DIRECT PASS@k OPTIMIZATION FOR THOUGHT PRUNING: BALANCING QUALITY AND DIVERSITY IN TEST-TIME REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time scaling has emerged as a key inference strategy for reasoning models, where a model generates a large number of candidate solutions in the hope of finding at least one correct solution. Running all candidate solutions to completion can be computationally demanding, so recent work prunes the candidate pool after generating only a short prefix, committing further computation only to the surviving candidates. Existing pruning strategies divide broadly into two categories: Quality-based methods assign a score to each partial solution and discard those predicted to ultimately lead to an incorrect final solution, while diversity-based methods select partial solutions that follow different reasoning trajectories and are therefore likely to cover different final solutions. The tension between these two objectives is an instance of exploration versus exploitation, yet little to no prior work on pruning exposes this trade-off directly, let alone optimizes it. In this paper, we introduce a pruning procedure that addresses this trade-off directly by optimizing pass@k over the selected solution pool. In particular, it chooses the pool to maximize the probability that at least one of its partial solutions, when continued to completion, yields a correct answer. Because this objective simultaneously rewards high-quality partial solutions and penalizes redundancy among them, quality and diversity are naturally balanced by the criterion itself rather than through separate ad-hoc design choices. On eight open-weight reasoning models, our method outperforms existing quality-based and diversity-based pruning strategies. The gains are particularly pronounced when the pruning procedure keeps a moderate number of partial solutions for continued generation, highlighting the value of directly balancing quality and diversity during pruning.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in complex reasoning, showcasing a strong aptitude for solving complex problems that require multi-step logic [15, 18, 21, 45, 52]. To further enhance these capabilities, test-time scaling has emerged as a prominent strategy, where generating multiple candidate solutions for a single problem allows the model to explore diverse reasoning strategies and achieve substantially better final performance [3, 35, 44, 48]. However, a significant challenge in scaling test-time compute is the prohibitive cost of generating many full solutions, which we refer to as *rollouts* moving forward [23, 33]. There is also the problem of evaluating the finished rollouts in best-of-N methods, which score each rollout with a reward model, and this is computationally demanding on a large pool of long reasoning traces [11, 26, 36]. This computational bottleneck necessitates test-time pruning of partial rollouts before they reach a final answer and terminate. In other words, we generate all N

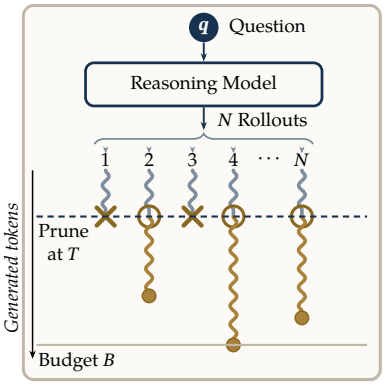
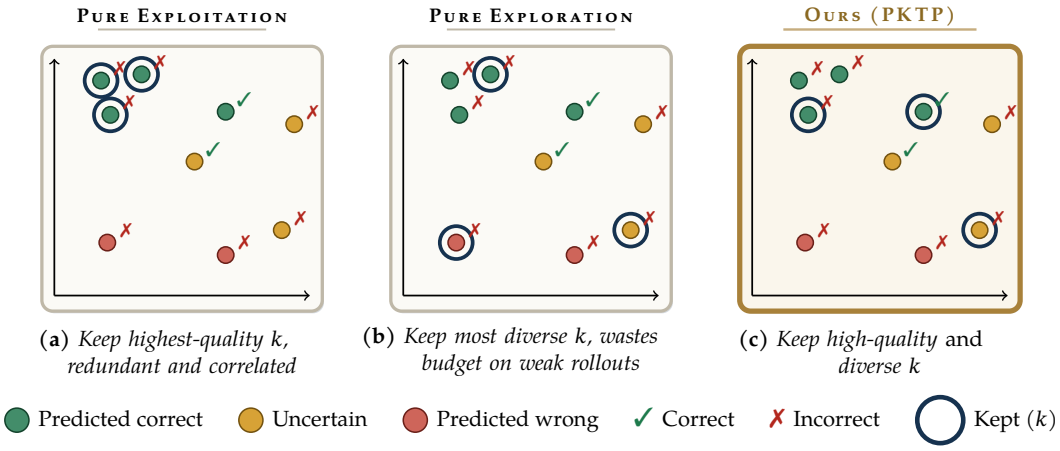


Figure 1: **Test-time pruning.** Only k of the N rollouts (ringed) continue past the cutoff T, up to the budget B.

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.



(a) *Keep highest-quality k, redundant and correlated*

(b) *Keep most diverse k, wastes budget on weak rollouts*

(c) *Keep high-quality and diverse k*

Figure 2: **Conceptual illustration.** The same pool of candidate rollouts, positioned by their partial rollout representations in embedding space, under three pruning strategies. Each rollout is colored by its *predicted* correctness and marked ✓/✗ by its *actual* outcome; the $k = 3$ kept rollouts are ringed. **(a) Pure exploitation** keeps the highest-scoring rollouts, but they are correlated and cluster together, so they miss the correct rollout. **(b) Pure exploration** keeps a maximally diverse subset yet spends its budget on low-quality rollouts, also missing it. **(c) Our method** optimizes $\text{pass@}k$ directly over the kept subset, balancing quality and diversity, and retains the correct rollout.

rollouts only up to a cutoff of T tokens, keep the k most promising of them and running those to the full token budget B, as illustrated in Fig. 1.

One family of methods examines the confidence dynamics of the model during generation, utilizing metrics such as logit entropy or confidence [17, 30, 43, 50]. The underlying intuition is that if a model is overly confident at an intermediate decoding step, extending the reasoning process is a waste of compute; conversely, if the model is entirely lacking in confidence, it is unlikely to reach a correct solution regardless of the additional compute provided, allowing for early termination. Another line of work leverages the model's internal state, specifically the hidden states of its transformer blocks, to identify which rollouts are worth continuing [1, 54]. By "probing" these internal states, one can predict whether the model will ultimately arrive at a correct solution, effectively framing the pruning problem as a classification task. This allows for the early discarding of partial rollouts that are unlikely to yield a correct final answer. In parallel, a separate line of research prioritizes representation exploration (RepExp) to select a subset of rollouts that cover as many distinct reasoning strategies as possible [39]. By selecting partial rollouts whose corresponding embeddings are maximally dissimilar to those of already-selected rollouts, RepExp ensures that the retained subset explores a diverse set of reasoning trajectories.

These approaches, however, sit at the two ends of a spectrum, and neither directly optimizes the quantity we actually care about. The problem with probing methods is that the rollouts they rank highest tend to be similar to one another. As a result, the probe may be confidently wrong about all of the kept rollouts at once, and none of them reaches a correct answer. This is the problem of pure exploitation (Fig. 2a). RepExp, on the other hand, finds diverse rollouts, but because it ignores the quality of the rollouts, it can spend much of its budget on rollouts that are unlikely to succeed (Fig. 2b). These limitations motivate selecting the rollouts by the metric we actually want to optimize, choosing the set S of k rollouts that maximizes $\text{pass@}k$, that is, $\max_{|S|=k} \text{pass@}k(S)$, which we formalize in Eq. 2. Optimizing this objective directly keeps the selected rollouts diverse enough to avoid shared failures while still favoring high-quality ones (Fig. 2c).

To this end, we propose **PKTP** (**P**ass@k **T**hought **P**runing), a pruning framework that chooses the kept rollouts by maximizing $\text{pass@}k$ directly. The main difficulty is that, even if the rollouts are sampled independently, their outcomes are not independent (conditioned on the partial observation of the rollout). Hence, $\text{pass@}k$ depends on their joint distribution (i.e., rollouts with similar partial rollouts tend to succeed or fail together). We model this joint distribution with a latent Gaussian model,

in which the correctness of each rollout is obtained by thresholding a Gaussian score. Evaluating pass@k under this model requires an expectation over the latent variables, so we instead derive a closed-form second-order approximation of it. We show that this approximation is submodular, enabling us to approximately solve it using a greedy procedure to find a subset of size k. An important aspect of the proposed framework is that the latent model is trained only once and does not depend on k, and a single greedy pass yields a nested sequence of subsets, so the budget k can be chosen online at inference time without retraining.

Contributions. First, we cast pruning as maximizing the pass@k of the kept set, and we model how similar rollouts fail together. Second, we derive a closed-form, submodular approximation of pass@k that links the probe and RepExp. Third, through experiments on eight reasoning models, we show that PKTP improves over existing pruning methods, especially for moderate values of k.

2 Problem Formulation and Related Work

Setup. Given a question $\boldsymbol{q}$, we draw N candidate reasoning rollouts $y^{(i)} \sim \pi(\cdot \mid \boldsymbol{q})$ from a model π. Each rollout is observed only through its first T tokens, the prefix $y^{(i)}_{1:T}$. We encode this prefix as a representation vector $h_i = h\left(y^{(i)}_{1:T}\right)$, a hidden-state embedding of the reasoning so far. Pruning then keeps $k \le N$ rollouts, runs them to the full token budget B, and discards the rest. A completed rollout $y_{1:B}$ is either correct or not, recorded by the sign $C(y_{1:B}) \in \{-1, +1\}$,

$$C(y_{1:B}) \;=\; \begin{cases} +1, & \text{if } \textsc{Verify}(y_{1:B})\,, \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

Here Verify is a *task-specific checker* that returns true exactly when a completed rollout reaches a correct final answer, for instance by matching a reference solution on a math problem or passing the unit tests on a coding task. We write $C(y^{(i)})$ for the outcome of the i-th rollout once it is run to the budget B.

Representations. To prune partial rollouts, we need to map each prefix to a vector that captures its essential information. One line of work relies on hand-designed features of the rollout and the model's internal states, such as token confidence, the geometry of the hidden-state trajectory, or the convergence of intermediate answers. These features are usually very low dimensional [13, 19, 28, 38, 40]. A second line of work instead keeps the hidden states of the transformer and pools them into a single vector [29, 39, 42], and this is the class we use. In particular, we take h_i to be an exponential moving average (EMA) of the hidden states, which has proven an effective pooling operation [29]. Our improvements are thus shown for this class of embeddings, although our framework can be applied to other embeddings as well. Appendix F gives the details of the representation we use.

Objective. At pruning time the outcomes $C(y^{(i)})$ are still unknown, since each depends on a (random) continuation we have not yet generated. How, then, should we choose the set $S \subseteq \{1, \dots, N\}$ of $|S| = k$ rollouts to keep? Two extreme choices are as follows.

(i) **(Pure exploitation)** Keep in S the rollouts with the highest predicted success $\mathbb{P}\left(C(y^{(i)}) = +1 \mid \boldsymbol{q}, y^{(i)}_{1:T}\right)$, estimated by a linear probe trained on the representations h_i [54], that is, the k rollouts with the largest scores $w^\top h_i$ for a learned weight vector w. We refer to this pruning policy as **Linear Probe**.

(ii) **(Pure exploration)** Build a diverse S, starting from a random rollout and greedily adding to S

$$\arg\max_{i \notin S} \; h_i^\top \left(\tau I + \sum_{j \in S} h_j h_j^\top\right)^{-1} h_i,$$

the rollout whose representation is least explained by those already chosen, where $\tau > 0$ is a regularizer. This is the **RepExp** pruning policy [39].

Neither extreme balances quality against diversity. What we actually want is the set of k rollouts with the best pass@k. Mathematically, we seek the set S most likely to contain a correct rollout,

Pass@k Pruning Objective

$$\max_{S \subseteq \{1,\dots,N\}} \mathbb{P}\left(\sum_{i \in S} \mathbf{1}\{C(y^{(i)}) = +1\} \;\geq\; 1 \;\middle|\; \boldsymbol{q},\, y^{(1)}_{1:T}, \dots, y^{(N)}_{1:T} \right) \quad \text{s.t.} \quad |S| = k. \tag{2}$$

Here, $\mathbf{1}\{\cdot\}$ is the indicator function, i.e., $\mathbf{1}\{True\} = 1$ and $\mathbf{1}\{False\} = 0$. The objective in Eq. 2 is exactly pass@k of the kept pool, the probability that at least one surviving rollout, once run to the budget B, yields a correct answer. We also assume that the final outcomes are binary, although the same approach generalizes to continuous outcomes, where each completed rollout is scored by a reward model and the goal is to maximize the best of the kept rollouts [10], in which case Eq. 2 becomes a max@k objective [41].

Related work. Existing pruning methods score each rollout on its own, using the model's confidence [13, 30], a reward model [36], or a probe on its hidden states [25, 54], or they select a diverse set, as RepExp does [39]. Pass@k has also been used as a training objective for reasoning models [8, 41], but not to decide which partial rollouts to keep. Outside of language models, choosing a set so that at least one item succeeds has been studied in information retrieval and Bayesian optimization [6, 34, 46]. Appendix G gives an extended discussion. In what follows, we discuss how such an objective can be solved.

3 Method

There are some challenges in solving Eq. 2: First, we need to obtain the pass@k of a given set of rollouts. We address this by learning the joint distribution of the rollouts' outcomes under a probabilistic model, from which the pass@k of any set can be estimated. Second, even with such an estimate, finding the subset of k rollouts with the highest estimated pass@k is a mixed-integer program. We can therefore view our framework as two components, learning the joint distribution of the outcomes (Section 3.1), and solving this mixed-integer program at selection time (Section 3.2).

3.1 Probabilistic model

Let $H = [h_1, \dots, h_N] \in \mathbb{R}^{d \times N}$ collect the representations of the N generated partial rollouts, $Q_\theta : \mathbb{R}^d \to \mathbb{R}$ be a learnable mapping, and $Q_\theta(H) = [q_1, \dots, q_N]^\top$ with $q_i := Q_\theta(h_i)$. We model the latent scores of the rollouts as

$$\mathbf{Z} = \underbrace{Q_\theta(H)}_{\text{Rollout Quality}} + \underbrace{H^\top \mathbf{f}}_{\text{Rollouts Correlation}} + \underbrace{\mathbf{m}}_{\text{Outcome Noise}}, \qquad \mathbf{m} \sim \mathcal{N}(0, I_N), \quad \mathbf{f} \sim \mathcal{N}(0, I_d), \tag{3}$$

where $\mathbf{m}$ and $\mathbf{f}$ are independent. Intuitively, $\mathbf{Z}_i$ is a score of how promising rollout i is, so the higher its quality q_i, the higher its score. On top of q_i, the term $h_i^\top \mathbf{f}$ is a random effect shared by rollouts with similar representations, which makes them succeed or fail together. Indeed, for distinct rollouts, $\mathrm{Cov}(\mathbf{Z}_i, \mathbf{Z}_j \mid H) = h_i^\top h_j$, so rollouts with aligned h_i have positively correlated outcomes. Finally, $\mathbf{m}_i$ is noise specific to rollout i.

The correctness $C(y^{(i)})$ of rollout i is then modeled by thresholding its score at zero. More specifically, we define the random variable $\mathbf{C}_i = 2 \times \mathbf{1}\{\mathbf{Z}_i \geq 0\} - 1$. In other words, a positive score implies that the rollout is correct, while a negative score implies that it is incorrect. Since the scores are independent conditionally on $\mathbf{f}$, the likelihood of an observed pattern $c \in \{-1, 1\}^N$ is

$$\Pr[\mathbf{C} = c \mid H] = \mathbb{E}_{\mathbf{f}}\left[\prod_{i=1}^{N} \Phi\big(c_i (q_i + h_i^\top \mathbf{f})\big) \right], \tag{4}$$

where Φ is the standard Gaussian CDF, and the derivation can be found in Appendix A.1.

The mapping Q_θ is trained on questions indexed by ℓ, each consisting of the representations $H^{(\ell)}$ of a pool and its observed outcomes $c^{(\ell)} \in \{-1, 1\}^N$, by maximizing their likelihood under Eq. 4, that

is, by solving

$$\max_{\theta} \sum_{\ell} \log \Pr_{\theta}\big[\mathbf{C} = c^{(\ell)} \mid H^{(\ell)}\big], \tag{5}$$

where $\Pr_\theta$ is the likelihood of Eq. 4 with $q_i = Q_\theta(h_i)$. Solving Eq. 5 directly is hard, since the expectation over $\mathbf{f}$ in Eq. 4 has no closed form. We therefore expand each log-factor to second order in $h_i^\top \mathbf{f}$, which turns the expectation into a Gaussian integral that can be evaluated exactly (Theorem 1).

Theorem 1 (Second-order surrogate). *For a set $S \subseteq [N] := \{1, \dots, N\}$ with outcomes $\mathbf{C}_S = (\mathbf{C}_i)_{i \in S}$ and a pattern $c \in \{-1, 1\}^S$, let ϕ be the standard Gaussian density, $\lambda_i(c_i) = \phi(q_i)/\Phi(c_i q_i)$, $\kappa_i(c_i) = \lambda_i(c_i)\big(\lambda_i(c_i) + c_i q_i\big)$,*

$$\Sigma_S(c) = \sum_{i \in S} \kappa_i(c_i)\, h_i h_i^\top, \qquad b_S(c) = \sum_{i \in S} c_i \lambda_i(c_i)\, h_i.$$

Then, as $\varepsilon \to 0$,

$$\log \Phi\big(c_i(q_i + \varepsilon)\big) = \log \Phi(c_i q_i) + c_i \lambda_i(c_i)\, \varepsilon - \frac{\kappa_i(c_i)}{2} \varepsilon^2 + O(|\varepsilon|^3),$$

and replacing each log-factor of $\Pr[\mathbf{C}_S = c \mid H]$ by this expansion at $\varepsilon = h_i^\top \mathbf{f}$ gives

$$\widetilde{\Pr}[\mathbf{C}_S = c \mid H] = \Big(\prod_{i \in S} \Phi(c_i q_i)\Big) \det\big(I_d + \Sigma_S(c)\big)^{-1/2} \exp\Big(\frac{1}{2}\, b_S(c)^\top \big(I_d + \Sigma_S(c)\big)^{-1} b_S(c)\Big). \tag{6}$$

The proof is in Appendix B.1. The expansion is local, so we use Eq. 6 as a tractable surrogate rather than as an exact identity or a bound. We therefore replace $\Pr_\theta$ in Eq. 5 by $\widetilde{\Pr}_\theta$.

Instead of using h_i directly in the correlation term, one could learn a decoding mapping of the representations, but its helpfulness depends on the amount of data. Since our data is limited, we keep h_i, and for the same reason we use a simple linear model for Q_θ, as in the Linear Probe [54].

3.2 Subset selection

Maximizing pass@k in Eq. 2 is the same as minimizing the probability that every kept rollout fails. Under the latent Gaussian model, and suppressing the conditioning on H, this failure probability is

$$L(S) = \Pr\big[\mathbf{C}_i = -1 \ \forall i \in S\big] = \mathbb{E}_{\mathbf{f} \sim \mathcal{N}(0, I_d)}\Big[\prod_{i \in S} \Phi\big(-(q_i + h_i^\top \mathbf{f})\big)\Big], \tag{7}$$

and we seek the set of k rollouts that minimizes it. Since $L(S)$ has no closed form, we approximate it with Theorem 1.

Corollary 1 (Selection surrogate). *Let $\lambda_i = \phi(q_i)/\Phi(-q_i)$, $\kappa_i = \lambda_i(\lambda_i - q_i)$, $\Sigma_S = \sum_{i \in S} \kappa_i\, h_i h_i^\top$ and $b_S = \sum_{i \in S} \lambda_i\, h_i$. Applying Theorem 1 with $c_i = -1$ for every $i \in S$ gives a surrogate $\widetilde{L}(S)$ of $L(S)$ with*

$$\log \widetilde{L}(S) = \sum_{i \in S} \log \Phi(-q_i) - \frac{1}{2} \log\det(I_d + \Sigma_S) + \frac{1}{2}\, b_S^\top (I_d + \Sigma_S)^{-1} b_S. \tag{8}$$

We therefore minimize $\log \widetilde{L}(S)$ over $|S| = k$. Each term of Eq. 8 has a direct interpretation. The first term favors large q_i, because the baseline failure factor $\Phi(-q_i)$ decreases with q_i. The negative log-determinant rewards a collection of h_i's that covers several directions instead of repeatedly selecting the same direction. The last term is nonnegative and penalizes sets whose weighted vectors $\lambda_i h_i$ add up in the same direction, which the determinant alone cannot detect since it treats h_i and $-h_i$ identically.

Minimizing $\log \widetilde{L}(S)$ exactly over all sets of size k is NP-hard, even when all rollouts have the same quality (Appendix B.4). When the objective is monotone and submodular, however, a greedy optimizer that adds one rollout at a time is guaranteed to come close to the optimum [32]. The exact

pass probability $P(S) := 1 - L(S)$ is known to be monotone and submodular [46], and we show that the same holds for its surrogate $\widetilde{P}(S) := 1 - \widetilde{L}(S)$, so the greedy optimizer and its approximation guarantee apply to our problem.

Theorem 2 (Greedy guarantee). *$\widetilde{P}$ is normalized, monotone, and submodular. Consequently, the set S_k^{greedy} obtained by adding, k times, the candidate that most increases $\widetilde{P}$ satisfies*

$$\widetilde{P}(S_k^{\text{greedy}}) \geq (1 - e^{-1}) \max_{|S| \leq k} \widetilde{P}(S).$$

The proof is in Appendix B.2.

Greedy optimization. The greedy optimizer starts from the empty set and, at each step, adds the candidate $j \notin S$ with the smallest change $\Delta_j(S) = \log \widetilde{L}(S \cup \{j\}) - \log \widetilde{L}(S)$. Since adding a candidate changes Σ_S by a rank-one matrix, this change has a closed form that can be updated cheaply without rebuilding the determinant and the inverse (Appendix A.2). Since the greedy steps do not depend on k, a single pass produces a nested ordering of the rollouts that can be truncated at any budget k, and it costs $O\big(k(Nd + d^2)\big)$ time and $O(Nd + d^2)$ memory.

3.3 Relation to Linear Probe and RepExp

The surrogate makes the quality–diversity trade-off explicit and connects to the two baseline criteria in limiting regimes. If the correlation term is switched off, then $\log \widetilde{L}(S) = \sum_{i \in S} \log \Phi(-q_i)$, so minimizing it keeps the k largest quality scores and recovers a quality-only, probe-style ranking; it coincides with Linear Probe when Q_θ is the corresponding linear probe. At the other extreme, suppose that the quality scores are equal and that the imbalance term is negligible throughout selection. Then every κ_i equals a common value κ, so $\Sigma_S = \kappa \sum_{i \in S} h_i h_i^\top$ and the relevant set-dependent term is $-\frac{1}{2} \log \det(I_d + \Sigma_S)$. By the matrix determinant lemma, its greedy marginal ranks a candidate j by $h_j^\top (I_d + \Sigma_S)^{-1} h_j$, the residual-variance criterion used by RepExp on the same representations. Between these limits, the modular quality, log-determinant diversity, and imbalance terms jointly determine the selected set.

Between the two limits. These limits do not show how PKTP moves from Linear Probe to RepExp as k grows. To see this, we expand $\log L(S)$ and its surrogate when the correlations are weak.

Proposition 1 (Weak correlations). *Fix S, q_i and h_i, and replace $\mathbf{f}$ by $\rho\,\mathbf{f}$. As $\rho \to 0$,*

$$\log L(S) = \sum_{i \in S} \Big[\log \Phi(-q_i) + \tfrac{\rho^2}{2} \lambda_i q_i \|h_i\|^2 \Big] + \rho^2 \sum_{i < j \in S} \lambda_i \lambda_j \, h_i^\top h_j + O(\rho^4),$$

and $\log \widetilde{L}(S)$ has the same expansion.

The first sum expands the failure probability of each rollout on its own, and the second is the only interaction between rollouts at this order. It penalizes pairs of kept rollouts with similar representations, like maximal marginal relevance [5] but with weights $\lambda_i \lambda_j$ set by the latent model. Adding a rollout to a set of size t contributes its own failure term, which does not depend on the set, together with t pairwise penalties, each of order ρ^2 for bounded q_i. The penalties are therefore absent for the first pick and accumulate as the set grows, so quality decides at small k, while at larger k PKTP heuristically spreads its picks out, as RepExp does. See Appendix C for more discussion of this relation.

4 Experiments

4.1 Setup

Models. We evaluate on eight open-weight reasoning models in two size tiers. The small tier is Qwen3-0.6B, Qwen3-1.7B [49], OpenMath-Nemotron-1.5B [31] and DeepSeek-R1-Distill-Qwen-1.5B [15], and the large tier is Qwen3-8B, OpenMath-Nemotron-7B, DeepSeek-R1-Distill-Qwen-7B and Skywork-OR1-Math-7B [16]. Every model generates its rollouts with temperature 0.8, top-p of 0.95, and a full token budget of $B = 32{,}768$ tokens.

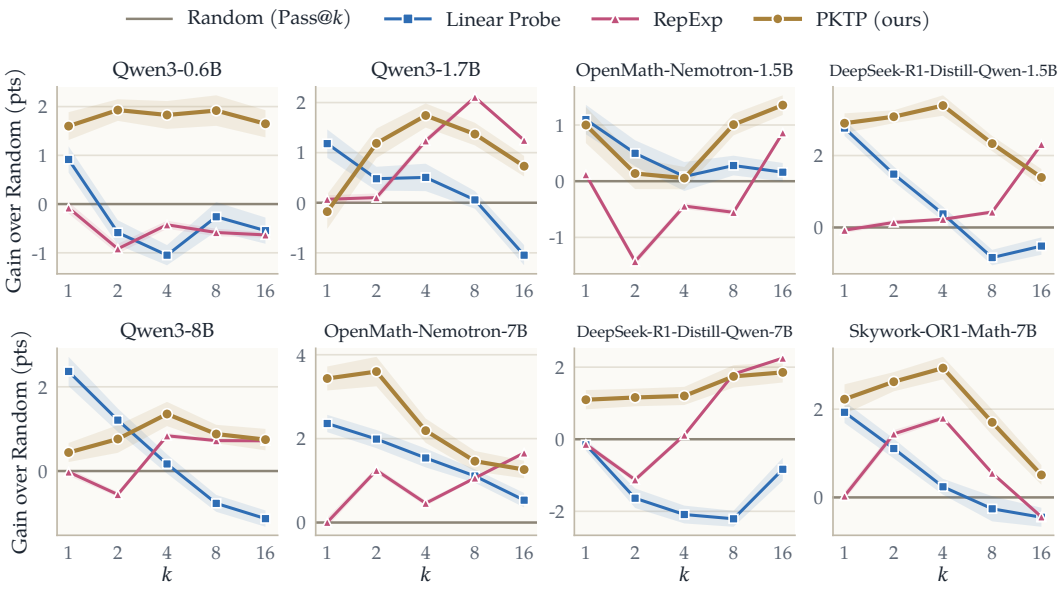


Figure 3: **Gain in** pass@k **over random pruning.** For every model, the pooled pass@k of each pruning policy minus that of Random (Pass@k), in percentage points, as a function of the number k of rollouts kept, with a band of one standard error of the repetition-by-repetition difference. The absolute numbers are in Table 1 and the per-benchmark breakdown in Appendix D.

Datasets. We pool problems from several benchmark sources rather than a single one, since a pruning policy has to be trained on part of the problems and evaluated on the rest, and no single competition benchmark is large enough for both. The two model tiers are evaluated on different benchmark substrates, chosen so that neither tier is saturated. Both tiers use AIME 2024, 2025 and 2026 and HMMT 2024, 2025 and 2026 [2], together with RIMO-N [9]. The small tier adds BeyondAIME [4], OlymMATH (Easy) [37] and Omni-MATH levels 7 and 8 [14], while the large tier replaces these by AMO-Bench [27], OlymMATH (Hard) and Omni-MATH levels 8 and 9. This gives around 600 problems per model. For each model we then remove the problems on which it generates all correct or all wrong solutions in 128 trials.

Verification. A completed rollout counts as correct when its final answer matches the reference answer of the problem. We take that answer from the `\boxed{}` expression of the rollout and ask a Qwen3-32B judge [49] whether it is equivalent to the reference answer. We use a judge rather than a symbolic checker such as `math-verify` [24], because benchmarks like AMO-Bench have descriptive answers that a symbolic checker cannot handle.

Evaluation. Only a few hundred problems are available for each model, so a single split into training and test problems would leave too few on either side, and we evaluate with cross-validation instead. For each model we partition its problems into eight folds, with each benchmark spread evenly across the folds. Each fold serves once as the test set while the learned components are trained on the other seven, so every problem is scored exactly once by a model that was not trained on it. One such pass over all eight folds forms a repetition. Because the outcome of a repetition depends on how the problems happen to be divided, we run 20 repetitions, each with a new random partition obtained by shuffling the problems of every benchmark before assigning them to folds, and with fresh random seeds for training. Every reported number is the average over the 20 repetitions, and its standard error is computed across them. For RepExp we set the regularizer to $\tau = 1$. For all methods, the representation h_i of a partial rollout is the EMA of its hidden states with rate 10^{-3}.

4.2 Experimental Evaluation and Comparison

Figure 3 reports our main setting, in which each method selects $k \in \{1, 2, 4, 8, 16\}$ of the $N = 128$ rollouts of a problem from their prefixes, and we set the cutoff threshold to $T = 4000$ tokens. For every model it shows the pass@k of each pruning policy minus that of Random (Pass@k), pooled

over all test problems. Random keeps k rollouts chosen uniformly at random, which is equivalent to sampling k rollouts without any pruning, so it is the natural reference point. The absolute numbers are given in Table 1 and the breakdown by benchmark in Appendix D.

As evident in Fig. 3, the Linear Probe performs strongly at $k = 1$, but as k increases it gets closer and closer to Random, as in the Qwen models, and in some cases it even becomes worse than Random at large k. This can be attributed to a bias towards a cluster of rollouts that the model predicts to be correct but that may not be, in the case where the classifier is wrong. In other words, ranking by the probe sharpens the distribution of the kept rollouts around what the probe considers correct. This mirrors the effect observed in reinforcement learning, where models trained with GRPO outperform their base models at small k but are outperformed by them at large k [53]. As an example, Fig. 4 shows the 128 rollouts of one problem (Qwen3-8B, AMO-Bench), projected on the first two principal components of their representations. The 16 rollouts ranked highest by the probe are all close to each other and all wrong, while 16 random picks include 7 correct ones.

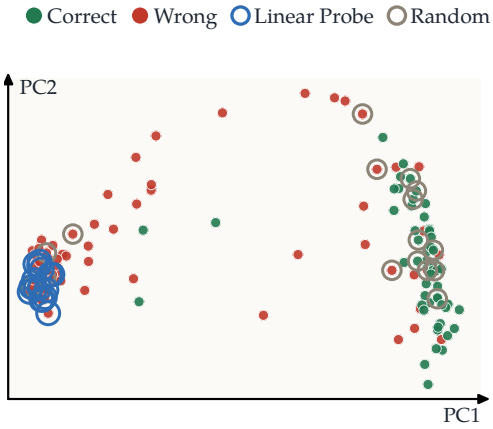


Figure 4: **Linear Probe versus random selection** of 16 rollouts on one problem.

Figure 3 also shows that RepExp behaves in the opposite way to the Linear Probe. At $k = 1$ it shows, as expected, a performance similar to Random, since its first pick is random. As k increases, the general trend is that RepExp gets better and better than Random, especially at $k = 16$ (e.g., the DeepSeek-R1-Distill models). Interestingly, however, for moderate k it is sometimes outperformed by Random, for instance on OpenMath-Nemotron-1.5B and Qwen3-8B. We attribute this to a tendency of RepExp to favor outliers at small and moderate k, since the rollouts farthest from those already selected are often unusual ones that are unlikely to be correct. A similar effect is reported by Tuyls et al. [39], where RepExp stops helping under high-temperature sampling, since the less coherent responses look novel in representation space yet do not necessarily contain correct answers. As an example, Fig. 5 shows the 128 rollouts of one problem (Qwen3-8B, Omni-MATH), embedded in two dimensions with multidimensional scaling on their representations. After its first, random pick, RepExp selects outliers far from the other rollouts, and all 4 of its picks are wrong, while 4 random picks include 1 correct one.

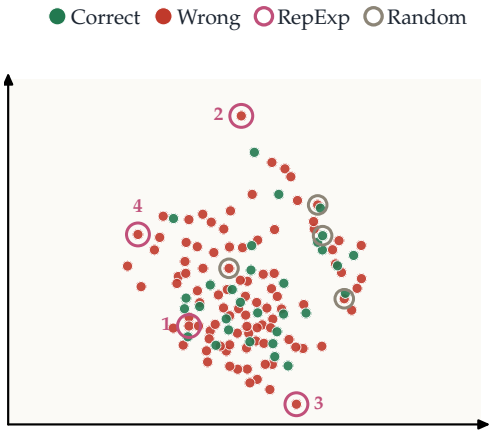


Figure 5: **RepExp versus random selection** of 4 rollouts on one problem.

Finally, Fig. 3 shows that PKTP outperforms both the Linear Probe and RepExp, and its advantage is clearest at moderate k, where each baseline suffers from its own bias. On DeepSeek-R1-Distill-Qwen-1.5B at $k = 4$, for example, PKTP is 3.0 points above the Linear Probe and 3.2 points above RepExp. On Qwen3-0.6B it leads both baselines by more than 2 points at every $k \geq 2$, and on DeepSeek-R1-Distill-Qwen-7B it is 3.3 points above the Linear Probe at $k = 4$. Unlike the two baselines, PKTP also stays above Random on every model for $k \geq 2$, since it keeps the correctness signal of the probe while spreading the kept rollouts over outcomes that are unlikely to fail together. At large k, however, the performance of PKTP and RepExp gets closer. We believe this is due to the weak correctness signal in the embeddings of the partial rollouts, which raises the question of how a quality embedding can be obtained from a partial rollout, and whether this is possible at all.

Results for the other pool sizes, $N \in \{32, 48, 64, 80, 96, 112\}$ with the same cutoff $T = 4000$, are given in Appendix E.

4.3 Effect of the cutoff threshold

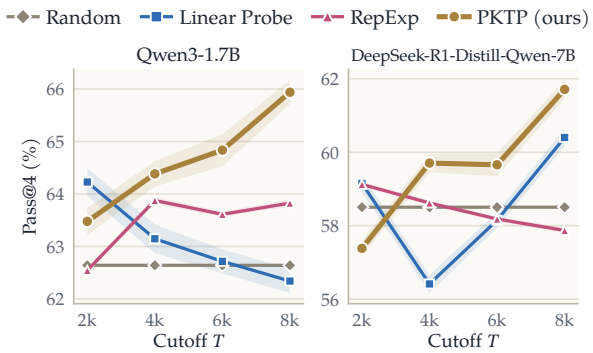


Figure 6: **Effect of the cutoff threshold.** Pass@4 as the cutoff T grows, with $N = 128$. Bands are one standard error.

An important aspect of pruning is the cutoff threshold T, and how the algorithm behaves as the cutoff grows. Figure 6 varies T on Qwen3-1.7B and DeepSeek-R1-Distill-Qwen-7B. For every $T \geq 4000$, PKTP outperforms all other methods on both models. An interesting observation is that a later cutoff in general improves the performance of PKTP. On Qwen3-1.7B its pass@4 rises from 63.5% at $T = 2000$ to 65.9% at $T = 8000$, and on DeepSeek-R1-Distill-Qwen-7B from 57.4% to 61.7%. Intuitively, the more tokens are generated, the more is revealed about the quality of a rollout and where it is heading, so there is more signal to exploit. Interestingly, however, the same does not necessarily happen for the Linear Probe and RepExp. On Qwen3-1.7B, for example, the Linear Probe drifts down from 64.2% to 62.3% and ends below Random. One reason can be the representation we use. Since the EMA with rate 10^{-3} effectively averages over the last 1000 tokens, it may lose the signal in the earlier tokens.

4.4 Effect of the pool size

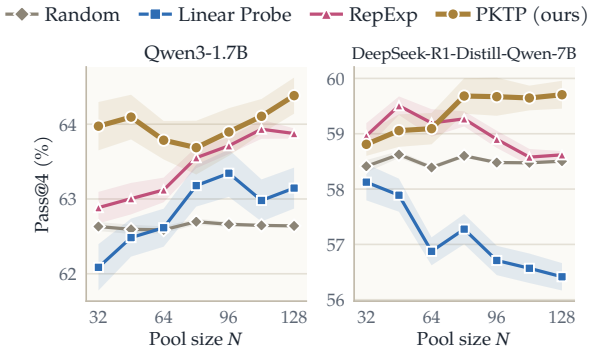


Figure 7: **Effect of the pool size.** Pass@4 as the pool size N grows, with $T = 4000$. Bands are one standard error.

Another important aspect of pruning is the size N of the pool the k rollouts are chosen from. Figure 7 varies N on Qwen3-1.7B and DeepSeek-R1-Distill-Qwen-7B. PKTP outperforms all other methods for every N on Qwen3-1.7B, and on DeepSeek-R1-Distill-Qwen-7B once the pool reaches $N = 80$. A larger pool offers more candidates to choose from, and RepExp benefits from it on Qwen3-1.7B, rising from 62.9% at $N = 32$ to 63.9% at $N = 128$ as the pool becomes more diverse. Interestingly, however, the same does not hold for the Linear Probe, which falls from 58.1% to 56.4% on DeepSeek-R1-Distill-Qwen-7B and stays below Random for every N. One reason can be that with a larger pool the probe is more likely to find a cluster of rollouts it is confidently wrong about, and it then keeps all of them.

5 Conclusion

We introduced PKTP, a pruning framework that chooses which partial rollouts to keep by directly optimizing pass@k. PKTP learns the joint distribution of the rollouts' outcomes with a latent Gaussian model, approximates the resulting likelihood in closed form, and selects the kept set greedily, with an approximation guarantee on the surrogate objective. Across eight reasoning models, it outperforms both quality-based and diversity-based pruning, and the gains are largest at moderate k, where each of these baselines suffers from its own bias. Since PKTP can be combined with any representation of partial rollouts, representations that better capture the correctness of a rollout would directly strengthen it, and developing them is a natural direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with several parts of the project. For the write-up, we used them to edit and polish the text and to format the LaTeX source. For the figures, we used them to write the scripts that plot our experimental results. For the code, we used them to help implement parts of our implementation. For the theory, we used them to help derive and write parts of the proofs in the appendix. We have not used generative AI tools to generate data or experimental results. All AI-assisted content, including the code and the proofs, was reviewed and verified by the authors, who take full responsibility for the content of this paper.

REFERENCES

[1] Anum Afzal, Florian Matthes, Gal Chechik, and Yftah Ziser. Knowing before saying: LLM representations encode information about chain-of-thought success before completion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12791–12806, 2025.

[2] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

[3] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

[4] ByteDance Seed. Seed1.5-Thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025.

[5] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[6] Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 429–436, 2006.

[7] Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Yannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. In *International Conference on Learning Representations*, 2026.

[8] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

[9] Ziye Chen, Chengwei Qin, and Yao Shu. RIMO: An easy-to-evaluate, hard-to-solve olympiad benchmark for advanced mathematical reasoning. *arXiv preprint arXiv:2509.07711*, 2025.

[10] Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. Inference-aware fine-tuning for best-of-N sampling in large language models. In *International Conference on Learning Representations*, 2025.

[11] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[12] Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 5th edition, 2019. doi: 10.1017/9781108591034.

[13] Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations*, 2026.

[14] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *International Conference on Learning Representations*, 2025.

[15] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[16] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

[17] Parsa Hosseini, Sumit Nawathe, Mahdi Salmani, Meisam Razaviyayn, and Soheil Feizi. Early stopping for large reasoning models via confidence dynamics. *arXiv preprint arXiv:2604.04930*, 2026.

[18] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[19] Xinyan Jiang, Ninghao Liu, Di Wang, and Lijie Hu. Beyond scalars: Evaluating and understanding LLM reasoning via geometric progress and stability. In *International Conference on Machine Learning*, 2026.

[20] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213, 2022.

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

[23] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[24] Hynek Kydlíček. Math-Verify: Math verification library. https://github.com/huggingface/Math-Verify, 2025.

[25] Zhixiang Liang, Beichen Huang, Zheng Wang, and Minjia Zhang. Hidden states as early signals: Step-level trace evaluation and pruning for efficient test-time scaling. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 26800–26813, 2026.

[26] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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*, 2024.

[27] Junlin Liu, Shengnan An, Shuang Zhou, Dan Ma, Yehao Lin, Xinxuan Lv, Xuanlin Wang, Xiaoyu Li, Ziwen Wang, Xuezhi Cao, and Xunliang Cai. AMO-Bench: Large language models still struggle in high school math competitions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 2120–2137, 2026.

[28] Xin Liu and Lu Wang. Answer convergence as a signal for early stopping in reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17896–17907, 2025.

[29] Haolang Lu, Minghui Pan, Ripeng Li, Guoshun Nan, Jialin Zhuang, Zijie Zhao, Zhongxiang Sun, Kun Wang, and Yang Liu. Streaming hallucination detection in long chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 21157–21183, 2026.

[30] Rohin Manvi, Anikait Singh, and Stefano Ermon. Adaptive inference-time compute: LLMs can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*, 2024.

[31] Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. AIMO-2 winning solution: Building state-of-the-art mathematical reasoning models with OpenMathReasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

[32] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[33] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624, 2023.

[34] Scott Sanner, Shengbo Guo, Thore Graepel, Sadegh Kharazmi, and Sarvnaz Karimi. Diverse retrieval via greedy optimization of expected 1-call@k in a latent subtopic relevance model. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 2011.

[35] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

[36] Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-N decoding via speculative rejection. In *Advances in Neural Information Processing Systems*, 2024.

[37] Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17438–17457, 2026.

[38] Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. LLM reasoning as trajectories: Step-specific representation geometry and correctness signals. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26872–26887, 2026.

[39] Jens Tuyls, Dylan J. Foster, Akshay Krishnamurthy, and Jordan T. Ash. Representation-based exploration for language models: From test-time to post-training. In *International Conference on Learning Representations*, 2026.

[40] Martina G. Vilas, Safoora Yousefi, Besmira Nushi, Eric Horvitz, and Vidhisha Balachandran. Tracing the traces: Latent temporal signals for efficient and accurate reasoning. In *International Conference on Learning Representations*, 2026.

[41] Christian Walder and Deep Tejas Karkhanis. Pass@K policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems*, volume 38, pages 152416–152445, 2025.

[42] Sophie L. Wang, Phillip Isola, and Brian Cheung. The truth lies somewhere in the middle (of the generated tokens). *arXiv preprint arXiv:2605.09969*, 2026.

[43] Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. Entropy after `</think>` for reasoning model early exiting. *arXiv preprint arXiv:2509.26522*, 2025.

[44] 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. In *International Conference on Learning Representations*, 2023.

[45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[46] James T. Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2018.

[47] Xueqing Wu, Langxing Bai, Hritik Bansal, Po-Nien Kung, Shuo Li, Hao Liu, Nanyun Peng, and Kai-Wei Chang. Planned test-time scaling with coordinated reasoning paths. *arXiv preprint arXiv:2609.27374*, 2026.

[48] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations*, 2025.

[49] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[50] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *International Conference on Learning Representations*, volume 2026, pages 88170–88210, 2026.

[51] Lijie Yang, Hongyin Luo, Jiawei Zhao, Tri Dao, and Ravi Netravali. Thought-level beam search for reasoning. *arXiv preprint arXiv:2608.08020*, 2026.

[52] 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. *Advances in Neural Information Processing Systems*, 36:11809–11822, 2023.

[53] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

[54] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Conference on Language Modeling*, 2025.

A Derivations

A.1 The likelihood

Write $\mathbf{Z}_i = q_i + \mathbf{m}_i + h_i^\top \mathbf{f}$. Conditioned on $\mathbf{f}$, the $\mathbf{Z}_i$ are independent of each other, so

$$\begin{aligned}\Pr\left[\mathbf{C}_1 = c_1, \ldots, \mathbf{C}_N = c_N \mid \mathbf{f}\right] &= \prod_{i=1}^{N} \Pr[c_i \mathbf{Z}_i \geq 0 \mid \mathbf{f}] \\ &= \prod_{i=1}^{N} \Pr[c_i(q_i + \mathbf{m}_i + h_i^\top \mathbf{f}) \geq 0 \mid \mathbf{f}] \\ &= \prod_{i=1}^{N} \Pr[c_i \mathbf{m}_i \geq -c_i(q_i + h_i^\top \mathbf{f}) \mid \mathbf{f}].\end{aligned}$$

Since the standard Gaussian is symmetric about 0, $-c_i\mathbf{m}_i$ follows the same distribution as $\mathbf{m}_i$, and with $\Phi(x) := \Pr_{Z\sim\mathcal{N}(0,1)}[Z \leq x]$ each factor equals $\Phi\big(c_i(q_i + h_i^\top \mathbf{f})\big)$. Taking the expectation with respect to $\mathbf{f}$ on both sides gives Eq. 4. Setting $c_i = -1$ for every $i \in S$ and marginalizing the remaining coordinates gives Eq. 7 in the same way.

A.2 Rank-one greedy updates

Write $\Omega_S = (I_d + \Sigma_S)^{-1}$, $r_j = h_j^\top \Omega_S h_j$ and $u_j = h_j^\top \Omega_S b_S$. Adding a candidate $j \notin S$ changes Σ_S by the rank-one matrix $\kappa_j h_j h_j^\top$. The matrix determinant lemma gives

$$\det(I_d + \Sigma_S + \kappa_j h_j h_j^\top) = \det(I_d + \Sigma_S)\big(1 + \kappa_j r_j\big), \tag{9}$$

while Sherman–Morrison gives

$$(I_d + \Sigma_S + \kappa_j h_j h_j^\top)^{-1} = \Omega_S - \frac{\kappa_j \Omega_S h_j h_j^\top \Omega_S}{1 + \kappa_j r_j}. \tag{10}$$

Substituting Eq. 9 and Eq. 10 into Eq. 8, and using $b_{S\cup\{j\}} = b_S + \lambda_j h_j$, the corresponding change in the unhalved quadratic form is

$$2\lambda_j u_j + \lambda_j^2 r_j - \frac{\kappa_j (u_j + \lambda_j r_j)^2}{1 + \kappa_j r_j} = \frac{2\lambda_j u_j + \lambda_j^2 r_j - \kappa_j u_j^2}{1 + \kappa_j r_j}.$$

Half of this, together with the change $-\frac{1}{2}\log(1 + \kappa_j r_j)$ in the determinant term and the new quality term $\log \Phi(-q_j)$, yields

$$\Delta_j(S) = \log \widetilde{L}(S\cup\{j\}) - \log \widetilde{L}(S) = \log \Phi(-q_j) - \frac{1}{2}\log(1+\kappa_j r_j) + \frac{2\lambda_j u_j + \lambda_j^2 r_j - \kappa_j u_j^2}{2(1 + \kappa_j r_j)}. \tag{11}$$

The greedy step adds the candidate with the smallest $\Delta_j(S)$. After selecting j, the maintained state updates are

$$\Omega_{S\cup\{j\}} = \Omega_S - \frac{\kappa_j \Omega_S h_j h_j^\top \Omega_S}{1 + \kappa_j r_j}, \qquad b_{S\cup\{j\}} = b_S + \lambda_j h_j.$$

The remaining scalars r_ℓ and u_ℓ can be updated recursively as well. If $g_\ell = h_\ell^\top \Omega_S h_j$, then

$$r_\ell \leftarrow r_\ell - \frac{\kappa_j g_\ell^2}{1 + \kappa_j r_j}, \qquad u_\ell \leftarrow u_\ell + \frac{\lambda_j - \kappa_j u_j}{1 + \kappa_j r_j}\, g_\ell.$$

With all h_i retained in memory, each greedy step costs $O(Nd + d^2)$, so one forward pass costs $O\big(k(Nd + d^2)\big)$ time and $O(Nd + d^2)$ storage.

B PROOFS

B.1 PROOF OF THEOREM 1

Here $\Pr[\mathbf{C}_S = c \mid H] = \mathbb{E}_{\mathbf{f}}\big[\prod_{i\in S} \Phi\big(c_i(q_i + h_i^\top \mathbf{f})\big)\big]$, as in Eq. 4 restricted to S. Fix a set $S \subseteq [N]$ and outcomes $c \in \{-1, 1\}^S$, and write λ_i, κ_i, Σ_S and b_S for $\lambda_i(c_i)$, $\kappa_i(c_i)$, $\Sigma_S(c)$ and $b_S(c)$. For each $i \in S$ and a scalar perturbation x, let $\ell_i(x) := \log \Phi\big(c_i(q_i + x)\big)$, and let $r(u) := \phi(u)/\Phi(u)$, which satisfies $r'(u) = -r(u)\big(u + r(u)\big)$. Since $c_i^2 = 1$ and ϕ is even, $r(c_i q_i) = \lambda_i$, and

$$\ell_i'(x) = c_i\, r\big(c_i(q_i + x)\big) \implies \ell_i'(0) = c_i\lambda_i, \qquad \ell_i''(0) = r'(c_i q_i) = -\lambda_i(\lambda_i + c_i q_i) = -\kappa_i.$$

To see that $\kappa_i > 0$, let $u = c_i q_i$. For $u \geq 0$ this follows immediately from $\lambda_i > 0$. For $u < 0$,

$$\Phi(u) = \int_{|u|}^{\infty} \phi(t)\, dt < \frac{1}{|u|} \int_{|u|}^{\infty} t\phi(t)\, dt = \frac{\phi(u)}{|u|},$$

and hence $\lambda_i > |u| = -c_i q_i$. Taylor expansion around $x = 0$ now gives

$$\log \Phi\big(c_i(q_i + x)\big) = \log \Phi(c_i q_i) + c_i\lambda_i x - \frac{\kappa_i}{2}x^2 + R_i(x), \qquad R_i(x) = O(|x|^3). \tag{12}$$

We now substitute $x = h_i^\top \mathbf{f}$ and sum Eq. 12 over $i \in S$. Since $\mathbf{f}$ is common for all $i \in S$, the only aggregate quantities that arise are Σ_S and b_S, and the term inside the expectation can be written as

$$\prod_{i\in S} \Phi\big(c_i(q_i + h_i^\top \mathbf{f})\big) = \left(\prod_{i\in S} \Phi(c_i q_i)\right) \exp\left(b_S^\top \mathbf{f} - \frac{1}{2}\mathbf{f}^\top \Sigma_S \mathbf{f} + \sum_{i\in S} R_i(h_i^\top \mathbf{f})\right).$$

Dropping the accumulated remainder gives the quadratic approximation

$$\widetilde{\Pr}[\mathbf{C}_S = c \mid H] := \left(\prod_{i\in S} \Phi(c_i q_i)\right) \mathbb{E}_{\mathbf{f}} \exp\left(b_S^\top \mathbf{f} - \frac{1}{2}\mathbf{f}^\top \Sigma_S \mathbf{f}\right). \tag{13}$$

The expectation in Eq. 13 has a closed form. Combining its exponent with the standard Gaussian density gives

$$\mathbb{E}_{\mathbf{f}} \exp\left(b_S^\top \mathbf{f} - \frac{1}{2}\mathbf{f}^\top \Sigma_S \mathbf{f}\right) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2} f^\top (I_d + \Sigma_S) f + b_S^\top f\right) df.$$

Because $\kappa_i > 0$, $\Sigma_S \succeq 0$ and the matrix $I_d + \Sigma_S$ is positive definite. Completing the square,

$$\begin{aligned} -\frac{1}{2} f^\top (I_d + \Sigma_S) f + b_S^\top f = & -\frac{1}{2}\big(f - (I_d + \Sigma_S)^{-1} b_S\big)^\top (I_d + \Sigma_S)\big(f - (I_d + \Sigma_S)^{-1} b_S\big) \\ & + \frac{1}{2} b_S^\top (I_d + \Sigma_S)^{-1} b_S. \end{aligned}$$

After translating the integration variable, the remaining Gaussian integral contributes $\det(I_d + \Sigma_S)^{-1/2}$, which gives Eq. 6. □

Proof of Corollary 1. Take $c_i = -1$ for every $i \in S$. Then $\lambda_i(-1) = \phi(q_i)/\Phi(-q_i)$, $\kappa_i(-1) = \lambda_i(\lambda_i - q_i)$, and $b_S(c) = -\sum_{i\in S} \lambda_i h_i$, and throughout the appendix we write $b_S = \sum_{i\in S} \lambda_i h_i$. Since b_S enters Eq. 6 only through $b_S^\top (I_d + \Sigma_S)^{-1} b_S$, the surrogate of $L(S)$ is $\widetilde{L}(S)$, and since the logarithm is increasing, minimizing $\widetilde{L}(S)$ is equivalent to minimizing $\log \widetilde{L}(S) = F(S)$. □

B.2 SUBMODULARITY OF THE EXACT AND APPROXIMATE pass@k OBJECTIVE

The submodularity of the exact pass probability is a standard property of the probability of a union of events [46], and we prove it here for completeness. The new result is that the surrogate keeps this property.

Definition 1 (Submodular Functions)**.** *A set function $G : 2^{[N]} \to \mathbb{R}$ is submodular if, for every $A \subseteq B \subseteq [N]$ and every $j \notin B$,*

$$G(A \cup \{j\}) - G(A) \geq G(B \cup \{j\}) - G(B). \tag{14}$$

Such a function is also called submodular. It is monotone if $G(A) \leq G(B)$ whenever $A \subseteq B$, and normalized if $G(\varnothing) = 0$.

Submodularity of Exact $\mathrm{pass@}k$ Probability First, we show that the exact $\mathrm{pass@}k$ probability is already a normalized monotone submodular set function. To see this, let $\mathcal{C}_i$ denote the event that response i is correct. The exact pass probability can be written as

$$P(S) := \Pr\left(\bigcup_{i\in S} \mathcal{C}_i\right).$$

For any $A \subseteq B \subseteq [N]$ and $j \notin B$, the marginal improvement from adding response j is

$$P(A \cup \{j\}) - P(A) = \Pr\left(\mathcal{C}_j \setminus \bigcup_{i\in A} \mathcal{C}_i\right), \tag{15}$$

$$P(B \cup \{j\}) - P(B) = \Pr\left(\mathcal{C}_j \setminus \bigcup_{i\in B} \mathcal{C}_i\right). \tag{16}$$

Since $A \subseteq B$,

$$\mathcal{C}_j \setminus \bigcup_{i\in B} \mathcal{C}_i \subseteq \mathcal{C}_j \setminus \bigcup_{i\in A} \mathcal{C}_i.$$

Taking probabilities therefore gives

$$P(A \cup \{j\}) - P(A) \geq P(B \cup \{j\}) - P(B),$$

which is precisely the submodular function property. Moreover, $P(\varnothing) = 0$, and $P(A) \leq P(B)$ whenever $A \subseteq B$. Hence the exact pass probability is normalized, monotone, and submodular. This conclusion holds for an arbitrary joint distribution of the correctness events and does not require any latent-variable model or conditional independence.

The same fact can be seen directly from the Gaussian modelling of scores. Define $p_i(\mathbf{f}) := \Phi\big(-(q_i + h_i^\top \mathbf{f})\big) \in (0,1)$. Then for the exact objective, we get that

$$L(S) = \mathbb{E}_{\mathbf{f}} \prod_{i\in S} p_i(\mathbf{f}), \qquad P(S) = 1 - L(S). \tag{17}$$

Consequently,

$$\begin{aligned} P(A \cup \{j\}) - P(A) &= \mathbb{E}_{\mathbf{f}}\left[\big(1 - p_j(\mathbf{f})\big) \prod_{i\in A} p_i(\mathbf{f})\right] \\ &\geq \mathbb{E}_{\mathbf{f}}\left[\big(1 - p_j(\mathbf{f})\big) \prod_{i\in B} p_i(\mathbf{f})\right] = P(B \cup \{j\}) - P(B), \end{aligned} \tag{18}$$

where the inequality follows pointwise from $0 < p_i(\mathbf{f}) < 1$ and $A \subseteq B$. Equivalently, the exact failure probability L is monotone decreasing and supermodular; it is the pass probability $P = 1 - L$ that is submodular.

Submodularity of the Approximate Pass Probability Now, we show that the approximation under the Gaussian latent model itself is a monotone submodular function. Note that, every approximation of a submodular function may not necessarily be submodular. Before, we proceed, let's try to get an intuition on the fact that $\widetilde{L}(S)$ indeed represents some notion of a probability. Conditional on $\mathbf{f}$, each term of the product in $\widetilde{L}(S)$ can be written as

$$g_i(\mathbf{f}) := \Phi(-q_i) \exp\left(-\lambda_i h_i^\top \mathbf{f} - \frac{\kappa_i}{2}(h_i^\top \mathbf{f})^2\right).$$

Consequently,

$$\widetilde{L}(S) = \mathbb{E}_{\mathbf{f}\sim\mathcal{N}(0,I_d)} \prod_{i\in S} g_i(\mathbf{f}). \tag{19}$$

Hence, for $\widetilde{L}(S)$ to be a definition of some kind of failure probability of selection of correct elements conditioned on $\mathbf{f}$, it would suffice to prove that $g_i(\mathbf{f}) < 1$. Towards this, for a scalar q, write

$$\lambda(q) := \frac{\phi(q)}{\Phi(-q)}, \qquad \kappa(q) := \lambda(q)(\lambda(q) - q),$$

and define

$$\beta(q) := \Phi(-q)\exp\left(\frac{\lambda(q)^2}{2\kappa(q)}\right) = \Phi(-q)\exp\left(\frac{\lambda(q)}{2(\lambda(q)-q)}\right).$$

Lemma 1. *For every finite $q \in \mathbb{R}$, one has $0 < \beta(q) < 1$. Moreover, $\beta(q)$ is strictly decreasing in q.*

Proof. Let $\overline{\Phi}(q) := \Phi(-q) = 1 - \Phi(q)$ and put

$$v(q) := \lambda(q) - q > 0, \qquad r(q) := \log \beta(q) = \log \overline{\Phi}(q) + \frac{\lambda(q)}{2v(q)}.$$

Since $\overline{\Phi}'(q) = -\phi(q)$, we have

$$\frac{d}{dq}\log \overline{\Phi}(q) = -\lambda(q).$$

Also, direct differentiation gives

$$\lambda'(q) = \lambda(q)(\lambda(q) - q) = \lambda(q)v(q), \qquad v'(q) = \lambda(q)v(q) - 1.$$

Consequently,

$$r'(q) = -\lambda(q) + \frac{\lambda'(q)v(q) - \lambda(q)v'(q)}{2v(q)^2} = \frac{\lambda(q)}{2v(q)^2}\left(1 - qv(q) - 2v(q)^2\right). \tag{20}$$

It remains to show that the expression in parentheses is negative. For this, let $X \sim N(0,1)$. Using the fact that, $\mathbb{E}[X \mid X > q] = \lambda(q)$, and $\mathbb{E}[X^2 \mid X > q] = 1 + q\lambda(q)$, we get

$$\mathbb{E}\left[X - q \,\middle|\, X > q\right] = v(q), \qquad \mathbb{E}\left[(X-q)^2 \,\middle|\, X > q\right] = 1 - qv(q). \tag{21}$$

Hence, it would suffice to prove that

$$\frac{1}{2}\mathbb{E}\left[(X-q)^2 \,\middle|\, X > q\right] \le \left(\mathbb{E}\left[X - q \,\middle|\, X > q\right]\right)^2.$$

To prove the above expression, consider

$$T(x) := \Pr(X - q > x | X > q) = \frac{\overline{\Phi}(q+x)}{\overline{\Phi}(q)}, \qquad x \ge 0.$$

Under the conditional law given $X > q$, the random variable $X - q$ is nonnegative. Hence, by the tail-integral formula for second moments [12, Lemma 2.2.13],

$$\begin{aligned}\frac{1}{2}\mathbb{E}\big[(X-q)^2 \mid X > q\big] &= \mathbb{E}\left[\int_0^\infty t\,\mathbf{1}_{\{X-q>t\}}\,dt \,\middle|\, X > q\right] \\ &= \int_0^\infty t\,\Pr(X - q > t \mid X > q)\,dt = \int_0^\infty tT(t)\,dt.\end{aligned}$$

The interchange of expectation and integration is justified by Tonelli's theorem [12, Theorem 1.7.2], since the integrand is nonnegative. Moreover, substituting $t = x + y$ in the inner integral and then exchanging the order of integration by Tonelli's theorem again, we obtain

$$\int_0^\infty \int_0^\infty T(x+y)\,dx\,dy = \int_0^\infty \int_y^\infty T(t)\,dt\,dy = \int_0^\infty \int_0^t T(t)\,dy\,dt = \int_0^\infty tT(t)\,dt.$$

Here the integration region is $\{(y,t) : 0 \le y \le t < \infty\}$, and nonnegativity of T justifies the exchange. We know that λ is strictly increasing because $\lambda'(q) = \kappa(q) > 0$. Hence, for $x, y > 0$,

$$\frac{T(x+y)}{T(x)} = \exp\left(-\int_x^{x+y}\lambda(q+t)\,dt\right) < \exp\left(-\int_0^y \lambda(q+t)\,dt\right) = T(y),$$

and therefore $T(x+y) < T(x)T(y)$. Thus, we get that

$$\begin{aligned}\frac{1}{2}\mathbb{E}\left[(X-q)^2 \,\middle|\, X > q\right] &= \int_0^\infty tT(t)\,dt = \int_0^\infty \int_0^\infty T(x+y)\,dx\,dy \\ &< \left(\int_0^\infty T(x)\,dx\right)^2 = \left(\mathbb{E}\left[X - q \,\middle|\, X > q\right]\right)^2.\end{aligned} \tag{22}$$

Combining Eq. 21 and Eq. 22 yields

$$1 - qv(q) - 2v(q)^2 < 0.$$

By Eq. 20, $r'(q) < 0$. Finally, as $q \to -\infty$, $\overline{\Phi}(q) \to 1$, $\lambda(q) \to 0$, and $\lambda(q)/(2v(q)) \to 0$. Thus $r(q) \to 0$. Since r is strictly decreasing, $r(q) < 0$ for every finite q. Equivalently, $0 < \beta(q) < 1$, and β is strictly decreasing. □

Thus, coming back to $g_i(\mathbf{f})$, completing the square in the scalar $x = h_i^\top \mathbf{f}$ gives

$$g_i(\mathbf{f}) = \Phi(-q_i) \exp\left(-\frac{\kappa_i}{2} \left(x + \frac{\lambda_i}{\kappa_i} \right)^2 + \frac{\lambda_i^2}{2\kappa_i} \right) \leq \beta(q_i) < 1$$

by Lemma 1. Thus $0 < g_i(\mathbf{f}) < 1$ for every i and every $\mathbf{f}$. Hence, one can define the approximate pass@k probability by

$$\widetilde{P}(S) := 1 - \widetilde{L}(S). \tag{23}$$

Theorem 3. *$\widetilde{P}$ is normalized, monotone, and submodular.*

Proof. It follows immediately from Eq. 19 that $\widetilde{L}$ decreases under set inclusion, and hence $\widetilde{P} = 1 - \widetilde{L}$ is monotone. Also, $\widetilde{P}(\varnothing) = 1 - 1 = 0$, so $\widetilde{P}$ is normalized.

The following part follows exactly in the same manner as the proof of submodularity of the latent-Gaussian pass@k objective. Now let $A \subseteq B$ and $j \notin B$. The marginal improvement obtained by adding j is

$$\widetilde{P}(A \cup \{j\}) - \widetilde{P}(A) = \widetilde{L}(A) - \widetilde{L}(A \cup \{j\}) = \mathbb{E}_{\mathbf{f}} \left[(1 - g_j(\mathbf{f})) \prod_{i \in A} g_i(\mathbf{f}) \right].$$

Because every $g_i(\mathbf{f})$ lies in $(0, 1)$,

$$(1 - g_j(\mathbf{f})) \prod_{i \in A} g_i(\mathbf{f}) \geq (1 - g_j(\mathbf{f})) \prod_{i \in B} g_i(\mathbf{f})$$

pointwise in $\mathbf{f}$. Taking expectations proves that $\widetilde{P}$ is submodular. □

Theorem 4. *For an integer $1 \leq k \leq N$, let S_k^{greedy} be the set obtained after k steps, starting from the empty set and, at each step, adding a candidate $j \notin S$ with the smallest marginal change*

$$F(S \cup \{j\}) - F(S).$$

If $S_k^\star$ maximizes $\widetilde{P}(S)$ among all sets of size at most k, then

$$\widetilde{P}(S_k^{\text{greedy}}) \geq (1 - e^{-1}) \widetilde{P}(S_k^\star). \tag{24}$$

Proof. For a fixed current set S,

$$\widetilde{P}(S \cup \{j\}) - \widetilde{P}(S) = \widetilde{L}(S) - \widetilde{L}(S \cup \{j\}) = \widetilde{L}(S) \left(1 - \exp\big(F(S \cup \{j\}) - F(S)\big) \right).$$

Since $\widetilde{L}(S)$ is fixed while j varies and the exponential is increasing, maximizing the marginal gain in $\widetilde{P}$ is exactly equivalent to minimizing the marginal change in F. Thus the greedy update derived in the preceding derivation is the standard greedy algorithm for $\widetilde{P}$. Since $\widetilde{P}$ is a monotone submodular function, by Nemhauser et al. [32], the utility guarantee of the greedy solution holds. □

B.3 Proof of Proposition 1

This section proves Proposition 1, which describes the failure probability and its surrogate when the correlations between rollouts are weak. To weaken the correlations while keeping the quality scores fixed, we replace $\mathbf{f}$ by $\rho\,\mathbf{f}$ for $\rho > 0$, so that

$$\mathbf{Z}_i = q_i + \rho\, h_i^\top \mathbf{f} + \mathbf{m}_i, \qquad \mathrm{Cov}(\mathbf{Z}_i, \mathbf{Z}_j \mid H) = \rho^2 h_i^\top h_j \quad (i \neq j),$$

and $\rho = 1$ recovers the model of Section 3.1. The quality scores $q_i = Q_\theta(h_i)$ and the representations h_i are held fixed. Since only the products $\rho\, h_i$ enter the model, this is equivalent to scaling the representations by ρ with the q_i fixed, as in assumption (A2) of Appendix C. The failure probability of a set S becomes

$$L(S) = \mathbb{E}_{\mathbf{f}} \prod_{i \in S} \Phi\big(-q_i - \rho\, h_i^\top \mathbf{f}\big),$$

and its surrogate is Eq. 8 with every h_i replaced by $\rho\, h_i$.

Proof of Proposition 1. Let $\varepsilon_i = \rho\, h_i^\top \mathbf{f}$ for $i \in S$, so that $\varepsilon_S \sim \mathcal{N}(0, K)$ with $K_{ij} = \rho^2 h_i^\top h_j$, and let $u_i(x) = \log \Phi(-q_i - x)$. Then $u_i'(0) = -\lambda_i$ and $u_i''(0) = -\kappa_i$, and every derivative of $\log \Phi$ of order two or higher is bounded on $\mathbb{R}$. By Taylor's theorem, for all $x \in \mathbb{R}$,

$$u_i(x) - u_i(0) = -\lambda_i x - \frac{1}{2}\kappa_i x^2 + \frac{1}{6} u_i'''(0)\, x^3 + r_i(x), \qquad |r_i(x)| \le C x^4,$$

where C does not depend on ρ. Let $X = \sum_{i \in S} \big(u_i(\varepsilon_i) - u_i(0)\big)$, so that $L(S) = \prod_{i \in S} \Phi(-q_i)\, \mathbb{E}\, e^X$. Since $\Phi \le 1$, we have $e^X \le M_S := \prod_{i \in S} \Phi(-q_i)^{-1}$, and Taylor's theorem for the exponential gives $\big|e^X - 1 - X - \frac{1}{2}X^2 - \frac{1}{6}X^3\big| \le \frac{M_S}{24} X^4$. Moreover, $|X| \le C'\big(\|\varepsilon_S\| + \|\varepsilon_S\|^2\big)$ for a constant C', so $\mathbb{E}X^4 = O(\rho^4)$ and

$$\mathbb{E}\, e^X = 1 + \mathbb{E}X + \frac{1}{2}\mathbb{E}X^2 + \frac{1}{6}\mathbb{E}X^3 + O(\rho^4).$$

The odd moments of ε_S vanish, and its moments of order four are $O(\rho^4)$. Hence

$$\mathbb{E}X = -\frac{1}{2}\sum_{i \in S} \kappa_i K_{ii} + O(\rho^4), \qquad \mathbb{E}X^2 = \sum_{i,j \in S} \lambda_i \lambda_j K_{ij} + O(\rho^4), \qquad \mathbb{E}X^3 = O(\rho^4).$$

With $a := -\frac{1}{2}\sum_{i \in S} \kappa_i K_{ii} + \frac{1}{2}\sum_{i,j \in S} \lambda_i \lambda_j K_{ij} = O(\rho^2)$, this gives $\mathbb{E}\, e^X = 1 + a + O(\rho^4)$, and therefore $\log \mathbb{E}\, e^X = a + O(\rho^4)$. Splitting the double sum into its diagonal and off-diagonal parts and using $\lambda_i^2 - \kappa_i = \lambda_i q_i$ yields

$$a = \frac{\rho^2}{2} \sum_{i \in S} \lambda_i q_i \|h_i\|^2 + \rho^2 \sum_{i < j \in S} \lambda_i \lambda_j\, h_i^\top h_j,$$

which proves the expansion of $\log L(S)$.

For the surrogate, Eq. 8 with $\rho\, h_i$ in place of h_i reads

$$\log \widetilde{L}(S) = \sum_{i \in S} \log \Phi(-q_i) - \frac{1}{2} \log\det(I_d + \rho^2 \Sigma_S) + \frac{\rho^2}{2}\, b_S^\top (I_d + \rho^2 \Sigma_S)^{-1} b_S.$$

Since $\log\det(I_d + \rho^2 \Sigma_S) = \rho^2\, \mathrm{tr}\, \Sigma_S + O(\rho^4)$ and $(I_d + \rho^2 \Sigma_S)^{-1} = I_d + O(\rho^2)$,

$$\log \widetilde{L}(S) = \sum_{i \in S} \log \Phi(-q_i) - \frac{\rho^2}{2} \sum_{i \in S} \kappa_i \|h_i\|^2 + \frac{\rho^2}{2} \Big\| \sum_{i \in S} \lambda_i h_i \Big\|^2 + O(\rho^4),$$

and expanding the square gives the same expansion as for $\log L(S)$. □

B.4 HARDNESS OF EXACT SELECTION

Proposition 2. *Minimizing* $\log \widetilde{L}(S)$ *over all* $S \subseteq [N]$ *with* $|S| = k$ *is NP-hard, even when all* q_i *equal the same integer and all* h_i *have entries in* $\{0, 1\}$.

Proof. We reduce from the independent set problem, which asks whether a graph contains k pairwise non-adjacent vertices and is NP-complete [20]. Let G be a graph on the vertex set $[N]$ with edge set E, adjacency matrix A and maximum degree $\Delta \geq 1$ (the case without edges is trivial). For $S \subseteq [N]$, let A_S be the adjacency matrix of the subgraph induced by S and $e(S)$ its number of edges. We construct an instance whose minimizers are independent sets of size k whenever such a set exists.

Let $d = |E| + N\Delta$. The first $|E|$ coordinates of $\mathbb{R}^d$ are indexed by the edges of G, and the remaining ones are split into N blocks of Δ coordinates, one block per vertex. For each vertex i, let $h_i \in \{0, 1\}^d$ have a one in the coordinate of every edge incident to i and a one in $\Delta - \deg(i)$ coordinates of its own block. Two distinct vertices share a nonzero coordinate only if they are adjacent, and then they share exactly one, so $\|h_i\|_2^2 = \Delta$ and $h_i^\top h_j = A_{ij}$ for $i \neq j$, and the Gram matrix of $\{h_i\}_{i \in S}$ is $\Delta I_k + A_S$. We further set $q_i = q$ for all i, for an integer $q \leq -1$ fixed below, so that $\lambda_i = \lambda$ and $\kappa_i = \kappa$. Then $\Sigma_S = \kappa \sum_{i \in S} h_i h_i^\top$ and $b_S = \lambda \sum_{i \in S} h_i$, and Eq. 8 reads

$$\log \widetilde{L}(S) = k \log \Phi(-q) - \frac{1}{2} D(S) + R(S),$$

where $D(S) = \log\det(I_d + \Sigma_S)$ and $R(S) = \frac{1}{2} b_S^\top (I_d + \Sigma_S)^{-1} b_S \geq 0$. Since $q \leq -1$, we have $\Phi(-q) \geq \frac{1}{2}$, hence $\lambda \leq 2\phi(q) < \frac{1}{2}$ and $\lambda|q| \leq 2|q|\phi(q) \leq 2\phi(1) < \frac{1}{2}$, where we used that $t\phi(t)$ is decreasing for $t \geq 1$. Consequently,

$$\lambda|q| \leq \kappa = \lambda^2 + \lambda|q| < 1. \tag{25}$$

We first show that D is maximized exactly by the independent sets, with an explicit margin. By Sylvester's determinant identity and the form of the Gram matrix,

$$D(S) = \log\det\big((1 + \kappa\Delta) I_k + \kappa A_S\big) = k \log(1 + \kappa\Delta) + \sum_{j=1}^{k} \log(1 + x_j), \qquad x_j = \frac{\kappa \mu_j}{1 + \kappa\Delta},$$

where $\mu_1, \ldots, \mu_k$ are the eigenvalues of A_S. They satisfy $\sum_j \mu_j = \operatorname{tr} A_S = 0$, $\sum_j \mu_j^2 = \operatorname{tr} A_S^2 = 2e(S)$ and $|\mu_j| \leq \Delta$, hence $\sum_j x_j = 0$ and $|x_j| < 1$. If S is independent, then $A_S = 0$ and $D(S) = k \log(1 + \kappa\Delta)$. Otherwise $e(S) \geq 1$, and the inequality $\log(1 + x) \leq x - x^2/8$ for $|x| < 1$, together with Eq. 25, gives

$$D(S) \leq k \log(1 + \kappa\Delta) - \frac{1}{8} \sum_{j=1}^{k} x_j^2 = k \log(1 + \kappa\Delta) - \frac{\kappa^2 e(S)}{4(1 + \kappa\Delta)^2} \leq k \log(1 + \kappa\Delta) - \frac{\kappa^2}{4(1 + \Delta)^2}.$$

We next bound the imbalance term uniformly over S. Since $(I_d + \Sigma_S)^{-1} \preceq I_d$ and $\big\|\sum_{i \in S} h_i\big\|_2^2 = k\Delta + 2e(S) \leq k(\Delta + k)$, we have $0 \leq R(S) \leq \frac{1}{2}\|b_S\|_2^2 \leq \frac{1}{2}\lambda^2 k(\Delta + k)$.

Now suppose that G contains an independent set S° of size k, and let S be a set of size k with $e(S) \geq 1$. Combining the two bounds with Eq. 25,

$$\begin{aligned} \log \widetilde{L}(S) - \log \widetilde{L}(S^\circ) &\geq \frac{1}{2}\big(D(S^\circ) - D(S)\big) - R(S^\circ) \geq \frac{\kappa^2}{8(1 + \Delta)^2} - \frac{\lambda^2}{2} k(\Delta + k) \\ &\geq \lambda^2 \Big(\frac{q^2}{8(1 + \Delta)^2} - \frac{k(\Delta + k)}{2} \Big), \end{aligned}$$

and the right-hand side is positive for $q = -2(1 + \Delta)(\Delta + k)$. With this choice, every minimizer of $\log \widetilde{L}$ is an independent set whenever G contains one of size k, so G contains an independent set of size k if and only if a minimizer of $\log \widetilde{L}$ is independent. The instance consists of N binary vectors of dimension $d = |E| + N\Delta$ and a single integer q of polynomial magnitude, so it is constructed in polynomial time, which proves the claim. □

C SELECTION BEHAVIOR ACROSS BUDGETS

The preceding section establishes submodularity and a guarantee for the greedy rule under the approximate pass objective. Hereafter, the *surrogate greedy rule* starts from the empty set and successively adds a candidate minimizing $F(S \cup \{j\}) - F(S)$, whereas the *probe-style rule* selects the k candidates with the largest q_i. We now study when the surrogate greedy rule agrees with the probe-style rule and how the expected log-surrogate relates to the determinant criterion used by RepExp. At small budgets, a sufficiently large quality gap makes the two rules select the same top-k set; imposing the corresponding condition at every cutoff also makes their selection orders agree. At large budgets, minimizers of $\mathbb{E}F(S)$ over the admissible comparison family contain the spikes, after which $\mathbb{E}F(S)$ equals a set-independent offset minus one half of the determinant objective, up to a uniformly vanishing error. The latter is an objective-level determinant-regret result and does not imply that the two rules select the same set or ordering.

For clarity, recall the notation used in the previous section,

$$\lambda(q) := \frac{\phi(q)}{\Phi(-q)}, \qquad \kappa(q) := \lambda(q)(\lambda(q) - q),$$
$$G_S := \sum_{i \in S} h_i h_i^\top, \qquad \Sigma_S := \sum_{i \in S} \kappa(q_i) h_i h_i^\top, \qquad b_S := \sum_{i \in S} \lambda(q_i) h_i.$$

The surrogate can be written as

$$F(S) := \sum_{i \in S} \log \Phi(-q_i) - \frac{1}{2} \log\det(I_d + \Sigma_S) + \frac{1}{2} b_S^\top (I_d + \Sigma_S)^{-1} b_S. \tag{26}$$

We also write $\lambda_i = \lambda(q_i)$ and $\kappa_i = \kappa(q_i)$. Thus minimizing F is equivalent to maximizing $\widetilde{P} = 1 - \widetilde{L}$. We use k throughout for the selection budget, with $1 \le k \le N$ for a fixed candidate pool. For the large-budget regime, we let $k \to \infty$ along a sequence of candidate pools of sizes $N_k \ge k$; the dimension d, the spike set O, its size r, the dependence scale ρ, the bulk-quality law, and the spike qualities remain fixed across this sequence. The surrogate greedy rule may be expressed in either objective, since

$$\widetilde{P}(S \cup \{j\}) - \widetilde{P}(S) = \widetilde{L}(S) \left[1 - \exp\big(F(S \cup \{j\}) - F(S)\big)\right].$$

For a fixed current set S, maximizing this gain is exactly equivalent to minimizing the marginal change in F.

A common set of assumptions Consider the following set of assumptions on the quality scores and representations of a model.

(A1) **Bounded qualities with spikes.** There is a fixed nonempty set $O = \{o_1, \dots, o_r\}$ of $r \ge 1$ distinguished candidates. For $i \notin O$, the scores are independent draws $q_i \sim \mathcal{D}_q$, where $\mathcal{D}_q$ has support contained in a fixed interval $[u, w] \subseteq [-M, M]$ for a fixed finite M. We refer to these scores as the bulk scores and the distribution to be a bulk distribution. The fixed spike scores also lie in $[-M, M]$ and obey

$$q_{o_1} > q_{o_2} > \cdots > q_{o_r} > w.$$

(A2) **Weak dependence scale.** $h_i = \rho v_i$, where $\|v_i\|_2 \le 1$ and $0 < \rho \le 1$ is fixed across selection budgets. The vectors h_i are deterministic and independent of the ordinary quality scores. Write

$$\Lambda := \max_{q \in [-M, M]} \lambda(q) < \infty.$$

The small-budget result requires ρ to be sufficiently small relative to a quality gap. The large-budget result permits any fixed positive ρ.

(A3) **Balanced large subsets with nondegenerate covariance.** There are fixed constants $c_0, C_0, B_0 > 0$, independent of k and S, such that, for every sufficiently large k, $\mathcal{A}_k$ is a nonempty deterministic family of size-k subsets of $[N_k]$, chosen independently of the realized ordinary scores, and every $S \in \mathcal{A}_k$ satisfies

$$c_0 k I_d \preceq \sum_{i \in S} v_i v_i^\top \preceq C_0 k I_d, \qquad \left\| \sum_{i \in S \setminus O} v_i \right\|_2 \le B_0. \tag{27}$$

These are assumptions on the large subsets under comparison; they are not constraints imposed on the small-budget greedy steps. No common spike membership is imposed on $\mathcal{A}_k$.

Since ρ is fixed, (A3) implies $G_S \succeq c_0\rho^2 k I_d$. In particular, choosing ρ small does not destroy the large-budget covariance condition. It changes its fixed constant. All asymptotic constants below may depend on ρ and the constants in (A1)–(A3). We do not claim uniformity in a simultaneous limit $\rho \to 0$ and $k \to \infty$.

C.1 Small budgets: a quality gap yields probe selection

Order the realized quality scores as $q_{(1)} \geq \cdots \geq q_{(N)}$, and let T_k be its top-k set. For $1 \leq k < N$, define

$$\Gamma_k := \log \Phi(-q_{(k+1)}) - \log \Phi(-q_{(k)}). \tag{28}$$

Because $\frac{d}{dq} \log \Phi(-q) = -\lambda(q) < 0$, the cutoff is strict precisely when $\Gamma_k > 0$. This gap measures the quality loss from replacing the weakest member of T_k by the strongest candidate outside it.

Lemma 2. *Under assumptions (A1) and (A2), if $|S| = t$ and $j \notin S$, then*

$$|F(S \cup \{j\}) - F(S) - \log \Phi(-q_j)| \leq E_t, \qquad E_t := \frac{\rho^2}{2} + \Lambda^2 \left(t + \frac{1}{2}\right) \rho^2 + \frac{\Lambda^2 t^2 \rho^4}{2}. \tag{29}$$

Proof. We first record $0 < \kappa(q) < 1$. Positivity follows from $\lambda(q) > q$: this is immediate for $q \leq 0$, and for $q > 0$,

$$\Phi(-q) = \int_q^\infty \phi(t)\, dt < q^{-1} \int_q^\infty t\phi(t)\, dt = q^{-1}\phi(q).$$

For the upper bound, if $Z \sim N(0,1)$, consider the following two properties of Gaussian Random variables

$$\mathbb{E}[Z \mid Z > q] = \lambda(q), \qquad \mathbb{E}[Z^2 \mid Z > q] = 1 + q\lambda(q).$$

Hence, the conditional variance is

$$0 < \mathrm{Var}(Z \mid Z > q) = 1 + q\lambda(q) - \lambda(q)^2 = 1 - \kappa(q).$$

Let $A = I_d + \Sigma_S$, $a_j = h_j^\top A^{-1} h_j$, and $u_j = h_j^\top A^{-1} b_S$. The determinant lemma and the Sherman-Morrison formula give

$$\det(A + \kappa_j h_j h_j^\top) = \det(A)(1 + \kappa_j a_j), \;\; (A + \kappa_j h_j h_j^\top)^{-1} = A^{-1} - \frac{\kappa_j A^{-1} h_j h_j^\top A^{-1}}{1 + \kappa_j a_j}.$$

Expanding the quadratic term with $b_{S\cup\{j\}} = b_S + \lambda_j h_j$ consequently yields

$$F(S \cup \{j\}) - F(S) = \log \Phi(-q_j) - \frac{1}{2}\log(1 + \kappa_j a_j) + \frac{\lambda_j u_j + \lambda_j^2 a_j/2 - \kappa_j u_j^2/2}{1 + \kappa_j a_j}. \tag{30}$$

(A2) implies that $\|h_j\|_2 \leq \rho$. (A1) implies that $|q_i| \leq M$ which implies $\lambda_i \leq \Lambda$. Since $A \succeq I_d$, we have $0 \leq a_j \leq \rho^2$ and

$$|u_j| \leq \|h_j\|_2 \|A^{-1}\|_{\mathrm{op}} \|b_S\|_2 \leq t\Lambda\rho^2.$$

Using $\kappa_j < 1$, and $1 + \kappa_j a_j \geq 1$ in Eq. 30, the absolute correction to $\log \Phi(-q_j)$ is at most

$$\frac{\rho^2}{2} + \Lambda(t\Lambda\rho^2) + \frac{\Lambda^2\rho^2}{2} + \frac{t^2\Lambda^2\rho^4}{2} = E_t.$$

□

Theorem 5. *Under assumptions (A1) and (A2), if $k\rho^2 \leq 1$ and $\Gamma_k > (1 + 3\Lambda^2)k\rho^2$, then T_k is the unique size-k minimizer of F, and the surrogate greedy rule returns T_k after k steps.*

Proof. First, note that if $t \le k-1$ and $k\rho^2 \le 1$, then

$$2E_t \le k\rho^2 + 2\Lambda^2 k\rho^2 + \Lambda^2 k^2\rho^4 \le (1+3\Lambda^2)k\rho^2 < \Gamma_k.$$

For any $i \in T_k$, $j \notin T_k$, and common current set S excluding i, j, Lemma 2 gives, when $|S| = t \le k-1$,

$$F(S\cup\{i\}) - F(S\cup\{j\}) \le \log\Phi(-q_i) - \log\Phi(-q_j) + 2E_t \le -\Gamma_k + 2E_{k-1} < 0. \tag{31}$$

If the current set of the surrogate greedy rule is contained in T_k, at least one member of T_k is still available until the kth selection. By Eq. 31, every outside candidate is worse than an available inside candidate. Starting with the empty set and inducting therefore proves that the first k selections form T_k.

For global optimality, take a size-k set $R \ne T_k$. There exist $j \in R \setminus T_k$ and $i \in T_k \setminus R$. Apply Eq. 31 with $S = R\setminus\{j\}$. Replacing j by i strictly decreases F, so R cannot be an optimizer. This proves uniqueness of the optimizing set, even if some qualities within T_k are tied. □

The above result shows that the selected set from this objective aligns with the probe-style rule. To ensure that candidates are added in the same sequence as under that rule, the condition must hold at every cutoff $1,\dots,k$. Hence, when the quality gaps dominate the dependence correction, the surrogate greedy rule follows the probe-style ordering. The following claim formalizes this idea.

Corollary 2. *Under (A1) and (A2), suppose* $1 \le r < N$ *and define*

$$\Gamma_* := \min\left(\{\log\Phi(-q_{o_{j+1}}) - \log\Phi(-q_{o_j}) : 1 \le j < r\} \cup \{\log\Phi(-w) - \log\Phi(-q_{o_r})\}\right) > 0. \tag{32}$$

If $r\rho^2 \le 1$ *and* $(1+3\Lambda^2)r\rho^2 < \Gamma_*$*, then, for every realization of the ordinary scores, the first* r *selections of the surrogate greedy rule are* $o_1,\dots,o_r$ *in the order selected by the probe-style rule.*

Proof. All ordinary scores are at most w. Thus the top r candidates are the ordered spikes, and each of their first r cutoff gaps is at least Γ_*. Apply Theorem 5 separately at each cutoff. The nested top-k sets identify the full greedy order. □

This explains the role of spikes in the small-budget result. They supply a gap independent of the number of ordinary candidates while allowing the ordinary score variance to remain fixed.

C.2 Large k: the expected log objective gets closer to RepExp

We now keep $\rho > 0$ fixed and let k grow. Define the constants

$$\overline{c} := \mathbb{E}_{q\sim\mathcal{D}_q}\log\Phi(-q), \qquad \overline{\lambda} := \mathbb{E}_{q\sim\mathcal{D}_q}\lambda(q), \qquad \overline{\kappa} := \mathbb{E}_{q\sim\mathcal{D}_q}\kappa(q), \qquad v_\lambda := \mathrm{Var}_{q\sim\mathcal{D}_q}(\lambda(q)). \tag{33}$$

These expectations are finite by bounded support, and $\overline{\kappa} > 0$ since $\kappa(q) > 0$.

Theorem 6. *Under (A1)–(A3), uniformly over* $S \in \mathcal{A}_k$*,*

$$\mathbb{E}F(S) = k\overline{c} + \sum_{o\in S\cap O}\left(\log\Phi(-q_o) - \overline{c}\right) + \frac{d\,v_\lambda}{2\overline{\kappa}} - \frac{1}{2}\log\det(I_d + \overline{\kappa}G_S) + O(k^{-1/2}). \tag{34}$$

The expectation averages the ordinary scores for the fixed set S*.*

Note that this is a uniform approximation to the deterministic function $S \mapsto \mathbb{E}F(S)$, not a simultaneous sample-wise approximation to $S \mapsto F(S)$.

Proof. Fix $S \in \mathcal{A}_k$, and write $J = S\cap O$ and $B = S\setminus O$. The expectation of the scalar quality term is exactly

$$\mathbb{E}\sum_{i\in S}\log\Phi(-q_i) = k\overline{c} + \sum_{o\in J}(\log\Phi(-q_o) - \overline{c}). \tag{35}$$

We treat the determinant and quadratic terms separately. Define

$$G_B = \sum_{i\in B} h_i h_i^\top, \qquad \Sigma_J = \sum_{o\in J} \kappa(q_o) h_o h_o^\top, \qquad b_J = \sum_{o\in J} \lambda(q_o) h_o,$$
$$A_0 = I_d + \overline{\kappa} G_B + \Sigma_J, \qquad \Delta = \sum_{i\in B} (\kappa(q_i) - \overline{\kappa}) h_i h_i^\top, \qquad A = I_d + \Sigma_S = A_0 + \Delta.$$

There are at most r spike contributions. Removing them from G_S changes its operator norm by at most $r\rho^2$, so $G_B \succeq c_1 k I_d$ for all sufficiently large k, with $c_1 > 0$ depending on the fixed ρ. Compactness of the quality interval gives a positive lower bound on every $\kappa(q_i)$. Hence A, A_0, and every matrix on their connecting line segment have smallest eigenvalue at least $c_2 k$.

The determinant term. Independence and centering eliminate the cross terms in the squared Frobenius norm, yielding

$$\mathbb{E}\Delta = 0, \qquad \mathbb{E}\|\Delta\|_F^2 = \mathrm{Var}_{q\sim\mathcal{D}_q}(\kappa(q)) \sum_{i\in B} \|h_i\|_2^4 = O(k). \tag{36}$$

For $A_t = A_0 + t\Delta$, differentiation gives

$$\frac{d}{dt} \log\det A_t = \mathrm{tr}(A_t^{-1}\Delta), \qquad \frac{d^2}{dt^2} \log\det A_t = -\,\mathrm{tr}(A_t^{-1}\Delta A_t^{-1}\Delta).$$

The absolute second derivative is at most $\|A_t^{-1}\|_{\mathrm{op}}^2 \|\Delta\|_F^2 \le Ck^{-2}\|\Delta\|_F^2$. Taylor's formula and Eq. 36 therefore imply

$$\mathbb{E}\log\det A = \log\det A_0 + O(k^{-1}).$$

Also, $A_0 - (I_d + \overline{\kappa} G_S) = \sum_{o\in J} (\kappa(q_o) - \overline{\kappa}) h_o h_o^\top$ has bounded norm. The same line-segment derivative identity, with inverse norm $O(k^{-1})$, gives

$$\log\det A_0 = \log\det(I_d + \overline{\kappa} G_S) + O(k^{-1}).$$

The quadratic term. Decompose

$$b_S = m + \zeta, \qquad m = b_J + \overline{\lambda} \sum_{i\in B} h_i, \qquad \zeta = \sum_{i\in B} (\lambda(q_i) - \overline{\lambda}) h_i.$$

By balance and bounded spike contributions (Assumptions (A1) and (A3)), $\|m\| = O(1)$. Independence gives $\mathbb{E}\zeta = 0$ and $\mathbb{E}\zeta\zeta^\top = v_\lambda G_B$. Consequently,

$$\mathbb{E}[b_S^\top A_0^{-1} b_S] = m^\top A_0^{-1} m + v_\lambda \,\mathrm{tr}(A_0^{-1} G_B) = \frac{d\, v_\lambda}{\overline{\kappa}} + O(k^{-1}).$$

For the last equality, the first term is $O(k^{-1})$ and

$$A_0^{-1} G_B = \frac{1}{\overline{\kappa}} \big(I_d - A_0^{-1}(I_d + \Sigma_J)\big).$$

It remains to replace A_0^{-1} by the correlated random inverse A^{-1}. The resolvent identity yields $A^{-1} - A_0^{-1} = -A^{-1}\Delta A_0^{-1}$. Moreover, $\mathbb{E}\|b_S\|_2^4 = O(k^2)$. To justify this last estimate, write each coordinate of ζ as a sum of independent centered bounded variables X_i. Expanding its fourth power gives

$$\mathbb{E}\left(\sum_i X_i\right)^4 = \sum_i \mathbb{E}X_i^4 + 6\sum_{i<j} \mathbb{E}X_i^2 \mathbb{E}X_j^2 = O(k^2).$$

There are only d coordinates, and $(\sum_{a=1}^d z_a^2)^2 \le d\sum_{a=1}^d z_a^4$, so the vector estimate follows; adding $m = O(1)$ does not change its order. Cauchy–Schwarz now gives

$$\mathbb{E}\left|b_S^\top (A^{-1} - A_0^{-1}) b_S\right| \le \frac{C}{k^2} \mathbb{E}[\|b_S\|_2^2 \|\Delta\|_F] \le \frac{C}{k^2} \sqrt{\mathbb{E}\|b_S\|_2^4\, \mathbb{E}\|\Delta\|_F^2} = O(k^{-1/2}). \tag{37}$$

Combining Eq. 35–Eq. 37 with Eq. 26 proves the theorem. Every bound used only the common constants in (A1)–(A3), so the error is uniform over $\mathcal{A}_k$. □

The random weighted imbalance does not disappear, but its expected contribution becomes independent of the selected set. The spike correction in Eq. 34, on the other hand, remains membership-dependent. A finite number of spikes cannot simply be discarded from the objective when their inclusion is still a decision.

When the large-budget optimizer includes the spikes The preceding limits allow the selected spike set $J = S \cap O$ to vary. More specifically, the family $\mathcal{A}_k$ is independent of the bulk quality realizations, and the constants in the assumptions of Theorem 6 are uniform over $S \in \mathcal{A}_k$. However, a stronger result can be proven under a mild closure condition on the family of subsets $\mathcal{A}_k$.

Definition 2 (Closure under insertion of a missing spike)**.** *A family $\mathcal{A}_k$ of size-k subsets has the required closure property if, whenever $S \in \mathcal{A}_k$ and $o \in O \setminus S$, there is a bulk candidate $i \in S \setminus O$ such that*

$$S' = (S \setminus \{i\}) \cup \{o\} \in \mathcal{A}_k.$$

This condition only requires one permissible replacement for each missing spike. It is not automatic for an arbitrary family defined by a balance constraint, but there are concrete families for which it holds. Let $\mathcal{C}_k$ be any nonempty deterministic family of bulk-only size-k sets such that, for fixed constants $B_{\mathrm{core}}, c_{\mathrm{core}}, C_{\mathrm{core}} > 0$ independent of k,

$$\left\| \sum_{i \in T} v_i \right\|_2 \le B_{\mathrm{core}}, \qquad c_{\mathrm{core}} k I_d \preceq \sum_{i \in T} v_i v_i^\top \preceq C_{\mathrm{core}} k I_d, \qquad T \in \mathcal{C}_k. \tag{38}$$

As before, $h_i = \rho v_i$ with $\|v_i\|_2 \le 1$, and $\rho > 0$ is fixed. Define

$$\mathcal{A}_k = \{(T \setminus R) \cup J : T \in \mathcal{C}_k,\ J \subseteq O,\ R \subseteq T,\ |R| = |J|\}. \tag{39}$$

If $S = (T \setminus R) \cup J$ misses a spike o, replacing any $i \in T \setminus R$ by o gives the same representation with $R \cup \{i\}$ and $J \cup \{o\}$. Thus this family is closed in the sense of Definition 2, for $k \ge r$. Moreover, since at most r vectors are removed or inserted,

$$\left\| \sum_{i \in S \setminus O} h_i \right\|_2 \le \rho(B_{\mathrm{core}} + r), \tag{40}$$

$$\rho^2(c_{\mathrm{core}} k - r) I_d \preceq G_S \preceq \rho^2(C_{\mathrm{core}} k + r) I_d. \tag{41}$$

For the matrix bounds, use $0 \preceq v_i v_i^\top \preceq I_d$ for each removed or inserted vector. For all sufficiently large k, these imply (A3), for example with $B_0 = B_{\mathrm{core}} + r$, $c_0 = c_{\mathrm{core}}/2$, and $C_0 = C_{\mathrm{core}} + 1$. Concretely, $\mathcal{C}_k$ can consist of sets formed from antipodal pairs in the coordinate directions, with a positive fraction of the vectors in every direction. Their vector sums vanish, and their Gram matrices have all eigenvalues proportional to k. This provides nonempty families along the admissible even values of k.

Lemma 3 (Spikes are selected by the expected log-surrogate objective)**.** *Suppose $r \ge 1$, every spike has a fixed quality $q_o > w$, and $\mathcal{A}_k$ satisfies Definition 2. Under the assumptions of Theorem 6, every minimizer of $\mathbb{E}F(S)$ over $S \in \mathcal{A}_k$ contains all of O for all sufficiently large k.*

Proof. We first bound the effect of one replacement on a log determinant. Suppose $S' = (S \setminus \{i\}) \cup \{o\}$, where i is a bulk candidate, and both $S, S' \in \mathcal{A}_k$. Fix $\nu > 0$ and set

$$M(t) = I_d + \nu\big((1-t)G_S + tG_{S'}\big), \qquad 0 \le t \le 1.$$

The uniform covariance lower bound gives $M(t) \succeq (1 + \nu c k) I_d$ for some fixed $c > 0$. Also $G_{S'} - G_S = h_o h_o^\top - h_i h_i^\top$, so

$$\left| \frac{d}{dt} \log\det M(t) \right| = \nu \left| h_o^\top M(t)^{-1} h_o - h_i^\top M(t)^{-1} h_i \right| \le \frac{\nu(\|h_o\|_2^2 + \|h_i\|_2^2)}{1 + \nu c k} \le \frac{2\nu\rho^2}{1 + \nu c k}. \tag{42}$$

Integrating over $t \in [0,1]$ shows, uniformly in the permissible replacement,

$$|\log\det(I_d + \nu G_{S'}) - \log\det(I_d + \nu G_S)| = O(k^{-1}). \tag{43}$$

Write $c_o = \log\Phi(-q_o)$ and $\bar{c} = \mathbb{E}_{q \sim \mathcal{D}_q} \log\Phi(-q)$. Since $q \le w < q_o$ almost surely under $q \sim \mathcal{D}_q$ and $q \mapsto \log\Phi(-q)$ is strictly decreasing,

$$\eta := \min_{o \in O} (\bar{c} - c_o) > 0. \tag{44}$$

The set O is finite, so this minimum is a fixed positive number. If S misses spike o, choose the replacement supplied by the closure condition. The expansion in Theorem 6 and Eq. 43, with $\nu = \overline{\kappa}$, give

$$\begin{aligned}\mathbb{E}F(S') - \mathbb{E}F(S) &= c_o - \overline{c} - \frac{1}{2}\left[\log\det(I_d + \overline{\kappa}G_{S'}) - \log\det(I_d + \overline{\kappa}G_S)\right] + O(k^{-1/2}) \\ &\le -\eta + O(k^{-1/2}) < 0 \end{aligned} \tag{45}$$

for all sufficiently large k. The threshold is uniform over the family. Therefore no minimizer can omit a spike. □

Once all spikes are selected, their quality contribution is identical for every remaining comparison. The following consequence makes the resulting determinant approximation an optimization statement.

Corollary 3. *Let $\mathcal{A}_k^O = \{S \in \mathcal{A}_k : O \subseteq S\}$, and for $\nu > 0$ let*

$$D_\nu(S) = \log\det(I_d + \nu G_S), \qquad R_k(\nu) \in \arg\max_{S \in \mathcal{A}_k^O} D_\nu(S).$$

Under the assumptions of Lemma 3, let $\widehat{S}_k^{\log}$ minimize $\mathbb{E}F(S)$ over $\mathcal{A}_k$. Then, for all sufficiently large k,

$$0 \le D_{\overline{\kappa}}(R_k(\overline{\kappa})) - D_{\overline{\kappa}}(\widehat{S}_k^{\log}) = O(k^{-1/2}). \tag{46}$$

Proof. Lemma 3 places each optimizer in $\mathcal{A}_k^O$ for all sufficiently large k. On this family, Theorem 6 has the form

$$\mathbb{E}F(S) = C_k - \tfrac{1}{2}D_{\overline{\kappa}}(S) + e_k(S), \qquad \sup_{S \in \mathcal{A}_k^O} |e_k(S)| \le Ck^{-1/2},$$

where C_k is independent of S. Optimality of $\widehat{S}_k^{\log}$, compared with $R_k(\overline{\kappa})$, implies

$$\tfrac{1}{2}\left[D_{\overline{\kappa}}(R_k(\overline{\kappa})) - D_{\overline{\kappa}}(\widehat{S}_k^{\log})\right] \le e_k(R_k(\overline{\kappa})) - e_k(\widehat{S}_k^{\log}) \le 2Ck^{-1/2}.$$

This proves Eq. 46. □

Compatibility of the two budget regimes. We give a concrete construction satisfying the assumptions of both the small-budget probe result and the large-budget determinant results. Let N denote the total candidate-pool size, and consider a nested sequence of pools as $N \to \infty$. Each pool contains the same three spikes $O = \{o_1, o_2, o_3\}$, identified with the indices $\{1, 2, 3\}$, and a bulk set $\mathcal{B}_N = [N] \setminus O$ of size $N - 3$. The pools may be coupled by taking one infinite sequence of bulk candidates and letting $\mathcal{B}_N$ contain its first $N - 3$ members. Throughout the construction, the dimension d is fixed. For the large-budget regime, fix any exponent $\alpha \in (0, 1)$ and define the integer-valued budget schedule

$$k(N) := \lfloor N^\alpha \rfloor .$$

Then $k(N) \to \infty$ and $k(N)/N \to 0$ as $N \to \infty$, while $3 \le k(N) \le N - 3$ for all sufficiently large N. In the small-budget part of this construction, $k \in \{1, 2, 3\}$ denotes a fixed selection budget, whereas $k(N)$ is the budget used in the large-budget regime. We first choose the quality distribution, then fix the common scale ρ, and finally construct the growing pools and their large-budget comparison families.

For the quality model in (A1), draw the bulk scores independently from $\mathcal{D}_q = \mathrm{Unif}[-\nu, \nu]$, once for the nested sequence of pools, and let the three spike qualities be

$$q_{o_1} = 4\nu, \qquad q_{o_2} = 3\nu, \qquad q_{o_3} = 2\nu.$$

Here $\nu > 0$ is a fixed constant. Thus we may take $u = -\nu$, $w = \nu$, and $M = 4\nu$. Every spike has quality strictly above the entire ordinary quality distribution, and

$$\mathrm{Var}_{q \sim \mathcal{D}_q}(q) = \frac{\nu^2}{3} > 0.$$

In particular, the ordinary scores retain a fixed, nonzero variance as N grows.

Next, consider the small-budget quality gaps. Since $q \mapsto \log \Phi(-q)$ is strictly decreasing, the quantity

$$\Gamma_* = \min\big\{\log \Phi(-3\nu) - \log \Phi(-4\nu),\ \log \Phi(-2\nu) - \log \Phi(-3\nu),\ \log \Phi(-\nu) - \log \Phi(-2\nu)\big\}$$

is strictly positive. These gaps are independent of the number of ordinary candidates: adding more ordinary candidates cannot produce a score above ν. More explicitly, since $\frac{d}{dq} \log \Phi(-q) = -\lambda(q)$ and $\lambda'(q) = \kappa(q) > 0$, each difference in this minimum is the integral of λ over an interval of length ν contained in $[\nu, 4\nu]$. Consequently,

$$\Gamma_* \geq \nu\lambda(\nu) > 0.$$

Let $\Lambda = \max_{q \in [-4\nu, 4\nu]} \lambda(q)$, and choose the fixed scale $\rho > 0$ so that

$$0 < \rho^2 < \min\left\{\frac{1}{3},\ \frac{\Gamma_*}{3(1 + 3\Lambda^2)}\right\}.$$

Take deterministic vectors v_i with $\|v_i\|_2 \leq 1$, and set $h_i = \rho v_i$, as required by (A2). This choice gives

$$3\rho^2 \leq 1, \qquad 3(1 + 3\Lambda^2)\rho^2 < \Gamma_*.$$

For every $N > 3$ and every realization of the bulk scores, Corollary 2 therefore applies: the first three greedy selections are o_1, o_2, o_3 in that order. For each fixed $k \in \{1, 2, 3\}$, Theorem 5 also shows that the set $\{o_1, \ldots, o_k\}$ is the unique size-k surrogate optimum. Thus the small-budget conclusion holds uniformly as the pool size N grows.

It remains to construct large comparison sets satisfying (A3). Let the bulk directions belong to $\{\pm e_1, \ldots, \pm e_d\}$, where $e_1, \ldots, e_d$ are the standard basis vectors. An antipodal pair consists of two distinct candidates with directions e_j and $-e_j$. Its contributions to the vector sum and Gram matrix are, respectively,

$$e_j + (-e_j) = 0, \qquad e_j e_j^\top + (-e_j)(-e_j)^\top = 2 e_j e_j^\top.$$

Thus pairing cancels the vector sum while preserving covariance in direction e_j.

Write $b_\ell := 3 + \ell$ for the ℓth candidate in the infinite bulk sequence, and assign the directions of $b_1, b_2, \ldots$ in the repeating order

$$e_1, -e_1, e_2, -e_2, \ldots, e_d, -e_d, e_1, -e_1, \ldots.$$

Thus $\mathcal{B}_N = \{b_1, \ldots, b_{N-3}\}$. The directions are deterministic and independent of the random quality realizations, and the spike directions may be arbitrary deterministic vectors of norm at most one. For all sufficiently large N, define the bulk-only core and its singleton family by

$$T_N := \{b_1, \ldots, b_{k(N)}\}, \qquad \mathcal{C}_N := \{T_N\}.$$

Then $|T_N| = k(N)$. Let $m_{N,j}$ denote the number of complete antipodal pairs on axis j among the first $2\lfloor k(N)/2 \rfloor$ members of T_N. If $k(N)$ is even, T_N consists entirely of these pairs, and the cyclic construction gives

$$2\sum_{j=1}^{d} m_{N,j} = k(N), \qquad m_{N,j} \geq \frac{k(N)}{4d} \quad \text{for every } j$$

for all sufficiently large N. The lower bound means that at least a fraction $1/(2d)$ of the selected vectors lies on each axis. In this case,

$$\sum_{i \in T_N} v_i = 0, \qquad \frac{k(N)}{2d} I_d \preceq \sum_{i \in T_N} v_i v_i^\top = 2\,\mathrm{diag}(m_{N,1}, \ldots, m_{N,d}) \preceq k(N) I_d.$$

If $k(N)$ is odd, its first $k(N) - 1$ members form the complete pairs counted by $m_{N,j}$, and these counts satisfy

$$2\sum_{j=1}^{d} m_{N,j} = k(N) - 1, \qquad m_{N,j} \geq \frac{k(N) - 1}{4d}$$

for all sufficiently large N. The remaining member has direction e_{j_N} for an axis $j_N \in \{1, \dots, d\}$ determined by the cyclic ordering. Thus

$$\left\| \sum_{i \in T_N} v_i \right\|_2 = 1, \qquad \frac{k(N)-1}{2d} I_d \preceq \sum_{i \in T_N} v_i v_i^\top \preceq k(N) I_d.$$

Consequently, for both parities and all sufficiently large N, the core-family bounds hold with the fixed constants

$$B_{\mathrm{core}} = 1, \qquad c_{\mathrm{core}} = \frac{1}{4d}, \qquad C_{\mathrm{core}} = 1.$$

Now enlarge the core family using the construction in Eq. 39:

$$\mathcal{A}_N^{\mathrm{con}} = \{(T \setminus R) \cup J : T \in \mathcal{C}_N,\ J \subseteq O,\ R \subseteq T,\ |R| = |J|\}.$$

For such N, every $S \in \mathcal{A}_N^{\mathrm{con}}$ is a size-$k(N)$ subset of $[N]$. For any $S = (T \setminus R) \cup J$, its bulk part is $B = T \setminus R$. Since at most three vectors are removed,

$$\left\| \sum_{i \in B} v_i \right\|_2 \le \left\| \sum_{i \in T} v_i \right\|_2 + \left\| \sum_{i \in R} v_i \right\|_2 \le 4.$$

Furthermore, every removed or inserted rank-one matrix has operator norm at most one, so

$$\left(\frac{k(N)-1}{2d} - 3 \right) I_d \preceq \sum_{i \in S} v_i v_i^\top \preceq (k(N)+3) I_d.$$

For $k(N) \ge 12d + 2$, these imply (A3) with the expanded-family constants

$$B_0 = 4, \qquad c_0 = \frac{1}{4d}, \qquad C_0 = 2,$$

which are fixed independently of N. When applying the general large-budget results to this construction, their budget parameter k is instantiated as $k(N)$. To recover the generic budget-indexed notation, set

$$N_k := \left\lceil k^{1/\alpha} \right\rceil, \qquad \mathcal{A}_k := \mathcal{A}_{N_k}^{\mathrm{con}}.$$

For $0 < \alpha < 1$, this choice satisfies $\lfloor N_k^\alpha \rfloor = k$ and $N_k \ge k + 3$ for all sufficiently large k.

The expanded family also satisfies closure under insertion of a missing spike. Indeed, if $o \in O \setminus S$, choose an ordinary candidate $i \in T \setminus R$. Such a candidate exists because a missing spike implies $|J| = |R| \le 2$, whereas $|T| = k(N) \ge 3$. Then

$$(S \setminus \{i\}) \cup \{o\} = \big(T \setminus (R \cup \{i\})\big) \cup (J \cup \{o\}) \in \mathcal{A}_N^{\mathrm{con}}.$$

Consequently, the spike-inclusion lemma applies: for all sufficiently large N, every minimizer of $\mathbb{E}F(S)$ over $\mathcal{A}_N^{\mathrm{con}}$ contains all three spikes. Their quality contribution is then common to the remaining comparisons, yielding the determinant-regret conclusion in Corollary 3.

Finally, the small choice of ρ does not prevent the large-budget covariance from diverging. The preceding bounds give

$$G_S = \rho^2 \sum_{i \in S} v_i v_i^\top \succeq \rho^2 \left(\frac{k(N)-1}{2d} - 3 \right) I_d.$$

Since $\rho > 0$ is fixed and $k(N) = \Theta(N^\alpha)$, the smallest eigenvalue grows on the scale $N^\alpha \rho^2$. At the same time, $k(N)/N = \Theta(N^{\alpha-1}) \to 0$, so the selected fraction of the pool vanishes. For each fixed small budget $k \in \{1, 2, 3\}$, the dependence correction is dominated by the fixed quality gap through the condition $(1 + 3\Lambda^2) k \rho^2 < \Gamma_*$. For large budgets, the accumulated covariance instead grows without bound. Thus one fixed quality distribution and one fixed scale ρ support both regimes: probe selection for every fixed $k \le 3$, uniformly as N grows, and a determinant limit along the large-budget schedule $k(N) \to \infty$.

D BENCHMARK-STRATIFIED RESULTS FROM SECTION 4.2

This section reports the numbers behind Fig. 3. Table 1 gives, for every model, the pass@k of each pruning policy pooled over all test problems, and Tables 2 to 9 break these results down by benchmark, one table per model. All tables use the setting of Section 4.1, with a pool of $N = 128$ rollouts, a cutoff of $T = 4000$ tokens and eight-fold cross-validation repeated 20 times. Each entry is the average pass@k over the repetitions, in percent, and the value in parentheses is its standard error over those repetitions.

Table 1: Pass@k (%), pooled over problems.

Qwen3-0.6B

	@1	@2	@4	@8	@16
Random (Pass@k)	24.0 (0.00)	34.4 (0.00)	45.4 (0.00)	56.5 (0.00)	67.6 (0.00)
Linear Probe	24.9 (0.27)	33.8 (0.25)	44.3 (0.21)	56.2 (0.30)	67.1 (0.27)
RepExp	23.9 (0.12)	33.5 (0.10)	45.0 (0.08)	55.9 (0.08)	67.0 (0.06)
PKTP	25.6 (0.29)	36.3 (0.22)	47.2 (0.29)	58.4 (0.31)	69.3 (0.28)

Qwen3-1.7B

	@1	@2	@4	@8	@16
Random (Pass@k)	39.5 (0.00)	51.5 (0.00)	62.6 (0.00)	72.5 (0.00)	81.0 (0.00)
Linear Probe	40.7 (0.28)	52.0 (0.24)	63.1 (0.27)	72.5 (0.18)	79.9 (0.20)
RepExp	39.6 (0.08)	51.6 (0.08)	63.9 (0.07)	74.6 (0.04)	82.2 (0.03)
PKTP	39.3 (0.35)	52.7 (0.29)	64.4 (0.24)	73.8 (0.23)	81.7 (0.21)

OpenMath-Nemotron-1.5B

	@1	@2	@4	@8	@16
Random (Pass@k)	47.1 (0.00)	59.3 (0.00)	69.3 (0.00)	77.3 (0.00)	83.9 (0.00)
Linear Probe	48.2 (0.26)	59.8 (0.23)	69.4 (0.26)	77.6 (0.18)	84.0 (0.16)
RepExp	47.3 (0.06)	57.9 (0.07)	68.9 (0.05)	76.8 (0.05)	84.7 (0.03)
PKTP	48.1 (0.32)	59.4 (0.28)	69.4 (0.20)	78.3 (0.19)	85.2 (0.17)

DeepSeek-R1-Distill-Qwen-1.5B

	@1	@2	@4	@8	@16
Random (Pass@k)	27.4 (0.00)	38.4 (0.00)	49.5 (0.00)	60.2 (0.00)	70.3 (0.00)
Linear Probe	30.1 (0.22)	39.8 (0.21)	49.9 (0.20)	59.4 (0.22)	69.7 (0.24)
RepExp	27.3 (0.08)	38.5 (0.07)	49.8 (0.04)	60.6 (0.06)	72.6 (0.03)
PKTP	30.3 (0.28)	41.4 (0.18)	52.9 (0.28)	62.5 (0.21)	71.7 (0.23)

Qwen3-8B

	@1	@2	@4	@8	@16
Random (Pass@k)	46.9 (0.00)	58.0 (0.00)	67.2 (0.00)	75.2 (0.00)	82.3 (0.00)
Linear Probe	49.2 (0.35)	59.2 (0.28)	67.4 (0.23)	74.4 (0.20)	81.2 (0.19)
RepExp	46.9 (0.09)	57.4 (0.06)	68.1 (0.07)	75.9 (0.04)	83.0 (0.04)
PKTP	47.3 (0.23)	58.7 (0.33)	68.6 (0.29)	76.0 (0.22)	83.0 (0.26)

OpenMath-Nemotron-7B

	@1	@2	@4	@8	@16
Random (Pass@k)	44.0 (0.00)	55.5 (0.00)	66.0 (0.00)	75.2 (0.00)	83.1 (0.00)
Linear Probe	46.3 (0.21)	57.5 (0.22)	67.5 (0.23)	76.3 (0.16)	83.6 (0.17)
RepExp	44.0 (0.11)	56.8 (0.07)	66.4 (0.07)	76.3 (0.04)	84.7 (0.05)
PKTP	47.4 (0.29)	59.1 (0.35)	68.2 (0.29)	76.7 (0.24)	84.3 (0.21)

DeepSeek-R1-Distill-Qwen-7B

	@1	@2	@4	@8	@16
Random (Pass@k)	37.5 (0.00)	48.3 (0.00)	58.5 (0.00)	68.2 (0.00)	77.5 (0.00)
Linear Probe	37.4 (0.17)	46.7 (0.27)	56.4 (0.25)	66.0 (0.22)	76.6 (0.32)
RepExp	37.4 (0.08)	47.2 (0.08)	58.6 (0.07)	70.0 (0.04)	79.7 (0.04)
PKTP	38.6 (0.27)	49.5 (0.24)	59.7 (0.25)	70.0 (0.31)	79.3 (0.28)

Skywork-OR1-Math-7B

	@1	@2	@4	@8	@16
Random (Pass@k)	42.5 (0.00)	53.1 (0.00)	62.7 (0.00)	71.5 (0.00)	79.8 (0.00)
Linear Probe	44.4 (0.24)	54.2 (0.22)	62.9 (0.18)	71.3 (0.29)	79.4 (0.22)
RepExp	42.5 (0.09)	54.5 (0.09)	64.5 (0.06)	72.1 (0.04)	79.4 (0.05)
PKTP	44.7 (0.34)	55.7 (0.22)	65.6 (0.26)	73.2 (0.29)	80.3 (0.22)

Table 2: Per-benchmark pass@k (%), Qwen3-0.6B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	23.2 (0.00)	33.8 (0.00)	45.5 (0.00)	58.3 (0.00)	71.2 (0.00)
Linear Probe	26.6 (0.70)	35.6 (0.61)	43.6 (0.84)	56.5 (0.89)	72.5 (0.79)
RepExp	23.3 (0.28)	29.5 (0.23)	41.4 (0.20)	53.7 (0.22)	66.5 (0.15)
PKTP	25.2 (1.07)	36.7 (1.01)	50.2 (1.11)	64.2 (0.79)	75.5 (0.78)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	20.8 (0.00)	29.9 (0.00)	39.9 (0.00)	50.8 (0.00)	62.1 (0.00)
Linear Probe	27.1 (0.68)	33.3 (0.84)	43.0 (0.69)	49.9 (0.60)	60.5 (0.60)
RepExp	20.3 (0.24)	30.9 (0.25)	39.1 (0.20)	47.7 (0.20)	60.0 (0.14)
PKTP	26.3 (0.95)	39.0 (0.75)	48.5 (0.83)	55.0 (0.52)	63.4 (0.56)

BeyondAIME

	@1	@2	@4	@8	@16
Random (Pass@k)	11.7 (0.00)	19.5 (0.00)	29.6 (0.00)	41.2 (0.00)	54.2 (0.00)
Linear Probe	15.9 (0.76)	22.6 (0.76)	34.1 (0.86)	44.3 (1.09)	52.0 (0.89)
RepExp	11.7 (0.28)	19.2 (0.22)	30.8 (0.24)	44.8 (0.22)	58.8 (0.14)
PKTP	12.8 (0.72)	19.2 (0.76)	28.0 (0.76)	41.2 (1.02)	54.8 (0.68)

OlymMATH (Easy)

	@1	@2	@4	@8	@16
Random (Pass@k)	13.3 (0.00)	21.4 (0.00)	31.7 (0.00)	44.0 (0.00)	57.8 (0.00)
Linear Probe	13.1 (0.55)	19.6 (0.63)	29.0 (0.68)	41.2 (0.74)	53.0 (0.87)
RepExp	13.4 (0.18)	21.2 (0.18)	31.6 (0.18)	42.3 (0.21)	58.5 (0.15)
PKTP	13.3 (0.51)	19.5 (0.71)	28.5 (0.77)	43.1 (1.11)	59.7 (0.80)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	28.3 (0.00)	39.4 (0.00)	50.2 (0.00)	59.7 (0.00)	68.8 (0.00)
Linear Probe	27.1 (0.63)	36.7 (0.77)	48.4 (0.41)	60.9 (0.61)	70.1 (0.51)
RepExp	28.0 (0.20)	39.7 (0.17)	52.6 (0.12)	59.8 (0.11)	67.1 (0.10)
PKTP	28.7 (0.72)	40.2 (0.72)	52.5 (0.75)	62.5 (0.41)	71.9 (0.58)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	38.9 (0.00)	53.0 (0.00)	65.5 (0.00)	76.0 (0.00)	84.3 (0.00)
Linear Probe	36.5 (0.60)	49.9 (0.66)	61.6 (0.55)	75.7 (0.72)	86.1 (0.50)
RepExp	38.9 (0.28)	50.9 (0.20)	63.3 (0.21)	77.2 (0.11)	84.2 (0.08)
PKTP	41.0 (0.86)	56.5 (0.69)	67.6 (0.49)	76.5 (0.41)	83.2 (0.41)

Table 3: Per-benchmark pass@k (%), Qwen3-1.7B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	47.0 (0.00)	59.6 (0.00)	70.4 (0.00)	78.9 (0.00)	85.4 (0.00)
Linear Probe	51.7 (0.60)	63.3 (0.65)	73.2 (0.56)	81.0 (0.57)	87.3 (0.31)
RepExp	47.4 (0.21)	60.7 (0.23)	72.2 (0.19)	85.6 (0.09)	90.6 (0.07)
PKTP	44.0 (0.96)	58.3 (0.62)	71.5 (0.57)	80.3 (0.44)	87.2 (0.41)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	41.9 (0.00)	52.4 (0.00)	62.7 (0.00)	72.7 (0.00)	82.2 (0.00)
Linear Probe	45.8 (0.58)	56.0 (0.63)	62.1 (0.48)	69.9 (0.51)	78.1 (0.63)
RepExp	41.7 (0.21)	51.7 (0.22)	64.2 (0.18)	76.7 (0.18)	85.2 (0.09)
PKTP	42.7 (0.83)	54.8 (0.91)	66.6 (0.67)	74.7 (0.51)	82.1 (0.65)

BeyondAIME

	@1	@2	@4	@8	@16
Random (Pass@k)	35.0 (0.00)	45.7 (0.00)	55.8 (0.00)	65.8 (0.00)	75.7 (0.00)
Linear Probe	34.2 (0.67)	44.4 (0.81)	55.6 (0.88)	65.9 (0.60)	76.2 (0.60)
RepExp	34.9 (0.23)	41.6 (0.24)	55.2 (0.14)	66.7 (0.14)	77.3 (0.09)
PKTP	32.0 (0.79)	43.7 (0.70)	54.3 (0.74)	65.2 (0.61)	76.0 (0.75)

OlymMATH (Easy)

	@1	@2	@4	@8	@16
Random (Pass@k)	42.4 (0.00)	57.5 (0.00)	71.3 (0.00)	81.8 (0.00)	89.6 (0.00)
Linear Probe	41.8 (0.54)	56.1 (0.70)	72.3 (0.65)	84.2 (0.51)	89.7 (0.33)
RepExp	42.6 (0.19)	56.6 (0.18)	71.3 (0.14)	79.7 (0.10)	89.0 (0.09)
PKTP	42.9 (0.71)	59.2 (0.84)	73.4 (0.56)	84.5 (0.46)	91.4 (0.37)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	31.9 (0.00)	42.0 (0.00)	51.8 (0.00)	61.2 (0.00)	70.3 (0.00)
Linear Probe	27.8 (0.44)	38.5 (0.56)	49.7 (0.53)	59.5 (0.25)	66.4 (0.50)
RepExp	31.9 (0.17)	44.2 (0.16)	55.1 (0.11)	64.4 (0.09)	70.1 (0.08)
PKTP	31.3 (0.70)	42.1 (0.41)	51.2 (0.38)	61.2 (0.47)	70.2 (0.26)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	41.5 (0.00)	54.2 (0.00)	65.8 (0.00)	76.3 (0.00)	84.9 (0.00)
Linear Probe	48.8 (0.70)	58.9 (0.52)	68.6 (0.64)	76.3 (0.65)	84.6 (0.40)
RepExp	41.6 (0.24)	55.8 (0.20)	66.4 (0.19)	76.9 (0.16)	85.0 (0.10)
PKTP	45.7 (0.76)	60.8 (0.59)	72.3 (0.65)	79.2 (0.48)	85.4 (0.55)

Table 4: Per-benchmark pass@k (%), OpenMath-Nemotron-1.5B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	59.1 (0.00)	72.1 (0.00)	80.8 (0.00)	86.2 (0.00)	90.1 (0.00)
Linear Probe	63.5 (0.68)	75.4 (0.43)	83.3 (0.45)	89.1 (0.37)	92.1 (0.25)
RepExp	59.1 (0.17)	70.7 (0.15)	77.6 (0.11)	84.7 (0.10)	91.6 (0.06)
PKTP	62.1 (0.61)	74.1 (0.75)	82.3 (0.41)	85.9 (0.33)	89.9 (0.36)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	41.3 (0.00)	51.2 (0.00)	60.5 (0.00)	69.1 (0.00)	76.9 (0.00)
Linear Probe	45.5 (0.60)	55.2 (0.69)	63.9 (0.69)	71.6 (0.59)	77.4 (0.35)
RepExp	41.5 (0.26)	51.5 (0.24)	63.7 (0.16)	72.2 (0.15)	80.0 (0.10)
PKTP	49.3 (0.64)	56.4 (0.71)	65.6 (0.72)	74.2 (0.45)	83.0 (0.40)

BeyondAIME

	@1	@2	@4	@8	@16
Random (Pass@k)	43.8 (0.00)	54.7 (0.00)	63.8 (0.00)	71.5 (0.00)	78.1 (0.00)
Linear Probe	46.0 (0.71)	56.8 (0.67)	65.7 (0.67)	74.1 (0.46)	81.1 (0.42)
RepExp	43.8 (0.16)	52.9 (0.14)	63.4 (0.12)	72.3 (0.13)	79.6 (0.12)
PKTP	43.8 (0.51)	53.9 (0.63)	62.7 (0.55)	72.2 (0.45)	80.9 (0.56)

OlymMATH (Easy)

	@1	@2	@4	@8	@16
Random (Pass@k)	54.7 (0.00)	69.9 (0.00)	81.4 (0.00)	88.7 (0.00)	92.9 (0.00)
Linear Probe	58.8 (0.61)	69.2 (0.59)	79.4 (0.50)	88.4 (0.48)	93.9 (0.20)
RepExp	54.9 (0.16)	66.1 (0.16)	75.4 (0.14)	83.2 (0.13)	89.9 (0.06)
PKTP	59.4 (0.53)	72.2 (0.61)	80.8 (0.42)	87.6 (0.32)	91.5 (0.24)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	41.2 (0.00)	52.5 (0.00)	62.8 (0.00)	72.1 (0.00)	80.6 (0.00)
Linear Probe	37.4 (0.38)	50.3 (0.39)	61.2 (0.47)	70.1 (0.32)	78.6 (0.35)
RepExp	41.2 (0.15)	51.8 (0.15)	65.0 (0.13)	72.1 (0.08)	81.9 (0.04)
PKTP	38.0 (0.59)	50.5 (0.49)	62.9 (0.66)	74.6 (0.45)	82.2 (0.30)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	44.7 (0.00)	56.7 (0.00)	67.2 (0.00)	75.9 (0.00)	83.6 (0.00)
Linear Probe	43.8 (0.80)	56.4 (0.82)	66.0 (0.55)	74.4 (0.51)	81.7 (0.59)
RepExp	45.0 (0.21)	55.9 (0.22)	69.0 (0.19)	77.5 (0.13)	85.7 (0.09)
PKTP	41.5 (0.80)	52.5 (0.79)	63.1 (0.78)	74.9 (0.52)	84.2 (0.41)

Table 5: Per-benchmark pass@k (%), DeepSeek-R1-Distill-Qwen-1.5B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	31.8 (0.00)	42.5 (0.00)	53.1 (0.00)	63.4 (0.00)	73.2 (0.00)
Linear Probe	39.3 (0.55)	47.2 (0.41)	57.0 (0.57)	65.4 (0.59)	75.3 (0.50)
RepExp	31.9 (0.21)	38.9 (0.13)	50.2 (0.15)	63.1 (0.18)	73.8 (0.11)
PKTP	37.2 (0.67)	48.9 (0.84)	59.3 (0.73)	67.8 (0.63)	76.1 (0.42)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	27.0 (0.00)	37.1 (0.00)	47.5 (0.00)	57.3 (0.00)	66.9 (0.00)
Linear Probe	34.6 (0.92)	45.2 (0.73)	54.2 (0.62)	60.1 (0.50)	68.3 (0.65)
RepExp	26.8 (0.34)	36.1 (0.28)	47.7 (0.15)	54.0 (0.11)	67.0 (0.14)
PKTP	34.1 (0.73)	43.3 (0.72)	49.5 (0.69)	56.0 (0.65)	64.1 (0.72)

BeyondAIME

	@1	@2	@4	@8	@16
Random (Pass@k)	16.2 (0.00)	24.6 (0.00)	34.5 (0.00)	45.9 (0.00)	58.4 (0.00)
Linear Probe	17.0 (0.34)	23.2 (0.57)	29.4 (0.74)	38.3 (0.69)	51.3 (0.67)
RepExp	16.0 (0.16)	22.5 (0.13)	31.8 (0.15)	45.9 (0.21)	62.1 (0.12)
PKTP	19.2 (0.55)	25.6 (0.61)	36.3 (0.64)	46.5 (0.54)	58.2 (0.89)

OlymMATH (Easy)

	@1	@2	@4	@8	@16
Random (Pass@k)	17.2 (0.00)	27.8 (0.00)	40.8 (0.00)	54.4 (0.00)	67.3 (0.00)
Linear Probe	16.1 (0.45)	27.5 (0.68)	41.0 (0.58)	50.9 (0.64)	62.8 (0.48)
RepExp	16.9 (0.22)	31.9 (0.21)	42.4 (0.19)	53.6 (0.15)	68.7 (0.12)
PKTP	18.5 (0.63)	29.0 (0.99)	41.9 (0.91)	55.7 (0.56)	69.9 (0.38)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	29.3 (0.00)	40.9 (0.00)	51.5 (0.00)	60.7 (0.00)	69.7 (0.00)
Linear Probe	32.5 (0.42)	42.3 (0.46)	52.0 (0.50)	62.3 (0.51)	72.3 (0.42)
RepExp	29.2 (0.14)	42.5 (0.16)	54.7 (0.12)	63.3 (0.10)	74.5 (0.07)
PKTP	31.4 (0.57)	44.9 (0.65)	57.8 (0.53)	65.2 (0.30)	71.1 (0.40)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	39.8 (0.00)	53.1 (0.00)	65.2 (0.00)	75.3 (0.00)	82.7 (0.00)
Linear Probe	40.4 (0.48)	52.0 (0.68)	63.0 (0.74)	74.2 (0.56)	83.4 (0.64)
RepExp	40.0 (0.19)	52.9 (0.16)	64.8 (0.12)	77.1 (0.13)	84.2 (0.10)
PKTP	40.6 (0.82)	53.7 (0.67)	66.6 (0.59)	77.5 (0.51)	85.5 (0.39)

Table 6: Per-benchmark pass@k (%), Qwen3-8B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	70.2 (0.00)	80.6 (0.00)	86.3 (0.00)	90.1 (0.00)	93.1 (0.00)
Linear Probe	69.3 (0.64)	77.7 (0.50)	83.6 (0.34)	87.3 (0.32)	90.8 (0.38)
RepExp	70.0 (0.28)	80.9 (0.12)	86.6 (0.10)	91.0 (0.06)	92.8 (0.07)
PKTP	64.1 (0.69)	77.9 (0.55)	84.2 (0.43)	88.6 (0.43)	93.6 (0.41)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	46.2 (0.00)	56.3 (0.00)	65.3 (0.00)	73.1 (0.00)	80.3 (0.00)
Linear Probe	54.3 (0.57)	65.1 (0.61)	70.1 (0.57)	75.2 (0.33)	78.7 (0.33)
RepExp	46.3 (0.23)	56.5 (0.20)	65.5 (0.17)	74.9 (0.15)	84.3 (0.08)
PKTP	51.7 (0.68)	61.8 (0.49)	70.6 (0.54)	76.6 (0.59)	82.9 (0.50)

AMO-Bench

	@1	@2	@4	@8	@16
Random (Pass@k)	22.9 (0.00)	32.4 (0.00)	43.2 (0.00)	55.5 (0.00)	69.2 (0.00)
Linear Probe	25.7 (0.68)	34.1 (0.83)	47.2 (1.40)	57.8 (0.96)	71.0 (1.01)
RepExp	22.4 (0.30)	32.3 (0.35)	40.5 (0.33)	53.0 (0.29)	65.1 (0.26)
PKTP	27.2 (1.03)	37.9 (0.79)	48.4 (0.95)	60.5 (1.36)	73.8 (1.10)

OlymMATH (Hard)

	@1	@2	@4	@8	@16
Random (Pass@k)	30.2 (0.00)	44.0 (0.00)	57.8 (0.00)	70.0 (0.00)	80.4 (0.00)
Linear Probe	31.9 (0.67)	44.1 (0.80)	57.9 (0.60)	70.3 (0.53)	82.2 (0.67)
RepExp	30.3 (0.23)	46.2 (0.19)	61.7 (0.16)	73.8 (0.11)	83.2 (0.10)
PKTP	33.5 (0.80)	45.9 (0.73)	59.3 (0.78)	69.7 (0.76)	79.3 (0.56)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	46.0 (0.00)	56.9 (0.00)	65.8 (0.00)	73.1 (0.00)	79.7 (0.00)
Linear Probe	45.3 (0.72)	54.4 (0.51)	61.7 (0.43)	68.5 (0.43)	75.7 (0.43)
RepExp	46.0 (0.19)	55.6 (0.15)	68.6 (0.12)	74.5 (0.11)	79.9 (0.08)
PKTP	45.5 (0.52)	56.5 (0.58)	64.5 (0.44)	71.3 (0.33)	77.7 (0.47)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	55.3 (0.00)	65.9 (0.00)	74.0 (0.00)	80.3 (0.00)	85.7 (0.00)
Linear Probe	60.9 (0.68)	70.9 (0.59)	77.8 (0.67)	82.7 (0.50)	86.5 (0.29)
RepExp	55.3 (0.15)	61.8 (0.16)	70.7 (0.14)	76.5 (0.08)	84.8 (0.08)
PKTP	53.9 (1.00)	64.0 (0.76)	77.3 (0.74)	85.0 (0.63)	89.8 (0.32)

Table 7: Per-benchmark pass@k (%), OpenMath-Nemotron-7B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	68.7 (0.00)	80.0 (0.00)	87.0 (0.00)	91.7 (0.00)	95.2 (0.00)
Linear Probe	71.5 (0.67)	81.8 (0.47)	86.8 (0.35)	90.5 (0.35)	93.8 (0.35)
RepExp	69.1 (0.25)	81.1 (0.16)	89.6 (0.12)	93.5 (0.09)	98.1 (0.05)
PKTP	72.1 (0.71)	82.9 (0.29)	87.6 (0.31)	90.6 (0.45)	94.3 (0.42)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	41.4 (0.00)	52.3 (0.00)	62.9 (0.00)	73.1 (0.00)	82.4 (0.00)
Linear Probe	43.0 (0.50)	52.6 (0.70)	63.4 (0.46)	72.4 (0.40)	80.7 (0.63)
RepExp	41.4 (0.28)	54.1 (0.25)	62.5 (0.18)	73.6 (0.11)	83.6 (0.09)
PKTP	47.0 (0.68)	58.7 (0.88)	67.9 (0.55)	77.8 (0.44)	83.9 (0.51)

AMO-Bench

	@1	@2	@4	@8	@16
Random (Pass@k)	16.9 (0.00)	27.0 (0.00)	38.9 (0.00)	51.2 (0.00)	63.6 (0.00)
Linear Probe	18.8 (1.35)	29.2 (1.13)	41.8 (1.44)	54.0 (0.82)	66.3 (0.80)
RepExp	17.1 (0.35)	34.0 (0.33)	43.1 (0.28)	55.7 (0.27)	71.0 (0.19)
PKTP	20.5 (0.78)	27.0 (0.76)	35.5 (1.06)	47.7 (1.11)	67.7 (1.14)

OlymMATH (Hard)

	@1	@2	@4	@8	@16
Random (Pass@k)	24.5 (0.00)	36.9 (0.00)	50.4 (0.00)	63.1 (0.00)	73.7 (0.00)
Linear Probe	28.7 (0.87)	42.4 (0.68)	55.9 (0.68)	68.2 (0.52)	77.2 (0.51)
RepExp	24.4 (0.23)	38.6 (0.21)	53.3 (0.16)	67.1 (0.13)	75.4 (0.10)
PKTP	29.7 (0.75)	43.6 (0.90)	55.3 (0.65)	67.5 (0.45)	77.4 (0.50)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	47.4 (0.00)	59.1 (0.00)	69.6 (0.00)	78.7 (0.00)	86.1 (0.00)
Linear Probe	47.1 (0.36)	58.9 (0.48)	69.3 (0.41)	77.9 (0.39)	85.6 (0.29)
RepExp	47.2 (0.17)	57.9 (0.11)	65.8 (0.10)	77.4 (0.07)	87.4 (0.06)
PKTP	48.1 (0.44)	60.9 (0.57)	71.9 (0.61)	80.1 (0.43)	87.1 (0.22)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	49.5 (0.00)	61.1 (0.00)	70.6 (0.00)	78.3 (0.00)	84.9 (0.00)
Linear Probe	55.5 (0.82)	65.2 (0.75)	73.3 (0.50)	82.0 (0.55)	87.2 (0.35)
RepExp	49.6 (0.17)	64.1 (0.18)	74.0 (0.11)	78.9 (0.11)	83.9 (0.10)
PKTP	54.4 (0.81)	64.5 (0.65)	71.1 (0.43)	78.2 (0.54)	84.5 (0.42)

Table 8: Per-benchmark pass@k (%), DeepSeek-R1-Distill-Qwen-7B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	52.7 (0.00)	66.5 (0.00)	76.9 (0.00)	83.5 (0.00)	88.2 (0.00)
Linear Probe	51.6 (0.58)	65.4 (0.47)	75.1 (0.44)	80.9 (0.35)	87.4 (0.60)
RepExp	52.3 (0.25)	65.0 (0.25)	77.1 (0.16)	87.2 (0.09)	90.5 (0.04)
PKTP	55.2 (0.87)	68.6 (0.57)	78.1 (0.40)	86.1 (0.45)	90.2 (0.33)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	38.1 (0.00)	47.0 (0.00)	55.7 (0.00)	65.4 (0.00)	75.4 (0.00)
Linear Probe	41.5 (0.40)	45.4 (0.38)	51.7 (0.68)	62.3 (0.59)	77.1 (0.71)
RepExp	38.0 (0.16)	46.1 (0.20)	55.0 (0.22)	68.7 (0.19)	80.2 (0.10)
PKTP	41.0 (0.73)	48.6 (0.76)	57.9 (0.82)	68.3 (0.63)	77.5 (0.65)

AMO-Bench

	@1	@2	@4	@8	@16
Random (Pass@k)	7.0 (0.00)	12.5 (0.00)	21.1 (0.00)	33.3 (0.00)	48.8 (0.00)
Linear Probe	6.3 (0.59)	12.4 (0.79)	23.7 (1.11)	42.0 (1.52)	57.6 (1.26)
RepExp	7.4 (0.36)	12.2 (0.39)	26.2 (0.50)	41.7 (0.31)	60.4 (0.24)
PKTP	10.7 (1.32)	20.0 (1.82)	33.9 (1.40)	49.3 (1.45)	62.4 (1.27)

OlymMATH (Hard)

	@1	@2	@4	@8	@16
Random (Pass@k)	11.5 (0.00)	19.2 (0.00)	29.9 (0.00)	43.3 (0.00)	57.9 (0.00)
Linear Probe	13.7 (0.53)	21.2 (0.60)	28.0 (0.42)	37.7 (0.65)	52.7 (0.73)
RepExp	11.6 (0.18)	19.2 (0.20)	27.8 (0.17)	39.5 (0.16)	60.6 (0.10)
PKTP	10.6 (0.46)	20.3 (0.56)	29.4 (0.54)	43.4 (0.73)	58.0 (0.59)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	41.6 (0.00)	53.1 (0.00)	63.3 (0.00)	72.8 (0.00)	81.5 (0.00)
Linear Probe	40.6 (0.35)	50.2 (0.60)	60.5 (0.48)	70.6 (0.41)	80.8 (0.34)
RepExp	41.4 (0.14)	51.5 (0.11)	64.0 (0.12)	74.4 (0.08)	80.4 (0.04)
PKTP	42.8 (0.64)	54.0 (0.54)	64.4 (0.54)	74.0 (0.53)	83.5 (0.36)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	50.2 (0.00)	63.3 (0.00)	74.4 (0.00)	83.3 (0.00)	89.9 (0.00)
Linear Probe	47.2 (0.89)	59.3 (0.80)	73.3 (0.64)	82.1 (0.51)	87.8 (0.40)
RepExp	50.1 (0.30)	61.8 (0.22)	74.6 (0.17)	85.8 (0.11)	92.3 (0.03)
PKTP	49.4 (0.71)	61.7 (0.87)	73.0 (0.43)	81.1 (0.43)	89.2 (0.35)

Table 9: Per-benchmark pass@k (%), Skywork-OR1-Math-7B.

AIME

	@1	@2	@4	@8	@16
Random (Pass@k)	59.7 (0.00)	71.2 (0.00)	79.1 (0.00)	84.7 (0.00)	89.2 (0.00)
Linear Probe	62.3 (0.73)	70.7 (0.67)	80.2 (0.44)	85.3 (0.37)	89.8 (0.29)
RepExp	59.7 (0.23)	73.2 (0.16)	82.0 (0.12)	85.4 (0.08)	92.8 (0.08)
PKTP	63.6 (0.89)	75.4 (0.49)	80.6 (0.46)	84.6 (0.27)	88.4 (0.45)

HMMT

	@1	@2	@4	@8	@16
Random (Pass@k)	35.8 (0.00)	43.8 (0.00)	52.3 (0.00)	61.7 (0.00)	71.7 (0.00)
Linear Probe	40.3 (0.47)	48.2 (0.57)	54.5 (0.63)	62.5 (0.48)	70.8 (0.45)
RepExp	35.8 (0.18)	41.9 (0.20)	50.5 (0.16)	64.0 (0.13)	70.6 (0.10)
PKTP	37.1 (0.63)	46.2 (0.64)	56.1 (0.79)	64.8 (0.78)	71.5 (0.45)

AMO-Bench

	@1	@2	@4	@8	@16
Random (Pass@k)	11.9 (0.00)	18.7 (0.00)	26.5 (0.00)	35.0 (0.00)	45.0 (0.00)
Linear Probe	11.7 (0.84)	17.6 (0.80)	23.7 (0.74)	29.3 (0.76)	41.3 (0.92)
RepExp	12.1 (0.40)	21.7 (0.24)	30.9 (0.23)	38.4 (0.35)	53.1 (0.24)
PKTP	12.8 (0.97)	22.8 (1.30)	32.2 (1.15)	43.0 (1.44)	61.1 (1.65)

OlymMATH (Hard)

	@1	@2	@4	@8	@16
Random (Pass@k)	19.7 (0.00)	31.0 (0.00)	44.5 (0.00)	58.6 (0.00)	72.2 (0.00)
Linear Probe	19.0 (0.52)	28.5 (0.67)	39.8 (0.72)	55.6 (0.92)	72.1 (0.89)
RepExp	19.9 (0.18)	30.6 (0.23)	44.2 (0.20)	52.3 (0.14)	64.3 (0.20)
PKTP	20.4 (0.61)	31.7 (0.96)	45.6 (0.93)	58.0 (0.64)	72.2 (0.62)

Omni-MATH

	@1	@2	@4	@8	@16
Random (Pass@k)	50.2 (0.00)	61.4 (0.00)	70.7 (0.00)	78.8 (0.00)	85.6 (0.00)
Linear Probe	53.5 (0.60)	63.6 (0.40)	71.8 (0.42)	79.9 (0.41)	85.1 (0.32)
RepExp	50.1 (0.21)	63.5 (0.12)	73.4 (0.12)	81.5 (0.08)	84.8 (0.08)
PKTP	52.1 (0.63)	63.3 (0.41)	74.5 (0.38)	81.1 (0.42)	84.5 (0.26)

RIMO-N

	@1	@2	@4	@8	@16
Random (Pass@k)	53.1 (0.00)	65.1 (0.00)	74.6 (0.00)	82.4 (0.00)	89.2 (0.00)
Linear Probe	53.1 (0.61)	67.2 (0.58)	76.7 (0.51)	82.8 (0.45)	88.9 (0.46)
RepExp	53.1 (0.25)	69.5 (0.16)	78.4 (0.16)	83.4 (0.11)	90.3 (0.11)
PKTP	57.0 (1.01)	68.9 (0.82)	77.5 (0.60)	83.5 (0.59)	89.0 (0.46)

E RESULTS ON DIFFERENT POOL SIZES

This section repeats Fig. 3 for the other pool sizes, $N \in \{32, 48, 64, 80, 96, 112\}$, where the pool of each problem is drawn from its 128 generated rollouts and the cutoff is $T = 4000$ as in the main setting.

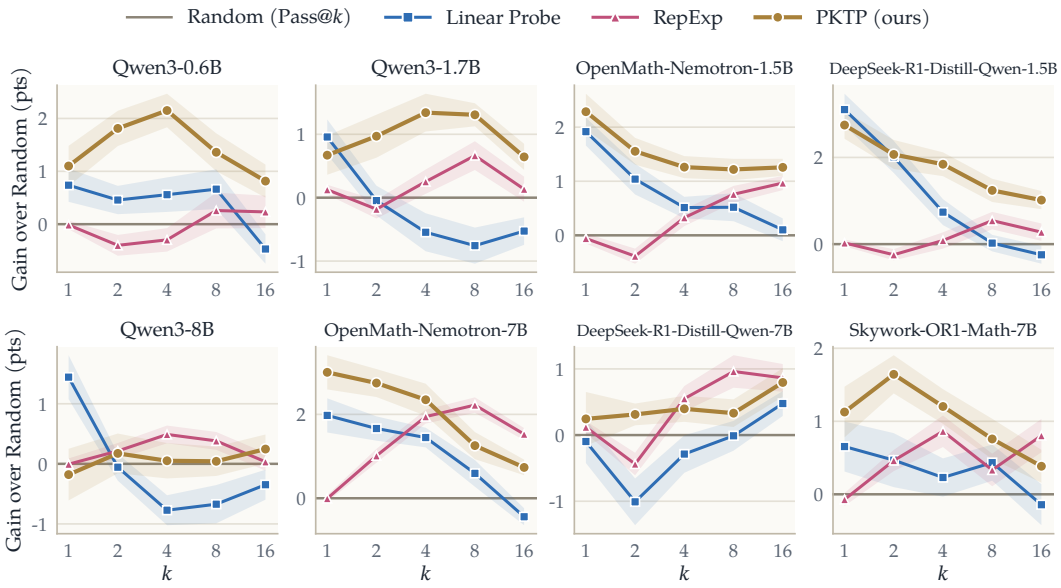


Figure 8: **Gain in pass@k over random pruning with $N = 32$.** As in Fig. 3, with a pool of $N = 32$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

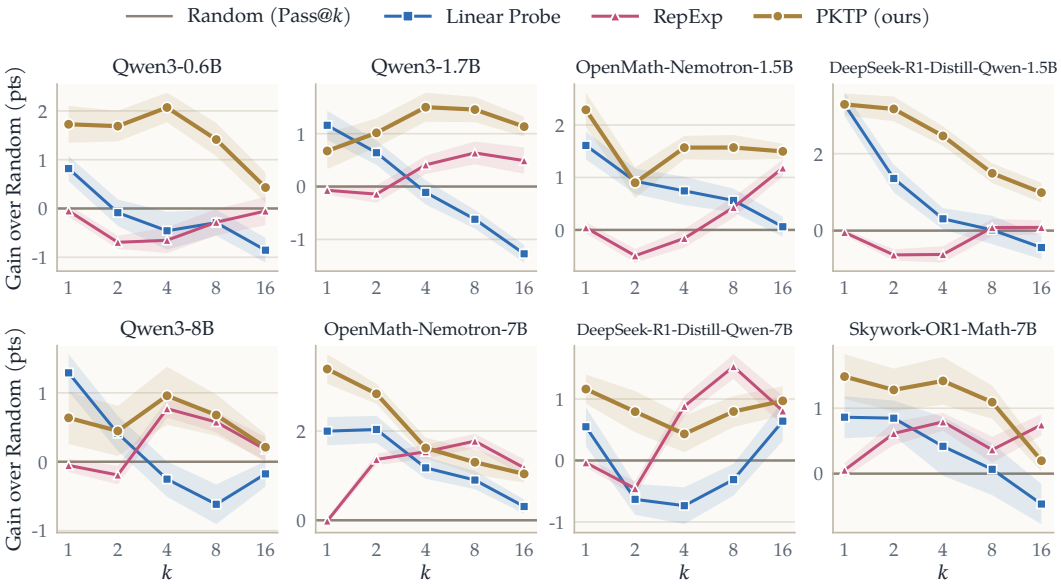


Figure 9: **Gain in pass@k over random pruning with $N = 48$.** As in Fig. 3, with a pool of $N = 48$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

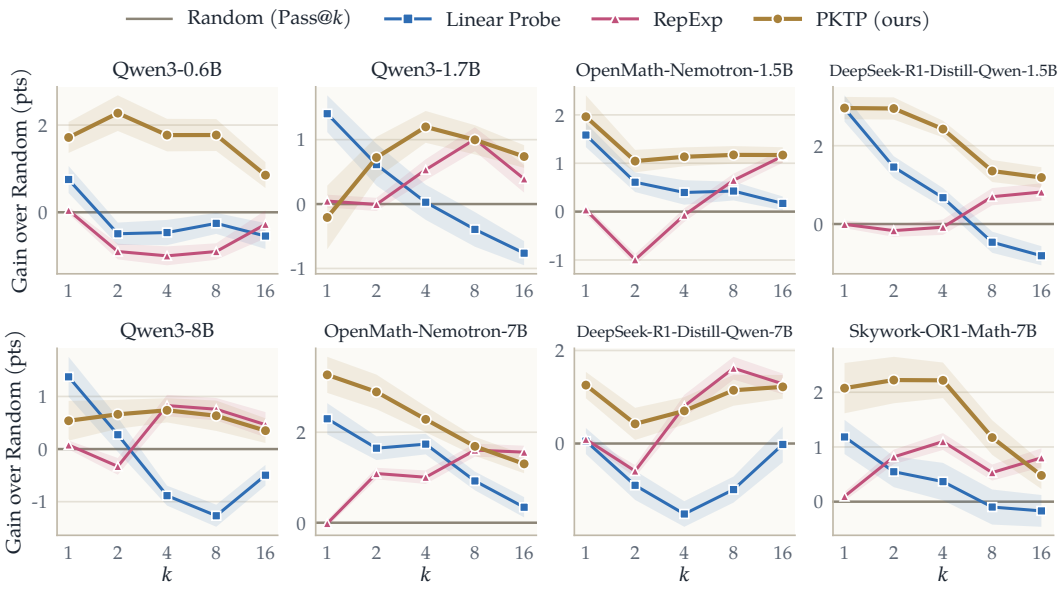


Figure 10: **Gain in** pass@k **over random pruning with** $N = 64$**.** As in Fig. 3, with a pool of $N = 64$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

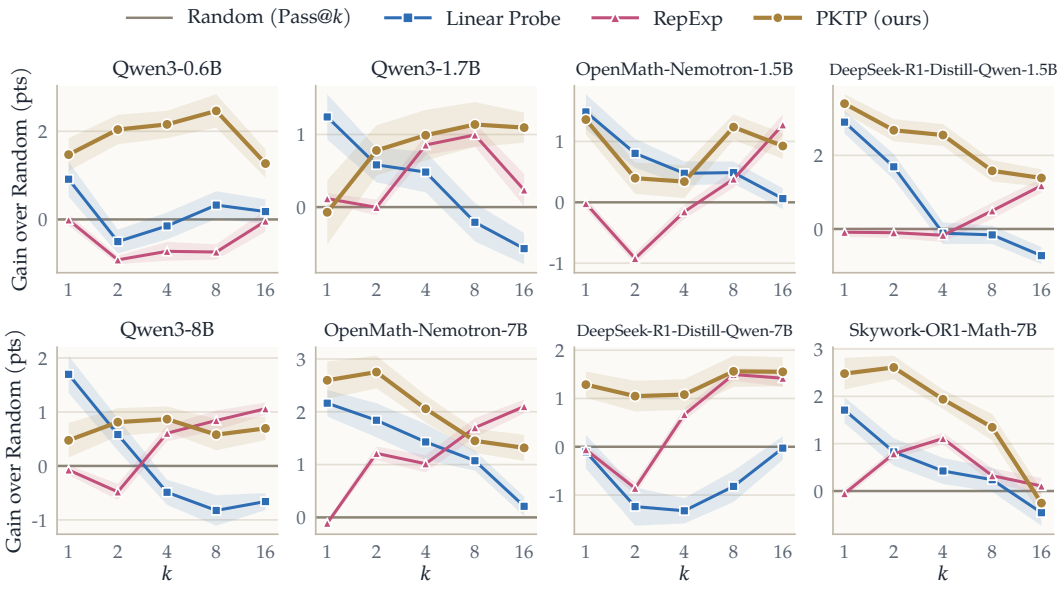


Figure 11: **Gain in** pass@k **over random pruning with** $N = 80$**.** As in Fig. 3, with a pool of $N = 80$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

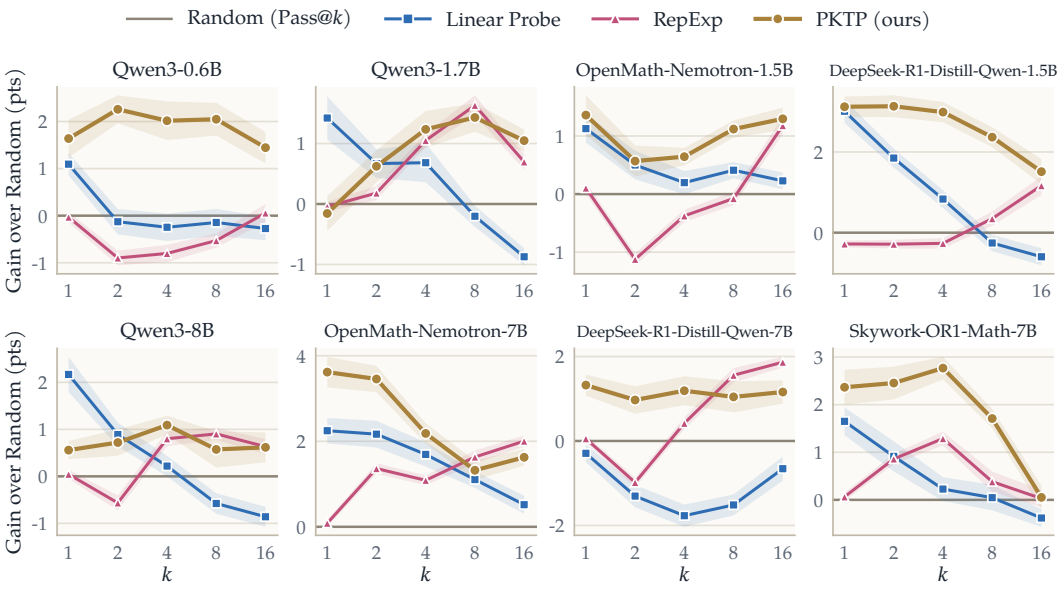


Figure 12: **Gain in** pass@k **over random pruning with** $N = 96$**.** As in Fig. 3, with a pool of $N = 96$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

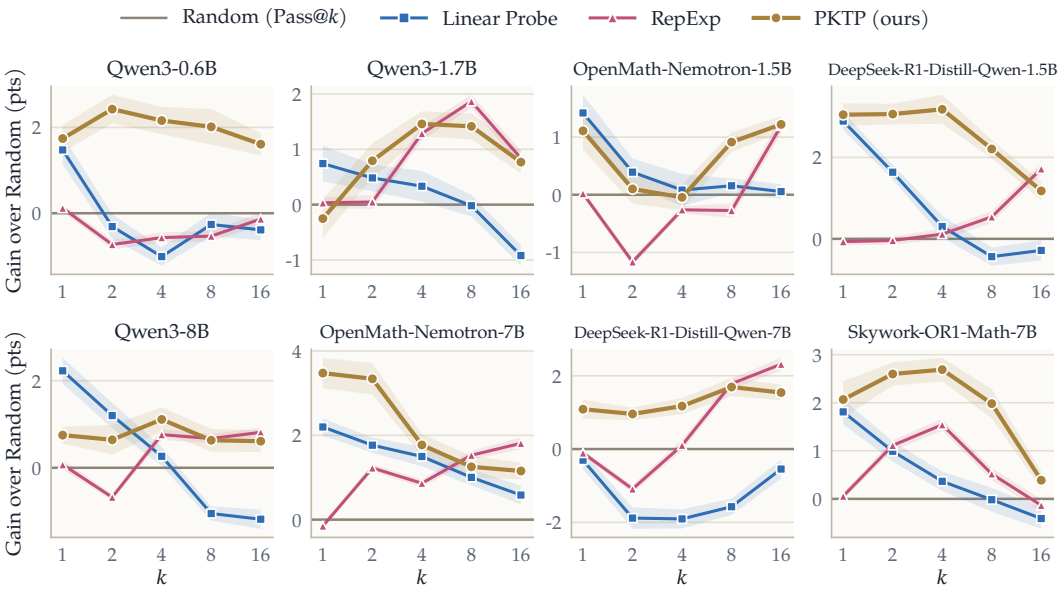


Figure 13: **Gain in** pass@k **over random pruning with** $N = 112$**.** As in Fig. 3, with a pool of $N = 112$ rollouts drawn from the 128 generated for each problem and $T = 4000$.

F REPRESENTATION DETAILS

For each rollout, let $x_t \in \mathbb{R}^d$ be the hidden state of the last transformer block of the model at the t-th generated token, where d is the hidden size of the model, which is 1024 for Qwen3-0.6B, 2048 for Qwen3-1.7B, 4096 for Qwen3-8B, 1536 for OpenMath-Nemotron-1.5B and DeepSeek-R1-Distill-Qwen-1.5B, and 3584 for OpenMath-Nemotron-7B, DeepSeek-R1-Distill-Qwen-7B and Skywork-OR1-Math-7B. The EMA follows $e_t = (1-\alpha)\, e_{t-1} + \alpha\, x_t$ with $\alpha = 10^{-3}$, and h_i is the normalized value of e_t after the first T generated tokens, or after all of them for a rollout that ends before the cutoff.

G EXTENDED RELATED WORK

Pruning rollouts at test time. Test-time scaling generates many rollouts per problem and aggregates them [3, 35, 44, 48], and a growing line of work saves compute by stopping unpromising rollouts early. One family uses the model's own confidence, such as token-level confidence or its self-estimate of success, to decide which rollouts to continue [13, 30]. Another scores partial rollouts with a reward model [36] or with a probe on the hidden states [1, 25, 54]. Yang et al. [51] concurrently use such a probe for a beam search over partial rollouts, pruning weak rollouts and branching new ones from the best prefixes. All of these score each rollout on its own and keep the highest-scoring ones, which is the pure exploitation extreme of Section 2. At the other extreme, RepExp [39] selects rollouts whose representations are least explained by those already chosen, without using their quality. PKTP sits between the two, since it models how the outcomes of the rollouts depend on each other and chooses the set with the highest predicted $\mathrm{pass@}k$.

Pass@k as an objective. Pass@k is a standard evaluation metric for reasoning and code generation, and several works optimize it during training. Walder and Karkhanis [41] derive low-variance policy-gradient estimators for $\mathrm{pass@}k$, Chen et al. [8] use $\mathrm{pass@}k$ as the reward to balance exploration and exploitation, and Chow et al. [10] fine-tune a model for best-of-N inference. Chen et al. [7] reward a group of responses for both quality and diversity, measuring diversity by the determinant of their similarity matrix. These are post-training methods, while PKTP leaves the model unchanged and optimizes $\mathrm{pass@}k$ at inference time by choosing which partial rollouts to continue. Concurrently, Wu et al. [47] also leave the reasoning model unchanged, but use $\mathrm{pass@}k$ as the reward to train a planner that writes a different outline for each rollout before it is generated. They thus change how the rollouts are sampled, while PKTP takes N independently sampled partial rollouts as given and chooses which k of them to continue.

Related objectives: maximizing at least one success. The same objective appears in information retrieval as the expected 1-call@k, the probability that at least one of the top k documents is relevant. Chen and Karger [6] optimize it greedily, and Sanner et al. [34] show that greedy optimization under a latent subtopic model of relevance leads to diverse rankings, in the same way that our latent factor leads to diverse kept sets. In Bayesian optimization, the batch probability of improvement is the probability that at least one point of a batch improves on the incumbent, and it is monotone and submodular, so it can be maximized greedily [32, 46]. Other methods balance quality and diversity with a hand-set trade-off, such as maximal marginal relevance [5] and determinantal point processes [22]. In contrast to these works, PKTP applies to partial reasoning rollouts, learns the joint distribution of their outcomes from their representations, and optimizes a closed-form surrogate that keeps the submodularity of the exact objective.