mathbb{R}} \delta_t(\lambda;\mathcal{H}_t)$ on $\mathcal{S}_{t-1}$ and $\delta^{\star}_t(\mathcal{H}_t) \triangleq 0$ on $\mathcal{S}^c_{t-1}$. Then,

$$\Pr(\mathcal{S}_{T_{\max}}) \leq \mathbb{E}\left[\prod_{t=1}^{T_{\max}} \left(1-\delta^{\star}_t(\mathcal{H}_t)\right)\right].$$

Theorem 4 separates one-step failure into generator coverage and verifier selection. The term $(1-p_t(\mathcal{H}_t))^{N_t}$ is the probability that the generator samples no viable continuation, while the

remaining terms average the cutoff-based over-optimization event of Theorem 1 over the random number of viable candidates. Thus, increasing N_t improves viable coverage but simultaneously creates more non-viable opportunities for verifier hacking. When $p_t(\mathcal{H}_t)$ is small, this imbalance can remain severe even when viable continuations are sampled.

Corollary 1 shows that these local risks compound across reasoning depth. Because a prefix that leaves the viable set cannot subsequently reach a successful trajectory, trajectory-level success requires avoiding both coverage and selection failures at every step. In particular, if $\delta_t^\star(\mathcal{H}_t) \geq \underline{\delta}_t$ almost surely, then

$$\Pr\left(\exists t \leq T_{\max} : \widehat{\tau}_{\text{greedy}}^{(t)} \notin \widetilde{\mathcal{P}}_t^\star\right) \geq 1 - \prod_{t=1}^{T_{\max}} (1 - \underline{\delta}_t).$$

Impact of beam search. Let $\mathcal{B}_t$ denote the top-w_t prefixes under $\widehat{V}$, with $w_t = 1$ recovering greedy search. The greedy cutoff argument certifies failure when all viable prefixes stay below the cutoff and at least one non-viable prefix exceeds it. For width w_t, at least w_t such non-viable prefixes are required. Accordingly, Corollary 2 gives

$$\Pr\left(\mathcal{B}_t \cap \widetilde{\mathcal{P}}_t^\star = \emptyset \mid \mathcal{C}_t\right) \geq q_t(\lambda) \Pr\left(\text{Binomial}(N_t^-, r_t(\lambda)) \geq w_t\right),$$

with $q_t(\lambda)$ and $r_t(\lambda)$ as in Theorem 1. Thus, beam search increases the verifier-outlier tolerance from one to w_t, mitigating but not removing over-optimization. Because the cutoff structure is unchanged, the generator-averaging and depth-accumulation arguments above carry over with the binomial tail replacing the single-outlier term.

Implications for more sophisticated guided search. The same analysis also clarifies what more sophisticated test-time procedures must improve. Methods such as MCTS, verifier refinement, or test-time training can help by increasing the effective separation $\Gamma_t(\mathcal{C}_t)$, reducing harmful verifier bias, or reducing stochastic uncertainty σ_t (Zhang et al., 2024; Hosseini et al., 2024; Snell et al., 2024). However, any residual probability of overvaluing non-viable prefixes can still be amplified by maximization over candidates and repeated selection across reasoning steps (Gao et al., 2023). Thus, the difficulty is not merely verifier imperfection itself, but the selection bias induced when search repeatedly maximizes an imperfect score (Thrun & Schwartz, 1993).

B Additional technical details for section 5

B.1 Relation between verifier regularity and RKHS smoothness

The RKHS model in §5 is intended to refine the structured component of Assumption 1, rather than replace it. More generally, write the signed verifier error as

$$\nu_i = r_i - s_i = \tilde{e}(\tau_i) + \zeta_i,$$

where $\tilde{e} \in \Phi_k$ with $\|\tilde{e}\|_{\Phi_k} \leq B$ is a smooth structured bias component and ζ_i is a residual stochastic component.

When $K_{ii} = k(\tau_i, \tau_i) = 1$, the reproducing property gives

$$|\tilde{e}(\tau_i)| \leq \|\tilde{e}\|_{\Phi_k} \sqrt{k(\tau_i, \tau_i)} \leq B.$$

Thus, the RKHS condition implies the same type of pointwise control used for the bounded-bias component in Assumption 1. The difference is that Assumption 1 constrains the structured error only pointwise, whereas the RKHS model additionally constrains how that error varies across trajectories.

In particular, nearby trajectories under the kernel are required to have related structured errors. This allows the RKHS model to capture correlated verifier misspecification across semantically similar prefixes. The RKHS assumption does not itself impose the i.i.d. Gaussian-noise assumption used in the extreme-value analysis of §4.

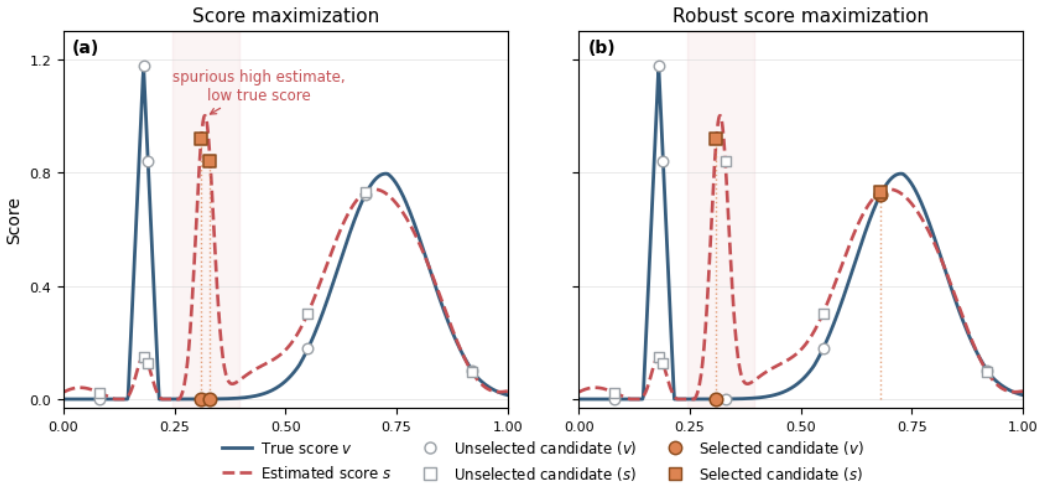


Figure 4: **Robust subset selection under score misspecification** ($N = 8, m = 2$)**.** The true score contains a sharp high-value region and a broader high-value region, while the estimated score introduces a spurious local peak with low true value. Standard score maximization selects two nearby candidates from the spuriously overestimated region. The robust objective penalizes this concentrated selection and instead retains a candidate from a distinct high-value region, reducing exposure to correlated score error.

B.2 Illustration of robust subset selection

Figure 4 illustrates why selecting several nearby candidates can be fragile under smooth verifier misspecification. If the verifier overestimates an entire local region of representation space, pure score maximization may allocate multiple beam slots to candidates sharing the same error. Because these candidates are mutually similar, their selection produces a large value of $q^\top Kq$, so the robust objective in (6) explicitly charges for this correlated exposure.

In contrast, a candidate from a different region contributes less pairwise similarity to the selected set. The robust objective can therefore prefer a slightly lower nominal PRM score when doing so substantially reduces the worst-case aggregate error radius. This is the geometric intuition behind the repulsive term $\sqrt{q^\top Kq}$.

B.3 Calibration-based choice of the RKHS radius

In Theorem 2, the constant B upper-bounds the RKHS norm of the unknown error function, $\|e\|_{\Phi_k} \leq B$. Estimating this radius directly is generally difficult (Maddalena et al., 2021; Scharnhorst et al., 2022; Tokmak et al., 2025).

When calibration data with both estimated and reference scores is available, let

$$\nu = (\nu_1, \ldots, \nu_N), \qquad \nu_i = r_i - s_i.$$

We define

$$\tilde{B} \triangleq \sqrt{\nu^\top K^\dagger \nu},$$

where $K^\dagger$ is the Moore–Penrose inverse of K. This quantity admits the variational characterization

$$\tilde{B} \in \min_{\tilde{e} \in \Phi_k : \tilde{e}(\tau_i) = \nu_i \ \forall i} \|\tilde{e}\|_{\Phi_k}.$$

Hence, $\tilde{B}$ is the smallest RKHS radius consistent with the observed residual pattern on the calibration sample.

In practice, this sample-based radius may be multiplied by a constant factor to reduce the risk of underestimating off-sample error complexity. When oracle step rewards are unavailable, we instead use the maximum within-trajectory variation in estimated step rewards as a proxy for the residual

magnitude. This heuristic is intended only to scale the robust penalty according to observed verifier instability and is not claimed to consistently estimate the true RKHS norm.

B.4 Inner submodular optimization details

For fixed $\eta > 0$, the surrogate objective

$$J_\eta(q) = s^\top q - \frac{\tilde{B}}{2\eta} q^\top K q - \frac{\tilde{B}}{2}\eta$$

is a cardinality-constrained non-monotone submodular objective.

We use a two-stage procedure consisting of randomized greedy initialization followed by improving pairwise swaps.

Randomized-greedy initialization. Let $1_j \in \{0,1\}^N$ denote the unit vector whose j-th component is one. Starting from $q_0 = \varnothing$, at round $r = 1, \dots, m$ we compute the marginal gain

$$\Delta_r(j) \triangleq J_\eta(q_{r-1} + 1_j) - J_\eta(q_{r-1})$$

for each unselected trajectory. We then form the set M_r containing the m largest marginal gains among the remaining trajectories, or all remaining trajectories if fewer than m remain, and sample $\iota(r)$ uniformly from M_r. The update is

$$q_r \leftarrow q_{r-1} + 1_{\iota(r)}.$$

After m rounds, this produces a feasible $q_m \in \mathcal{Q}$.

Following Buchbinder et al. (2014), the randomized-greedy solution satisfies

$$\mathbb{E}[J_\eta(q_m)] \geq \rho \max_{q \in \mathcal{Q}} J_\eta(q)$$

for an approximation factor $\rho \in (0, 1]$.

Pairwise-swap refinement. Starting from the randomized-greedy solution, we repeatedly replace an active trajectory i by an inactive trajectory j whenever the swap improves the surrogate objective. The corresponding gain is

$$\Delta_{i \to j}(q) = s_j - s_i - \frac{\tilde{B}}{\eta} \sum_{k \in [N] \setminus \{i\} : q_k = 1} (K_{jk} - K_{ik}). \tag{11}$$

We apply improving exchanges until reaching a 1-swap local optimum or the implementation cap is reached. Because every accepted swap strictly improves the objective over a finite feasible family, cyclic behavior is impossible.

For dense K, the randomized-greedy initialization requires $O(Nm)$ marginal updates. In our implementation, we cap the number of swap-refinement steps at 30 to keep the total cost controlled.

C Proof of claims

C.1 Proof of Theorem 1

Theorem 1. (Greedy reward hacking via deterministic cutoff). *Under Assumption 1 and any fixed* $\lambda \in \mathbb{R}$, *let* $\Gamma_t(\mathcal{C}_t) = \sup_{\tau_\star \in \mathcal{C}_t^\star, \tau_- \in \mathcal{C}_t^-} \{V(\tau_\star) + \beta_t(\tau_\star) - V(\tau_-) - \beta_t(\tau_-)\}$, $q_t(\lambda) = \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star}$, *and* $r_t(\lambda) = 1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t)+\lambda}{\sigma_t}\right)$. *Then,*

$$\Pr\left(\widehat{\tau}_{\text{greedy}}^{(t)} \notin \widetilde{\mathcal{P}}_t^\star \middle| \mathcal{C}_t\right) \geq q_t(\lambda)\left[1 - (1 - r_t(\lambda))^{N_t^-}\right] \triangleq \ell_t(\lambda; \mathcal{C}_t). \tag{2}$$

Consequently, $\Pr\left(\widehat{\tau}_{\text{greedy}}^{(t)} \notin \widetilde{\mathcal{P}}_t^\star \middle| \mathcal{C}_t\right) \geq \sup_{\lambda \in \mathbb{R}} \ell_t(\lambda; \mathcal{C}_t)$.

Proof. The core idea of the proof is to introduce a deterministic cutoff a_t and lower-bounds to the failure event by the intersection of two simpler events: $\left\{\max_{\tau_\star \in \mathcal{C}_t^\star} \widehat{V}(\tau_\star) \le a_t\right\} \cap \left\{\max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t\right\}$. Specifically, from Assumption 1, we can set $a_t = \sup_{\tau'_\star \in \mathcal{C}_t^\star}\{V_t(\tau'_\star) + b_t(\tau'_\star)\} + \lambda$ for any fixed $\lambda \in \mathbb{R}$. Given the best viable verifier score $M_t^\star = \max_{\tau_\star \in \mathcal{C}_t^\star} \widehat{V}(\tau_\star)$, greedy selection fails whenever $M_t^\star \le a_t$ and $\max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t$. Indeed, on this event, some non-viable prefix has verifier score at least as large as the best viable prefix. Therefore,

$$\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}_t^\star \mid \mathcal{C}_t\right) \ge \Pr\left(M_t^\star \le a_t,\ \max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t\right). \tag{12}$$

The viable and non-viable noise variables are independent, so the two events on the right-hand side are independent, and thus

$$\Pr\left(M_t^\star \le a_t,\ \max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t\right) = \Pr(M_t^\star \le a_t \mid \mathcal{C}_t) \Pr\left(\max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t\right). \tag{13}$$

First, for every viable prefix $\tau_\star \in \mathcal{C}_t^\star$, $\widehat{V}(\tau_\star) = V(\tau_\star) + \beta_t(\tau_\star) + \varepsilon_t(\tau_\star) \le \sup_{\tau'_\star \in \mathcal{C}_t^\star}\{V_t(\tau'_\star) + b_t(\tau'_\star)\} + \varepsilon_t(\tau_\star)$. Thus $M_t^\star \le a_t$ whenever $\varepsilon_t(\tau_\star) \le \lambda$ for all $\tau_\star \in \mathcal{C}_t^\star$. Therefore, we get

$$\Pr(M_t^\star \le a_t \mid \mathcal{C}_t) \ge \Pr\left(\varepsilon_t(\tau_\star) \le \lambda,\ \forall \tau_\star \in \mathcal{C}_t^\star | \mathcal{C}_t\right) = \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star}. \tag{14}$$

Second, for every non-viable prefix $\tau_- \in \mathcal{C}_t^-$, $\widehat{V}(\tau_-) = V(\tau_-) + \beta_t(\tau_-) + \varepsilon_t(\tau_-) \ge \inf_{\tau'_- \in \mathcal{C}_t^-}\{V(\tau'_-) + \beta_t(\tau'_-)\} + \varepsilon_t(\tau_-)$. Hence $\widehat{V}(\tau_-) > a_t$ whenever $\varepsilon_t(\tau_-) \ge a_t - \inf_{\tau'_- \in \mathcal{C}_t^-}\{V(\tau'_-) + \beta_t(\tau'_-)\}$. Since $a_t - \inf_{\tau'_- \in \mathcal{C}_t^-}\{V(\tau'_-) + \beta_t(\tau'_-)\} = \Gamma_t(\mathcal{C}_t) + \lambda$, we have

$$\Pr(\widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t) \ge 1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t) + \lambda}{\sigma_t}\right) = r_t(\lambda). \tag{15}$$

Since the non-viable noises are independent by Assumption 1,

$$\Pr\left(\max_{\tau_- \in \mathcal{C}_t^-} \widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t\right) \ge 1 - (1 - r_t(\lambda))^{N_t^-}. \tag{16}$$

Combining the two bounds gives

$$\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}_t^\star \mid \mathcal{C}_t\right) \ge \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star} \left[1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t) + \lambda}{\sigma_t}\right)^{N_t^-}\right]. \tag{17}$$

Since this holds for every $\lambda \in \mathbb{R}$, taking the supremum over λ gives the final claim. □

C.2 Proof of Theorem 4

Theorem 4. (Generator-averaged greedy failure). *Let us suppose the current greedy prefix at depth $t-1$ is viable, i.e., $\widehat{\tau}^{(t-1)}_{\text{greedy}} \in \widetilde{\mathcal{P}}^\star_{t-1}$. Let $\mathcal{H}_t$ denote the search history before generating candidates at depth t. Let us also assume a Markovian candidate generator: conditional on $\mathcal{H}_t$, the N_t candidate continuations are sampled independently, and each continuation remains viable with probability*

$$p_t(\mathcal{H}_t) = \Pr_{s \sim \pi(\cdot | \widehat{\tau}^{(t-1)}_{\text{greedy}})}\left(\widehat{\tau}^{(t-1)}_{\text{greedy}} \oplus s \in \widetilde{\mathcal{P}}_t^\star \;\middle|\; \mathcal{H}_t\right).$$

Finally, assume that for every mixed candidate set with $\mathcal{C}_t^\star \neq \emptyset$ and $\mathcal{C}_t^- \neq \emptyset$, there exists an $\mathcal{H}_t$-measurable $\overline{\Gamma}_t(\mathcal{H}_t)$ such that $\Gamma_t(\mathcal{C}_t) \le \overline{\Gamma}_t(\mathcal{H}_t)$. Given Assumption 1 and any fixed $\lambda \in \mathbb{R}$, define

$$a_t(\lambda) = \Phi\left(\frac{\lambda}{\sigma_t}\right), \qquad b_t(\lambda; \mathcal{H}_t) = \Phi\left(\frac{\overline{\Gamma}_t(\mathcal{H}_t) + \lambda}{\sigma_t}\right).$$

Then,

$$\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_{t} \;\middle|\; \mathcal{H}_t\right) \geq \delta_t(\lambda; \mathcal{H}_t),$$

where

$$\delta_t(\lambda; \mathcal{H}_t) = (1 - p_t(\mathcal{H}_t))^{N_t} + \sum_{k=1}^{N_t} \binom{N_t}{k} p_t(\mathcal{H}_t)^k \left(1 - p_t(\mathcal{H}_t)\right)^{N_t - k} a_t(\lambda)^k \left[1 - b_t(\lambda; \mathcal{H}_t)^{N_t - k}\right].$$

Proof. Let $K_t = N^{\star}_t = |\mathcal{C}^{\star}_t|$. By the Markovian independent-generation assumption, $K_t \sim \text{Binomial}(N_t, p_t(\mathcal{H}_t))$.

If $K_t = 0$, then no viable continuation is present in the candidate set. Therefore greedy search necessarily selects a non-viable prefix, and $\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_{t} \;\middle|\; K_t = 0, \mathcal{H}_t\right) = 1$.

We now fix $1 \leq k \leq N_t$. Conditional on $K_t = k$, we have $N^{\star}_t = k$ and $N^{-}_t = N_t - k$. If $k = N_t$, then there are no non-viable candidates, and the verifier-selection failure probability is zero. This case is also handled by (18) because $1 - b_t(\lambda; \mathcal{H}_t)^0 = 0$.

For $1 \leq k \leq N_t - 1$, Theorem 1 gives, for any realized mixed candidate set $\mathcal{C}_t$,

$$\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_{t} \;\middle|\; \mathcal{C}_t\right) \geq \Phi\left(\frac{\lambda}{\sigma_t}\right)^k \left[1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t) + \lambda}{\sigma_t}\right)^{N_t - k}\right]. \tag{18}$$

Using the envelope assumption $\Gamma_t(\mathcal{C}_t) \leq \overline{\Gamma}_t(\mathcal{H}_t)$, and the monotonicity of Φ, we obtain $\Phi\left(\frac{\Gamma_t(\mathcal{C}_t)+\lambda}{\sigma_t}\right) \leq b_t(\lambda; \mathcal{H}_t)$. Therefore,

$$\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_{t} \;\middle|\; K_t = k, \mathcal{H}_t\right) \geq a_t(\lambda)^k \left[1 - b_t(\lambda; \mathcal{H}_t)^{N_t - k}\right]. \tag{19}$$

Averaging over $K_t \sim \text{Binomial}(N_t, p_t(\mathcal{H}_t))$ gives

$$\begin{aligned}
\Pr\left(\widehat{\tau}^{(t)}_{\text{greedy}} \notin \widetilde{\mathcal{P}}^{\star}_{t} \;\middle|\; \mathcal{H}_t\right) &\geq \Pr(K_t = 0 \mid \mathcal{H}_t) + \sum_{k=1}^{N_t} \Pr(K_t = k \mid \mathcal{H}_t) a_t(\lambda)^k \left[1 - b_t(\lambda; \mathcal{H}_t)^{N_t - k}\right] \\
&= (1 - p_t(\mathcal{H}_t))^{N_t} + \sum_{k=1}^{N_t} \binom{N_t}{k} p_t(\mathcal{H}_t)^k \left(1 - p_t(\mathcal{H}_t)\right)^{N_t - k} a_t(\lambda)^k \left[1 - b_t(\lambda; \mathcal{H}_t)^{N_t - k}\right]. \quad (20)
\end{aligned}$$

Since the bound holds for every $\lambda \in \mathbb{R}$, taking the supremum over λ gives the final claim. □

C.3 Proof of Corollary 1

Corollary 1. (Depth-wise accumulation under Markovian generation). *Under the assumptions of Theorem 4 for all $t \in [T_{\max}]$, let*

$$\mathcal{S}_t = \left\{\widehat{\tau}^{(t)}_{\text{greedy}} \in \widetilde{\mathcal{P}}^{\star}_{t}\right\}$$

denote the event that greedy search remains viable through depth t. Define $\delta^{\star}_t(\mathcal{H}_t) = \sup_{\lambda \in \mathbb{R}} \delta_t(\lambda; \mathcal{H}_t)$ on $\mathcal{S}_{t-1}$ and $\delta^{\star}_t(\mathcal{H}_t) \triangleq 0$ on $\mathcal{S}^c_{t-1}$. Then,

$$\Pr(\mathcal{S}_{T_{\max}}) \leq \mathbb{E}\left[\prod_{t=1}^{T_{\max}} \left(1 - \delta^{\star}_t(\mathcal{H}_t)\right)\right].$$

Proof. By Theorem 4, on any surviving history $\mathcal{H}_t$, $\Pr(\mathcal{S}^c_t \mid \mathcal{H}_t, \mathcal{S}_{t-1}) \geq \delta^{\star}_t(\mathcal{H}_t)$. Equivalently, $\Pr(\mathcal{S}_t \mid \mathcal{H}_t, \mathcal{S}_{t-1}) \leq 1 - \delta^{\star}_t(\mathcal{H}_t)$.

Using the tower property recursively yields

$$\Pr(\mathcal{S}_{T_{\max}}) \leq \mathbb{E}\left[\prod_{t=1}^{T_{\max}} \left(1 - \delta^{\star}_t(\mathcal{H}_t)\right)\right]. \tag{21}$$

Taking complements gives the claimed failure lower bound. □

C.4 Proof of Theorem 2

Theorem 2. (Worst-case aggregate error under RKHS smoothness). *If the estimation error is induced by $e \in \Phi_k$ with $\|e\|_{\Phi_k} \leq B$, then for every $q \in \{0,1\}^N$,*

$$\sup_{\|e\|_{\Phi_k} \leq B} \left| \sum_{i=1}^{N} q_i e(\tau_i) \right| = B\sqrt{q^\top K q}. \tag{5}$$

Proof. By the reproducing property, $e(\tau_i) = \langle e, k(\tau_i, \cdot) \rangle_{\Phi_k}$. Thus, we have

$$\sum_{i=1}^{N} q_i e(\tau_i) = \left\langle e, \sum_{i=1}^{N} q_i k(\tau_i, \cdot) \right\rangle_{\Phi_k}. \tag{22}$$

By applying Cauchy–Schwarz, we get

$$\left| \sum_{i=1}^{N} q_i e(\tau_i) \right| \leq \|e\|_{\Phi_k} \left\| \sum_{i=1}^{N} q_i k(\tau_i, \cdot) \right\|_{\Phi_k}. \tag{23}$$

In (23), the second factor is exactly

$$\left\| \sum_{i=1}^{N} q_i k(\tau_i, \cdot) \right\|_{\Phi_k}^2 = \sum_{i,j=1}^{N} q_i q_j k(\tau_i, \tau_j) = q^\top K q. \tag{24}$$

Therefore, we have

$$\left| \sum_{i=1}^{N} q_i e(\tau_i) \right| \leq B\sqrt{q^\top K q}. \tag{25}$$

Equality is achieved by choosing e proportional to $\sum_i q_i k(\tau_i, \cdot)$, which proves (5).

□

C.5 Proof of Theorem 3

Theorem 3. (End-to-end approximation guarantee). *Let $q^\star \in \arg\max_{q \in \mathcal{Q}} J(q)$, and let $q^\dagger$ be the output of Algorithm 1. Let us assume that for every $\eta \in \mathcal{G}_\xi$, the randomized greedy warm start returns a set q_η^{rg} satisfying $\mathbb{E}\left[J_\eta\left(q_\eta^{\mathrm{rg}}\right)\right] \geq \rho \max_{q \in \mathcal{Q}} J_\eta(q)$ and that the subsequent pairwise-swap local search returns $q_\eta^\dagger$ with $J_\eta(q_\eta^\dagger) \geq J_\eta\left(q_\eta^{\mathrm{rg}}\right)$. Then, it holds that*

$$\mathbb{E}[J(q^\dagger)] \geq \rho \left(J(q^\star) - \frac{\tilde{B} m \xi^2}{2(1+\xi)} \right). \tag{10}$$

Proof. Let $q^\star$ be optimal for the original objective J, and let $\eta^\star \triangleq \sqrt{q^{\star\top} K q^\star} \in [\sqrt{m}, m]$. Since $\mathcal{G}_\xi$ is a multiplicative grid on this interval, there exists $t^\star$ such that $\eta^\star \leq \eta_{t^\star} \leq (1+\xi)\eta^\star$. Using the variational representation in (8), we get $J(q^\star) = J_{\eta^\star}(q^\star)$ and $J_{\eta_{t^\star}}(q^\star) = s^\top q^\star - \frac{\tilde{B}}{2\eta_{t^\star}} q^{\star\top} K q^\star - \frac{\tilde{B}}{2}\eta_{t^\star}$. Writing $z^\star \triangleq q^{\star\top} K q^\star$ and using $\eta^\star = \sqrt{z^\star}$, we obtain

$$J(q^\star) - J_{\eta_{t^\star}}(q^\star) = \tilde{B}\left[\frac{z^\star}{2\eta_{t^\star}} + \frac{\eta_{t^\star}}{2} - \sqrt{z^\star} \right]. \tag{26}$$

Let $r \triangleq \eta_{t^\star}/\eta^\star \in [1, 1+\xi]$. Then, we get

$$\frac{z^\star}{2\eta_{t^\star}} + \frac{\eta_{t^\star}}{2} - \sqrt{z^\star} = \frac{\eta^\star}{2}\left(\frac{1}{r} + r - 2 \right) = \frac{\eta^\star}{2} \cdot \frac{(r-1)^2}{r} \leq \frac{\eta^\star \xi^2}{2(1+\xi)} \leq \frac{m\xi^2}{2(1+\xi)}, \tag{27}$$

where the last step uses $\eta^\star \le m$. Hence

$$J_{\eta_{t^\star}}(q^\star) \ge J(q^\star) - \frac{\tilde{B} m \xi^2}{2(1+\xi)}. \tag{28}$$

Let $q^\dagger_{\eta_{t^\star}}$ denote the candidate returned by the inner solver at grid point $\eta_{t^\star}$. By the randomized greedy guarantee and the objective-improving post-processing step, we have

$$\mathbb{E}\left[J_{\eta_{t^\star}}\left(q^\dagger_{\eta_{t^\star}}\right)\right] \ge \mathbb{E}\left[J_{\eta_{t^\star}}\left(q^{\mathrm{rg}}_{\eta_{t^\star}}\right)\right] \ge \rho \max_{q \in Q} J_{\eta_{t^\star}}(q). \tag{29}$$

Since $q^\star$ is feasible, we have

$$\mathbb{E}\left[J_{\eta_{t^\star}}\left(q^\dagger_{\eta_{t^\star}}\right)\right] \ge \rho J_{\eta_{t^\star}}(q^\star). \tag{30}$$

On the other hand, Algorithm 1 returns $q^\dagger \in \arg\max_{\eta \in G_\xi} J(q^\dagger_\eta)$. Therefore, pointwise, we have

$$J(q^\dagger) \ge J\left(q^\dagger_{\eta_{t^\star}}\right). \tag{31}$$

Moreover, since $J(q) = \max_{\eta>0} J_\eta(q)$, we have

$$J\left(q^\dagger_{\eta_{t^\star}}\right) \ge J_{\eta_{t^\star}}\left(q^\dagger_{\eta_{t^\star}}\right). \tag{32}$$

Combining the previous inequalities and taking expectations gives

$$\mathbb{E}[J(q^\dagger)] \ge \rho J_{\eta_{t^\star}}(q^\star). \tag{33}$$

Using (28), we obtain

$$\mathbb{E}[J(q^\dagger)] \ge \rho \left(J(q^\star) - \frac{\widetilde{B} m \xi^2}{2(1+\xi)} \right), \tag{34}$$

which proves (10).

□

D Additional theoretical results

D.1 Beam-level reward hacking via deterministic cutoff

Corollary 2. *Under Assumption 1 and any fixed $\lambda \in \mathbb{R}$, let $\Gamma_t(\mathcal{C}_t) = \sup_{\tau_\star \in \mathcal{C}_t^\star, \tau_- \in \mathcal{C}_t^-} \{V(\tau_\star) + \beta_t(\tau_\star) - V(\tau_-) - \beta_t(\tau_-)\}$, $q_t(\lambda) = \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star}$, and $r_t(\lambda) = 1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t)+\lambda}{\sigma_t}\right)$. When $N_t^\star \ge 1$ and $N_t^- \ge w_t$, the probability that the beam of width w_t contains no viable prefix satisfies*

$$\Pr\left(\mathcal{B}_t \cap \widetilde{\mathcal{P}}_t^\star = \emptyset \middle| \mathcal{C}_t\right) \ge q_t(\lambda) \Pr\left(\mathrm{Binomial}(N_t^-, r_t) \ge w_t\right), \tag{35}$$

where $\Pr\left(\mathrm{Binomial}(N_t^-, r_t) \ge w_t\right) \triangleq \sum_{j=w_t}^{N_t^-} \binom{N_t^-}{j} r_t(\lambda)^j \left(1 - r_t(\lambda)\right)^{N_t^- - j}$.

Proof. Let $M_t^\star = \max_{\tau_\star \in \mathcal{C}_t^\star} \widehat{V}(\tau_\star)$ be the largest verifier score among viable prefixes. Also, for any fixed $\lambda \in \mathbb{R}$, set $a_t = \sup_{\tau'_\star \in \mathcal{C}_t^\star} \{V_t(\tau'_\star) + b_t(\tau'_\star)\} + \lambda$. Beam search fails at step t if the top w_t prefixes selected by $\widehat{V}$ are all non-viable. A sufficient condition for this failure is $M_t^\star \le a_t$ and $\left|\left\{\tau_- \in \mathcal{C}_t^- : \widehat{V}(\tau_-) > a_t\right\}\right| \ge w_t$. Indeed, on this event, at least w_t non-viable prefixes have verifier score at least as large as every viable prefix. Thus the top-w_t beam can be filled entirely by non-viable prefixes.

By the proof of Theorem 1, we have

$$\Pr(M_t^\star \le a_t \mid \mathcal{C}_t) \ge \Phi\left(\frac{\lambda}{\sigma_t}\right)^{N_t^\star} = q_t(\lambda) \tag{36}$$

and

$$\Pr(\widehat{V}(\tau_-) > a_t \mid \mathcal{C}_t) \geq 1 - \Phi\left(\frac{\Gamma_t(\mathcal{C}_t) + \lambda}{\sigma_t}\right) = r_t(\lambda). \tag{37}$$

Since the non-viable noises are independent by Assumption 1, the number of non-viable prefixes exceeding the cutoff stochastically dominates $\mathrm{Binomial}\left(N_t^-, r_t(\lambda)\right)$. Consequently, we get

$$\Pr\left(\left|\left\{\tau_- \in \mathcal{C}_t^- : \widehat{V}(\tau_-) > a_t\right\}\right| \geq w_t \middle| \mathcal{C}_t\right) \geq \Pr\left(\mathrm{Binomial}(N_t^-, r_t) \geq w_t\right). \tag{38}$$

As in Theorem 1, the event $\{M_t^\star \leq a_t\}$ and the event that at least w_t non-viable prefixes exceed a_t depends only on the non-viable noises being independent. Hence, we get our desired result as

$$\begin{aligned}
&\Pr\left(\mathcal{B}_t \cap \widetilde{\mathcal{P}}_t^\star = \emptyset \middle| \mathcal{C}_t\right) && (39)\\
&\quad \geq \Pr\left(M_t^\star \leq a_t, \left|\left\{\tau_- \in \mathcal{C}_t^- : \widehat{V}(\tau_-) > a_t\right\}\right| \geq w_t \middle| \mathcal{C}_t\right) && (40)\\
&\quad = \Pr(M_t^\star \leq a_t \mid \mathcal{C}_t) \Pr\left(\left|\left\{\tau_- \in \mathcal{C}_t^- : \widehat{V}(\tau_-) > a_t\right\}\right| \geq w_t \middle| \mathcal{C}_t\right) && (41)\\
&\quad \geq q_t(\lambda) \Pr\left(\mathrm{Binomial}(N_t^-, r_t) \geq w_t\right). && (42)
\end{aligned}$$

□

D.2 Kernel penalty breaks narrow beam-search failures

Theorem 5. *Let us fix a depth t with $C_t^\star \neq \emptyset$. Let $\widehat{B}_t^{\mathrm{prm}}$ be the top-w_t PRM beam, with $w_t \geq 2$, and suppose it fails (i.e., $\widehat{B}_t^{\mathrm{prm}} \subseteq C_t^-$). For $|S_t| = w_t$, we let*

$$J_t(S_t) = \sum_{\tau_i^{(t)} \in S_t} s_{t,i} - \widetilde{B}_t \sqrt{q_t(S_t)^\top K q_t(S_t)}, \tag{43}$$

where $\widetilde{B}_t > 0$, $s_{t,i} = \widehat{V}(\tau_i^{(t)})$, and $q_t(S_t) \in \{0,1\}^{|C_t|}$ is the indicator vector of S_t. Let us assume K is symmetric, $K_{ii} = 1$, and $0 \leq K_{ij} \leq 1$.

Let

$$\tau_{a_t}^{(t)} \in \arg\min_{\tau_i^{(t)} \in \widehat{B}_t^{\mathrm{prm}}} s_{t,i}, \qquad \tau_{v_t}^{(t)} \in \arg\max_{\tau_i^{(t)} \in C_t^\star} s_{t,i}, \tag{44}$$

and define the PRM edge gap

$$\Delta_t^{\mathrm{edge}} = s_{t,a_t} - s_{t,v_t}. \tag{45}$$

Suppose the failed PRM beam is more similar to its weakest selected prefix than to the best viable replacement; that is, $K_{a_t i} \geq \alpha_t$ and $K_{v_t i} \leq \beta_t$ for all $\tau_i^{(t)} \in \widehat{B}_t^{\mathrm{prm}} \setminus \{\tau_{a_t}^{(t)}\}$ given $\alpha_t > \beta_t$. In this case, if

$$0 \leq \Delta_t^{\mathrm{edge}} < \widetilde{B}_t \frac{w_t - 1}{w_t}(\alpha_t - \beta_t), \tag{46}$$

then $\widehat{B}_t^{\mathrm{prm}}$ is not a maximizer of J_t.

Proof. Since $\widehat{B}_t^{\mathrm{prm}}$ is the top-w_t PRM beam and $\tau_{v_t}^{(t)} \notin \widehat{B}_t^{\mathrm{prm}}$, the weakest selected score is at least s_{t,v_t}. Hence we have $\Delta_t^{\mathrm{edge}} = s_{t,a_t} - s_{t,v_t} \geq 0$.

Let

$$S_t = \widehat{B}_t^{\mathrm{prm}}, \qquad S_t' = \left(\widehat{B}_t^{\mathrm{prm}} \setminus \{\tau_{a_t}^{(t)}\}\right) \cup \{\tau_{v_t}^{(t)}\}, \tag{47}$$

and we denote

$$R_t(S_t) = \sqrt{q_t(S_t)^\top K q_t(S_t)}. \tag{48}$$

Then, we get

$$J_t(S_t') - J_t(S_t) = -\Delta_t^{\mathrm{edge}} + \widetilde{B}_t \left(R_t(S_t) - R_t(S_t')\right). \tag{49}$$

It remains to lower-bound the reduction in kernel penalty. Let us define

$$G_t = \sum_{\tau_i^{(t)} \in S_t \setminus \{\tau_{a_t}^{(t)}\}} (K_{a_t i} - K_{v_t i}) \,. \tag{50}$$

By the assumption about α_t, β_t, we have

$$G_t \geq (w_t - 1)(\alpha_t - \beta_t). \tag{51}$$

The swap only changes kernel terms involving $\tau_{a_t}^{(t)}$ or $\tau_{v_t}^{(t)}$. Since K is symmetric and $K_{a_t a_t} = K_{v_t v_t} = 1$,

$$R_t(S_t)^2 - R_t(S_t')^2 = 2G_t. \tag{52}$$

Also, since $0 \leq K_{ij} \leq 1$, every set T_t of size w_t satisfies $R_t(T_t) \leq w_t$. Therefore, we have

$$R_t(S_t) - R_t(S_t') = \frac{R_t(S_t)^2 - R_t(S_t')^2}{R_t(S_t) + R_t(S_t')} = \frac{2G_t}{R_t(S_t) + R_t(S_t')} \geq \frac{G_t}{w_t} \geq \frac{w_t - 1}{w_t}(\alpha_t - \beta_t). \tag{53}$$

Thus, it holds that

$$J_t(S_t') - J_t(S_t) \geq -\Delta_t^{\mathrm{edge}} + \widetilde{B}_t \frac{w_t - 1}{w_t}(\alpha_t - \beta_t). \tag{54}$$

Therefore, if

$$\Delta_t^{\mathrm{edge}} < \widetilde{B}_t \frac{w_t - 1}{w_t}(\alpha_t - \beta_t), \tag{55}$$

then $J_t(S_t') > J_t(S_t)$. Hence $\widehat{B}_t^{\mathrm{prm}}$ cannot be a maximizer of J_t. □

D.3 Shifted nonnegative surrogates

Corollary 3 (End-to-end guarantee with shifted nonnegative surrogates)**.** *Assume the same conditions as in Theorem 3, without the non-negativity of $J_\eta(q)$. For each grid point $\eta \in G_\xi$, let $Q_{\leq m} \triangleq \{q \in \{0,1\}^N : \mathbf{1}^\top q \leq m\}$ and $L_\eta \triangleq \min_{q \in Q_{\leq m}} J_\eta(q)$.*

Let $q^\dagger$ be the final output selected by Algorithm 1 according to the original objective J. Let $q^\star \in \arg\max_{q \in \mathcal{Q}} J(q)$ and define $L_{\mathcal{H}} \triangleq \min_{\eta \in \mathcal{G}_\xi} L_\eta$. Then, it holds that

$$\mathbb{E}[J(q^\dagger)] \geq \rho \left(J(q^\star) - \frac{\tilde{B} m \xi^2}{2(1+\xi)} \right) + (1-\rho) L_{\mathcal{H}}.$$

Proof. For any fixed grid point η, the shifted objective $\bar{J}_\eta(q) = J_\eta(q) - L_\eta$ is nonnegative on $Q_{\leq m}$, and hence on every intermediate set considered by the randomized greedy procedure.

By the assumed multiplicative guarantee in Theorem 3, we have

$$\mathbb{E}[\bar{J}_\eta(q_\eta^{\mathrm{rg}})] \geq \rho \max_{q \in \mathcal{Q}} \bar{J}_\eta(q) = \rho \left(\max_{q \in \mathcal{Q}} J_\eta(q) - L_\eta \right). \tag{56}$$

Since $\bar{J}_\eta(q) = J_\eta(q) - L_\eta$, this implies $\mathbb{E}[J_\eta(q_\eta^{\mathrm{rg}})] \geq \rho \max_{q \in \mathcal{Q}} J_\eta(q) + (1-\rho) L_\eta$. The local-search post-processing only improves J_η, so

$$\mathbb{E}[J_\eta(q_\eta^\dagger)] \geq \rho \max_{q \in \mathcal{Q}} J_\eta(q) + (1-\rho) L_\eta. \tag{57}$$

Let $q^\star \in \arg\max_{q \in \mathcal{Q}} J(q)$ and $\alpha^\star = \sqrt{q^{\star\top} K q^\star}$, and then consider $\eta_\xi^\star \in \mathcal{G}_\xi$ such that

$$\alpha^\star \leq \eta_\xi^\star \leq (1+\xi)\alpha^\star. \tag{58}$$

Then, by using the same technique for bounding the multiplicative grid as in Theorem 3, we have

$$J(q^\star) - J_{\eta_\xi^\star}(q^\star) = \tilde{B} \left(\frac{(\eta_\xi^\star - \alpha^\star)^2}{2\eta_\xi^\star} \right) \leq \frac{\tilde{B}\xi^2}{2(1+\xi)} \alpha^\star \leq \frac{\tilde{B} m \xi^2}{2(1+\xi)}. \tag{59}$$

Therefore,

$$\max_{q\in\mathcal{Q}} J_{\eta_\xi^\star}(q) \geq J_{\eta_\xi^\star}(q^\star) \geq J(q^\star) - \frac{\tilde{B}m\xi^2}{2(1+\xi)}. \tag{60}$$

Applying the shifted guarantee at $\eta_\xi^\star$, and using $L_{\eta_\xi^\star} \geq L_{\mathcal{H}}$, gives

$$\mathbb{E}[J_{\eta_\xi^\star}(q^\dagger_{\eta_\xi^\star})] \geq \rho\left(J(q^\star) - \frac{\tilde{B}m\xi^2}{2(1+\xi)}\right) + (1-\rho)L_{\mathcal{H}}. \tag{61}$$

Finally, Algorithm 1 returns the candidate with the largest original objective J among all grid solutions, and $J(q) \geq J_\eta(q)$ for every q and η. Thus,

$$J(q^\dagger) \geq J(q^\dagger_{\eta_\xi^\star}) \geq J_{\eta_\xi^\star}(q^\dagger_{\eta_\xi^\star}), \tag{62}$$

and the claimed bound follows after taking expectations. □

Corollary 3 shows why the nonnegativity shift cannot be treated as without loss for a multiplicative approximation guarantee. Shifting preserves the optimizer and the marginal gains, but it changes the scale on which the multiplicative approximation is measured. The resulting guarantee therefore contains the additional additive term $(1-\rho)L_{\mathcal{H}}$, which disappears only when the unshifted surrogates are already nonnegative. The shift is used only for the approximation analysis. The implementation does not require computing L_η, since subtracting a constant from J_η does not change marginal gains, local swaps, or the selected set.

E Literature review

E.1 Extended related work

Step-level guidance with PRMs. A growing body of work uses step-level evaluation to improve mathematical reasoning at inference time, including self-evaluation-guided beam search (Xie et al., 2023), AlphaZero-style tree search (Feng et al., 2023), step-level beam search (Chen et al., 2024), and actor–critic style decoding (Dong et al., 2025). Across these methods, a generator proposes candidate reasoning steps and a learned evaluator scores partial trajectories to guide search, improving performance by allocating compute more selectively across candidate chains.

Our work is most closely related to recent *self-improvement* approaches that address failures from reward misspecification. ReST-MCTS* (Zhang et al., 2024) combines PRM-guided search with iterative updates to the policy and value model; V-STaR (Hosseini et al., 2024) similarly improves output verification through iterative refinement; and PRIME (Cui et al., 2025) mitigates reward hacking by updating rewards from rollout outcomes. These methods reduce the mismatch between search and evaluation by adapting the policy, verifier, or reward model to deployment-induced trajectories. However, they rely on additional training, often incur substantial computational overhead, and in some cases require outcome supervision or ground-truth answers (Zhang et al., 2024; Hosseini et al., 2024; Cui et al., 2025).

By contrast, we study over-optimization in PRM-guided search from a robust optimization perspective at inference time. Rather than mitigating reward misspecification through retraining or test-time adaptation, we seek search procedures that remain less sensitive to the PRM error under a fixed generator and fixed PRM. To our knowledge, prior work has largely overlooked this robust-inference perspective on PRM-guided mathematical reasoning.

E.2 Comparison with MBR and diversity-based decoding

Maximin search belongs to a broader family of inference-time methods that use the geometry of a candidate set to avoid depending too strongly to a single imperfect score. The key distinction is what decision geometry is used to regularize. Minimum Bayes Risk (MBR) uses geometry to choose a representative final output. Diversity-promoting methods use geometry to reduce redundancy among candidates. Maximin search uses geometry to make intermediate pruning robust to verifier misspecification.

Maximin vs MBR. We first contrast maximin search with MBR decoding. Given completed candidates $\mathcal{Y} = \{y_i\}_{i=1}^N$, sample-based MBR selects

$$\hat{y}_{\mathrm{MBR}} \in \arg\max_{y \in \mathcal{Y}} \frac{1}{N} \sum_{i=1}^{N} U(y, y_i). \tag{63}$$

When the utility is the negative of a metric distance, this is equivalent to minimizing empirical expected distance, so MBR selects a Fréchet median of the empirical candidate distribution $\hat{p}_N = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$, i.e., an empirical 1-median (Jinnai et al., 2025). This view also underlies MBR-regularized Best-of-N decoding, where an MBR term discourages selecting high-reward outputs that are far from the reference-policy sample distribution (Jinnai et al., 2025).

Thus, MBR and maximin search share a common motivation: both use candidate-set geometry to reduce the risk of over-optimizing a noisy or misspecified reward. However, they use geometry in different ways. MBR is a consensus rule over completed outputs. It favors candidates that are central under the empirical output distribution. This is desirable when centrality correlates with correctness, because isolated high-reward outliers are suppressed. In mathematical reasoning, however, centrality can also be misleading: many sampled chains may share the same plausible but incorrect misconception, while a correct continuation may be rare. A purely central candidate can therefore correspond to a dominant wrong reasoning mode.

Maximin search addresses a different problem. It does not choose a single central completed output. Instead, it selects a subset of partial reasoning prefixes to preserve for future expansion. The goal is to avoid concentrating the search frontier in one high-similarity region when the verifier may be wrong in a coherent way on that region. In this sense, maximin search regularizes the pruning step rather than the final selection step. It preserves optionality under verifier uncertainty, whereas MBR aggregates evidence to choose a representative final candidate.

Maximin vs diversity-based decoding. Maximin search is also distinct from generic diversity regularization. Diverse beam search introduces diversity as a search heuristic, often at the lexical or semantic level, to reduce redundancy among beams (Vijayakumar et al., 2016). Determinantal point process (DPP) methods similarly select high-quality diverse subsets by optimizing a determinantal objective, typically rewarding large volume or near-orthogonality in the feature space (Kulesza et al., 2012; Meister et al., 2021). These methods are natural baselines because they also discourage redundant candidates. However, their diversity terms are not derived from a model of verifier error.

The maximin penalty has a different interpretation. Under our RKHS misspecification model, the term $\sqrt{q^\top K q}$ is proportional to the worst-case coherent verifier error over the selected trajectories. The regularization coefficient λ therefore plays the role of an uncertainty radius, not merely a diversity weight. This interpretation yields a principled way to choose λ from an estimated level of verifier misspecification, rather than treating it as an arbitrary diversity hyperparameter. The resulting subset-selection problem is

$$\max_{q \in \{0,1\}^N,\ \mathbf{1}^\top q = m} s^\top q - \lambda \sqrt{q^\top K q}, \tag{64}$$

where s contains the verifier scores and K measures similarity between prefixes. The penalty controls the RKHS norm of the selected empirical measure, which is exactly the quantity that bounds worst-case aggregate misspecification error on the retained trajectories.

Hence the selected set under maximin search is not merely diverse or representative, unlike MBR, diverse beam search, or DDP; it is the highest-scoring trajectories whose aggregate score remains stable under structured step-level reward misspecification.

F Additional experimental details

Baseline definitions. We evaluate all methods under matched inference budgets of $N \in \{16, 64\}$ trajectories. The budget is the maximum number of parallel trajectories maintained during decoding: the beam width for likelihood beam search, the sample count for Best-of-N methods, and $4m$ for PRM-guided search, where the branch factor is fixed at 4 and $m = \lfloor N/4 \rfloor$.

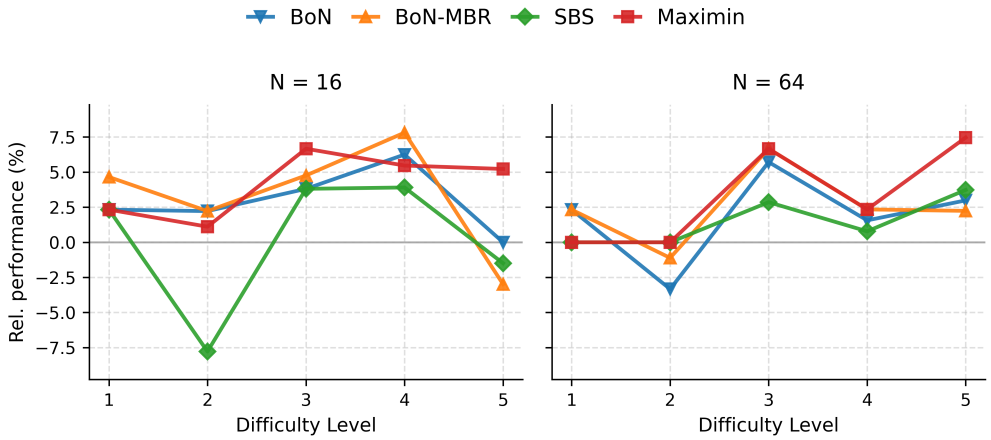


Figure 5: **Performance relative to beam search across MATH-500 difficulty levels (1–5) at $N = 16$ (left) and $N = 64$ (right).** Curves report accuracy differences in percentage points; positive values indicate improvement over beam search.

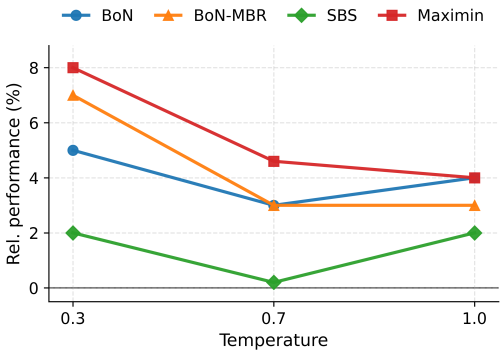


Figure 6: **Temperature sensitivity.** Accuracy difference from beam search across sampling temperatures.

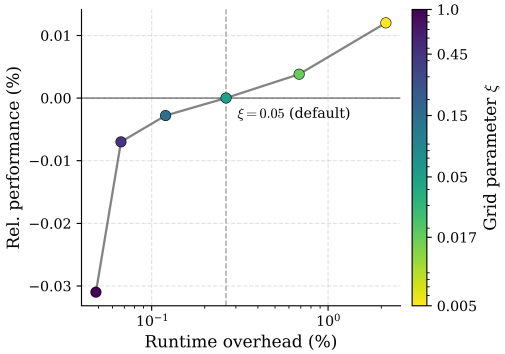


Figure 7: **Grid-resolution trade-off.** Accuracy change relative to $\xi = 0.05$ versus runtime overhead over SBS.

- **No external supervision:** greedy decoding and likelihood beam search. Greedy decoding selects the highest-probability continuation, while beam search retains partial trajectories according to generator likelihood.
- **Outcome-level supervision:** Best-of-N (BoN) and MBR-BoN (Jinnai et al., 2025). BoN selects the sampled trajectory with the highest outcome score. MBR-BoN uses $\widehat{V}(\tau) + 0.1\, N^{-1} \sum_i U(\tau, \tau_i)$, where $U(\tau, \tau') = \mathrm{cossim}(\mathrm{emb}(\tau), \mathrm{emb}(\tau'))$.
- **Step-level supervision:** SBS (Chen et al., 2024), which retains the top-m partial trajectories according to PRM score at each pruning step.

Performance by problem difficulty. Figure 5 shows that the methods are closer on easier MATH-500 problems, while maximin performs best on Level 5 for both budgets. The larger gains on harder problems are consistent with verifier uncertainty becoming more consequential in difficult, multi-step reasoning.

Sensitivity and computational trade-offs. Maximin remains strongest across temperatures $[0.3, 0.7, 1.0]$, improving over beam search by $+8.0$, $+4.6$, and $+4.0$ percentage points, respectively. Performance is also stable across moderate grid resolutions: the default $\xi = 0.05$ incurs only 0.26% runtime overhead, while finer grids provide small accuracy gains at higher cost and very coarse grids eventually degrade performance.

Median pairwise-distance heuristic. Given a set of trajectories $\{\tau_i\}_{i=1}^n$, we use the last hidden state of the PRM as an embedding $e_i = \mathrm{emb}(\tau_i) \in \mathbb{R}^d$. Then, the kernel is defined as $d^2(\tau_i, \tau_j) = 2 - 2\frac{e_i^\top e_j}{\|e_i\|_2 \|e_j\|_2}$, with the bandwidth determined by the median pairwise distance excluding diagonal self-distances:

$$\gamma = 1/(2\sigma^2), \quad \sigma = \sqrt{\mathrm{median}\{d^2(\tau_i, \tau_j) : i \neq j\}}. \tag{65}$$

We clamp the median distance and σ^2 by a constant 10^{-6} for numerical stability.

Generation configurations. For both generators, we use top-p sampling with $p = 0.9$ and temperature 0.7. Also, for a step-level generation, we use a stop token of "$\backslash\mathrm{n}\backslash\mathrm{n}$" with the maximum number of tokens in each step being 128. We set the maximum reasoning depth as 30. To ensure a fair comparison across methods, we halt any trajectory once its total generated length exceeds 2,048 tokens. Finally, for PRM-guided search methods, we use a constant branching factor of 4 at every reasoning depth. Thus, under $4m$ number of maximum trajectories, each search state maintains m parent trajectories and expands each of them with four continuations.

We use zero-shot chain-of-thought prompts for the generators, with an output format instruction. Exact prompts are given below.

Prompt for Qwen2.5-Math-1.5B

```
Below is an instruction that describes a task. Write a response
↪  that appropriately completes the request.

### Instruction:
{question}

### Response: Please reason step by step, and put your final
↪  answer within \\boxed{}.
```

Prompt for Phi-3.5-mini-instruct

```
<|system|>You are a helpful assistant<|end|>
<|user|>Below is an instruction that describes a task. Write a
↪  response that appropriately completes the request.

### Instruction:
{question}

### Response: Please reason step by step, and put your final
↪  answer within \\boxed{}.

<|end|>
<|assistant|>
```

We used the following constants for Algorithm 1. The outer multiplicative grid is set based on $\xi = 0.05$, with endpoints $\eta_{\min} = \sqrt{m}$ and $\eta_{\max} = m$. For each fixed η, we initialized the inner solver using the randomized greedy warm start as described in §5.2, together with deterministic greedy and random initializations. Each initialization was refined by 1-swap local search with at most 30 improvement passes and improvement tolerance 10^{-12}. Among the resulting local optima for a fixed η, we kept the one with the largest fixed-η surrogate value.

AIME standard deviations. In Table 1, we list the average accuracy for AIME datasets. Next, we give the standard deviations of the AIME results in Table 2.

Table 2: **Standard deviations of benchmark results on AIME datasets.** We report the standard deviations corresponding to the AIME results in Table 1.

Generator	Method	AIME'24		AIME'25	
		$N = 16$	$N = 64$	$N = 16$	$N = 64$
Qwen2.5-Math-1.5B	BoN	3.34	3.34	3.85	5.09
	BoN-MBR	1.93	3.34	3.85	3.85
	SBS	1.92	5.09	3.34	3.85
	Maximin	3.34	3.34	3.85	1.92
Phi-3.5-mini-instruct	BoN	3.34	1.93	1.92	3.34
	BoN-MBR	3.34	3.34	1.92	3.34
	SBS	1.93	1.92	1.92	3.34
	Maximin	1.92	3.85	1.92	1.92