A SCORE IS NOT A SELECTOR: FROM PREDICTION TO BRANCH SELECTION AT TEST TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Process reward models (PRMs) are benchmarked on whether they can tell which reasoning steps are correct, yet test-time search uses them to choose which branch to continue. Using the official step labels of PRM800K and ProcessBench as a checker that knows exactly which steps are valid, we find that even this checker loses to four to eight sampled Qwen3-4B-Instruct continuations per branch graded against the reference answer, by $\mathbf{1.71}$ and $\mathbf{2.65}$ points of final correctness, mostly because labels cannot rank branches that share a label. We therefore let sampled continuations rank the branches and consult a PRM only on the remaining ties. For reference-graded continuations, we prove that one more continuation per branch is worth exactly its expected gain from breaking the current top ties, so a PRM on these ties can be priced in continuations. On ProcessBench, Qwen-PRM's tie-breaking is worth about one continuation per branch when continuations are few. By eight continuations its gain falls to about zero on both datasets, and the gain from official labels shrinks eightfold. A synthetic example shows that even a perfect checker can hurt on ties. A process score should therefore be judged by what it adds beyond sampled continuations. In a reference-free search that ranks branches by self-consistency, PRM tie-breaking adds 0.16 points per selection point.

1 INTRODUCTION

Test-time search samples candidate reasoning steps, scores them, and continues from the most promising branch (Yao et al., 2023; Snell et al., 2024; Wang et al., 2025). The score usually comes from a process reward model (PRM) (Lightman et al., 2023; He et al., 2024). Some PRMs learn from human judgments of step correctness (Uesato et al., 2022; Lightman et al., 2023). Others learn from whether sampled continuations reach the answer, sometimes filtered by an LLM judge (Wang et al., 2024; Luo et al., 2024; Zhang et al., 2025). In any case, PRMs are commonly benchmarked on how well they locate the first incorrect step (Zheng et al., 2025; Pandit et al., 2025). Choosing a branch, however, ultimately depends on whether the policy can still reach a correct answer from it, which we call *bounded recoverability*. Step validity and recoverability can disagree, since a branch with a flawed step can still end correctly and a valid branch can still fail (Zheng et al., 2025; Zhang et al., 2025; Ding et al., 2026). Earlier work argues that process rewards should track progress or utility under a policy (Setlur et al., 2024; Zhou et al., 2026), and outcome value models, a related family of learned scores, predict the chance of reaching a correct answer from a prefix without sampling continuations (Yu et al., 2024; Shan et al., 2026).

An ideal PRM trained for step validity would reproduce official step labels. To separate the validity target from a learned PRM's prediction error, we use the labels of PRM800K and ProcessBench as a stand-in for such a PRM, that is, a perfect checker that knows which steps are valid. We ask how well this checker chooses branches and what it adds on top of sampled continuations. At each selection point we consider a fixed set of existing candidates, let each selector choose, and score the choice on held-out continuations (Figure 1, §2.2). On PRM800K the label is a human rating of the candidate's next step given the earlier steps, and on ProcessBench it marks whether the evaluated prefix is free of annotated errors. Learned PRMs differ from the stand-in, since some also learn from sampled outcomes, so we also test them as selectors on the same fixed candidates.

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

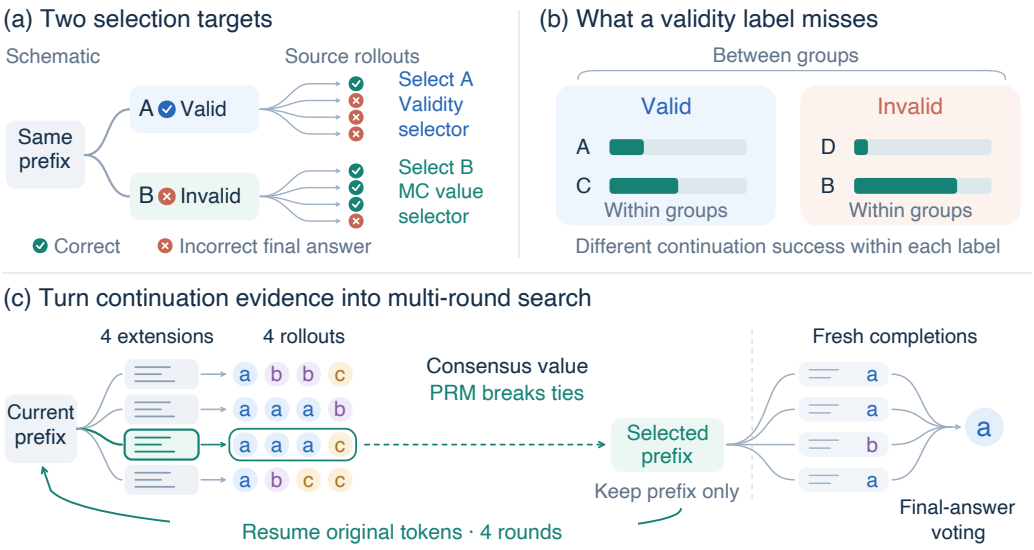

Figure 1: **From two selection targets to sequential search.** (a) Validity selection reads step labels and prefers A. MC value selection reads continuation success and prefers B. (b) Continuation success varies between validity groups and within them. (c) Reference-free search ranks four extensions by consensus value, with PRM tie-breaking, for four rounds. All panels are schematic.

We find that this checker still loses to a few sampled continuations. Selecting by official labels lowers final correctness by **1.71** points on PRM800K and **2.65** on ProcessBench compared with Monte Carlo (MC) value selection, which ranks branches by how many of four to eight sampled Qwen3-4B-Instruct continuations reach the reference answer. Labels tie every branch that shares a label, although those branches differ in how often they succeed. On PRM800K, 83% of MC value selection's gain over uniform choice comes from ranking branches within a label (Proposition 1). Learned PRMs sit between validity and recoverability. Qwen-PRM, trained on Monte Carlo outcomes filtered by a judge, captures 1.26 of MC value selection's 2.16 within-label points on PRM800K and none of the 2.69 on ProcessBench. Keeping only valid branches before ranking by MC value costs 0.01 points of final correctness on PRM800K and 0.22 on ProcessBench, so the labels still work as a filter. Validity selection gains 2.03 points over uniform under Qwen3-4B-Base, which rarely recovers from invalid prefixes, and at most 0.90 under five policies that recover more often.

If sampling selects better than even perfect labels, the useful question about a PRM is what it adds to sampling. The cleanest way to measure this lets sampling rank the branches and consults the PRM only on the ties it leaves, so every change the PRM makes is its own contribution. These tied branches are not a random sample of the pool. We construct an example showing that a perfect validity checker used this way can lower final correctness even when valid branches recover more often overall (Example C.1). For reference-graded continuations, one more continuation per branch helps only by resolving the current ties (Proposition F.2), so extra sampling is itself a tie-breaker against which a PRM can be priced. Measured in this currency, Qwen-PRM's tie-breaking on ProcessBench is worth about one more continuation per branch with up to four continuations, and its value shrinks as sampling grows. Breaking ties with official labels there adds **1.31** points with one continuation per branch but only **0.16** with eight. On fresh AIME 2024–2026 traces with sixteen candidates per selection point, every PRM's tie-breaking gain stays within 0.7 points of zero (Appendix E.2).

PRM tie-breaking is therefore most useful when continuations are few. With one continuation per branch, MC value selection with Qwen-PRM on its ties reaches 86.20% on PRM800K, above Qwen-PRM alone (85.66%) and MC value selection with four (85.96%). Without reference answers, we rank branches by *consensus value*, which grades continuations against the most common answer (Wang et al., 2023) and ranks branches exactly as MC value selection does whenever that answer is correct. On fixed candidates it still beats validity selection, 84.98% against 84.25% on PRM800K and 80.04% against 79.44% on ProcessBench.

Our three main findings build on one another.

- **Validity labels can constrain branches but not rank them.** Official step labels lose to a few sampled continuations as a branch selector, by 1.71 and 2.65 points on PRM800K and ProcessBench. An exact split of each selector's gain into a part between validity groups and a part within them (Proposition 1) traces most of the loss to branches that share a label. Used as a filter before sampling, the labels cost at most 0.22 points.
- **On top of sampling, a PRM is worth what it adds on the ties that sampling leaves.** For reference-graded sampling, we prove that one more continuation per branch helps only by resolving the current top-score ties, so extra sampling is itself a tie-breaker. At small budgets, Qwen-PRM's tie-breaking matches about one more continuation per branch on ProcessBench, and a synthetic example shows that even a perfect validity checker can hurt on ties once the candidate pool is large enough.
- **PRMs are therefore most useful when continuations are few.** From one to eight continuations per branch, Qwen-PRM's tie-breaking gain falls from 0.97 to -0.05 points on ProcessBench and from 1.35 to 0.05 on PRM800K. In a reference-free search with four continuations per branch, PRM tie-breaking adds 0.16 points per selection point on fresh AIME traces.

Together, these findings say that a process score should be judged by what it adds beyond sampled continuations rather than by how well it predicts step validity. Validity is best used to constrain the candidate set, and a PRM's added value lies in the ties that sampling leaves, which matter most when samples are few.

2 Validity, Recoverability, and Branch Selection

2.1 Step validity and bounded recoverability

Fix a problem x, prefix s, candidate extension z, verifier R, rollout policy π, and token budget H. *Bounded recoverability* is the probability that the policy finishes the branch correctly within budget:

$$V_{\pi,H}(s,z) = \Pr_{y\sim\pi_H(\cdot|x,s,z)}[R(x,s,z,y)=1]. \tag{1}$$

The mean correctness of m sampled continuations, or *rollouts*, gives the value estimate $\widehat{V}_{\pi,H,m}$. An outcome value model predicts it without sampling at test time (Yu et al., 2024).

Step validity grades whether the written reasoning is correct, the target of step-correctness PRMs. We write $C_j \in \{0,1\}$ for the validity label that each evaluation set supplies for candidate j. PRM800K gives a human rating of the candidate's next step, counted as valid unless it is -1. A ProcessBench prefix is valid if none of its steps contains an annotated error. Other sets use an *outcome-blind judge*, an LLM that sees only the problem and prefix, without the reference answer or any rollout (§3). Both official labels judge a candidate given its prefix. Aggregating validity scores to rank branches turns a checker into a selector. Validity concerns the annotated step or prefix. Recoverability also depends on the rollout policy and budget.

2.2 Fixed candidates and independent evaluation rollouts

A *state* is a branch-selection point: a problem x and its current reasoning prefix s. A **selector** uses candidate scores to choose the branch that the rollout policy will continue. Its **selection target** is the outcome used to evaluate that choice. Guided search repeatedly proposes candidates and applies a selector.

An *evaluation set* fixes states and ordered sets of K candidates, $Z = \{z_1,\dots,z_K\}$, shared by all selectors. From each candidate, we generate two disjoint sets of independent rollouts within the token budget. The *source rollouts* determine the choice. The *evaluation rollouts* measure its outcome. Exact ties receive equal weight. The uniform baseline weights all candidates equally.

For selection weights w_j, the three selection targets are

$$U_{\text{final}} = \sum_j w_j \widehat{V}_j^{\text{eval}}, \quad U_{\text{valid}} = \sum_j w_j C_j, \quad U_{\text{joint}} = \sum_j w_j C_j \widehat{V}_j^{\text{eval}}. \tag{2}$$

Label satisfaction U_{valid} is the probability that the selected candidate is valid. The *joint value* U_{joint} credits a choice only when the selected candidate is valid and its continuation succeeds.

Final correctness is mean accuracy on evaluation rollouts. On fixed candidates, we average single-continuation correctness over tied choices and both directions (the two rollout sets swap roles). In sequential selection, evaluation rollouts measure final correctness after each evaluated round.

MC value selection labels source rollouts with reference answers or tests and maximizes their Monte Carlo estimate $\widehat{V}^{\text{src}}$. It estimates recoverability V from finite samples. *Consensus value* needs no reference answer. It scores each candidate by the fraction of its source rollouts whose final answer equals the consensus answer $\hat{a}$. The consensus answer is the most common final answer among all candidates' source rollouts (Wang et al., 2023). *Validity selection* chooses uniformly among the candidates labeled valid, or among all candidates when none is. With official labels it maximizes label satisfaction by construction.

3 Experimental Design

3.1 Evaluation sets

Table 1 summarizes the four main fixed-candidate sets. One problem can contribute several states.

Table 1: Fixed-candidate evaluation sets. Each state is a problem with its current reasoning prefix (a branch-selection point in Section 2.2) and has K candidate branches. The last column lists each rollout policy (Qwen3-4B-Instruct-2507 or gpt-oss-20B) with the number m of source rollouts per candidate. Another m are evaluation rollouts.

Evaluation set	Problems	States	Candidates K	Rollouts m
PRM800K prefixes	73	357	5	Qwen3 4, gpt-oss 8
OpenR1	405	405	7	Qwen3 4
ProcessBench	322	322	2–4	Qwen3 8, gpt-oss 8
Code benchmarks	223	223	5	Qwen3 8, gpt-oss 8

In PRM800K, OpenR1 (Hugging Face, 2025), and code, candidates share a prefix. ProcessBench groups distinct three-step completions from the problem start. Code covers HumanEval+ and MBPP+ (Liu et al., 2023) and LiveCodeBench (Jain et al., 2024). We also evaluate reasoning prefixes on 64 SuperGPQA (M-A-P Team et al., 2025) scientific multiple-choice problems. ProcessBench's 3,400-trace error-localization task tests scorers' native validity prediction (Appendix Table 19).

3.2 Models, generation, and verification

We generate rollouts with Qwen3-4B-Instruct-2507 (Yang et al., 2025) and gpt-oss-20B (OpenAI, 2025) native chat at high reasoning effort. In the fixed-candidate sets, each prefix and candidate enter the rollout policy's prompt as text, and continuations start from this reformatted prefix. The search in §6 instead resumes the policy's own original token IDs. We refer to Qwen2.5-Math-PRM-7B as *Qwen-PRM* and write *Qwen3* for the Qwen3-4B-Instruct-2507 rollout policy in tables and figures. Qwen-PRM learns from hard step labels, positive when any of eight Qwen2.5-Math-Instruct completions from the step reaches the correct answer, kept only where these Monte Carlo labels and an LLM judge agree on the first error (Zhang et al., 2025). We also evaluate Qwen2.5-Math-7B-PRM800K (Qwen Team, 2025), the *human-label PRM*, fine-tuned on human step annotations only, as a second PRM. Per its model card, the fine-tuning data exclude PRM800K samples whose questions appear in the MATH test set, and all 73 PRM800K evaluation problems are MATH test problems. We further evaluate PUM (Zhou et al., 2026), a public checkpoint trained on prefix-utility gains across student policies. Qwen-PRM, the human-label PRM, PUM, and the 1.5B Skywork-PRM (He et al., 2024) retain their original tokenization and score aggregation, the product of step scores for Qwen-PRM.

On OpenR1, gpt-oss-20B at high reasoning effort judges each candidate once with the Hard2Verify prompt (Pandit et al., 2025). On code, GPT-5.4-mini judges each candidate in two draws. Neither

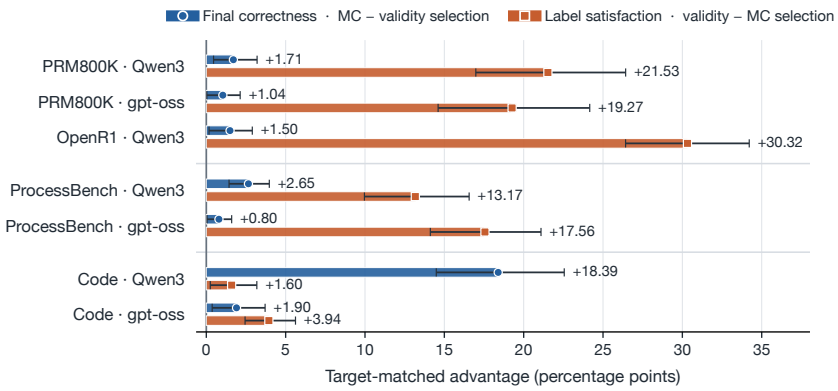

Figure 2: MC value selection versus validity selection on identical candidates. Blue measures the MC value selector's final-correctness advantage, orange the validity selector's label-satisfaction advantage. Intervals use problem-cluster bootstrap with three-target family-wise 95% coverage. Validity selection reads official labels on PRM800K/ProcessBench and judge labels on OpenR1. On code it uses continuous scores from one judge draw, evaluated against the other draw's labels. Except on code, validity selection maximizes label satisfaction by construction.

judge sees reference answers or rollout outcomes. Math answers are graded with `math-verify` 0.9.0 (Kydlíček, 2025). For SuperGPQA, the boxed letter must match the key.

3.3 Evaluation metrics and uncertainty estimation

Point estimates weight states equally, and differences are computed before rounding. We estimate confidence intervals with 20,000 bootstrap resamples, keeping states from the same problem together and including both source/evaluation directions. Figure 2 corrects its intervals across the three targets, and the other main-text intervals are pointwise.

4 Results

4.1 Each score on its own target

On 405 OpenR1 problems, the MC value selector ($m = 4$) raises final correctness from 46.18% to 47.84%, or **1.66 points** ($[+0.63, +2.74]$). On 64 SuperGPQA problems, it gains 18.38 points over uniform with eight rollouts per candidate (Appendix G.2). Qwen-PRM reaches 74.06 harmonic F1 on MATH first-error localization with a GSM8K-tuned threshold (Cobbe et al., 2021; Zheng et al., 2025) (Appendix Table 19).

4.2 The targets dissociate on identical candidates

Figure 2 shows the dissociation in all seven configurations. On PRM800K, the MC value selector beats validity selection by **1.71 final-correctness points**, while validity selection leads by **21.53 label-satisfaction points**. Official ProcessBench labels reproduce the pattern under both rollout policies. Official labels show the pattern arises without learned prediction error. The PRM800K final gap also survives binary, ordinal, and positive-only label orderings (Appendix Table 5).

The pattern also appears on OpenR1 prefixes and 223 code problems. On code, the Qwen3 MC value selector leads by **18.39 final-correctness points**, while validity selection leads by **1.60 label-satisfaction points** (intervals in Appendix Tables 20 and 22).

5 Why the Targets Dissociate

Every candidate has a step-validity label and an observed pass@m outcome, and Figure 3 shows substantial disagreement between them. Among invalid-labeled prefixes, 93.21% on PRM800K and

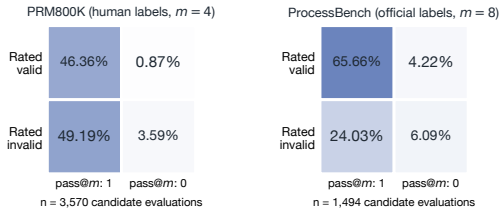

Figure 3: Joint distribution of step validity and pass@m, whether any of a candidate's m evaluation rollouts finishes correctly. The off-diagonal cells are invalid-labeled branches with a correct rollout and valid-labeled branches with none. Cells are candidate-weighted.

79.78% on ProcessBench succeed in at least one evaluation rollout. All rollouts fail for 1.84% and 6.03% of valid-labeled prefixes, respectively. Appendix Figure 5 shows one branch of each kind.

Binary validity also cannot rank candidates that share a label. On PRM800K, all candidates have the same label in 52.66% of states, where validity selection reduces to uniform choice. The two kinds of label tie differ. The MC value selector gains 0.62 points over uniform in the 23% of states with only valid candidates and 4.32 in the 30% with only invalid ones. The larger gain among all-invalid candidates shows the value of differences that a binary label cannot express.

Proposition 1 (Gain accounting identity)**.** *Fix a state with K candidates, labels $C_j \in \{0,1\}$, recoverabilities V_j, and any selector $w \in \Delta^K$. The labels split the candidates into a* valid group *and an* invalid group*. Let $u_j = 1/K$, $n_1 = \sum_j C_j$, $n_0 = K - n_1$, and $W_1 = \sum_{j:C_j=1} w_j$. Let $\bar{V}_1, \bar{V}_0$ be the mean recoverabilities of the two groups, assigning zero to the mean of an empty group. The gain over uniform is*

$$(w-u)^\top V = \left(W_1 - \tfrac{n_1}{K}\right)\left(\bar{V}_1 - \bar{V}_0\right) + \sum_j \left(w_j - \tfrac{1}{K}\right)\left(V_j - \bar{V}_{C_j}\right). \tag{3}$$

The first term is the between-group gain*: the extra weight on the valid group times the gap between group means. The second is the* within-group gain *from ranking candidates that share a label. For the validity selector, the within-group gain is zero. Its gain is $\frac{n_0}{K}(\bar{V}_1 - \bar{V}_0)$ on states containing both labels and zero otherwise. Raising all invalid-labeled recoverabilities by η therefore lowers this gain by exactly $\frac{n_0}{K}\eta$ per mixed-label state. (Proof in Appendix B.)*

By Proposition 1, the MC value selector's 2.61-point gain on PRM800K comprises 0.45 points between groups and 2.16 within groups. On ProcessBench, its 7.40-point gain splits into 4.70 and 2.69 points. The validity selector's gains, 0.90 and 4.74 points, are entirely between groups. On PRM800K, the 1.71-point gap equals MC value selection's within-group gain in single-label states (1.43 points) plus that in mixed-label states (0.73), minus the difference in between-group gains ($0.90 - 0.45 = 0.45$). On ProcessBench the terms are 1.96, 0.73, and 0.04 of 2.65 points (Appendix Table 6). Most of the within-group gain comes from invalid candidates on PRM800K (1.79 of 2.16 points) and valid candidates on ProcessBench (1.57 of 2.69). Across six rollout policies on PRM800K, validity selection gains 2.03 points over uniform for Qwen3-4B-Base, whose rollouts from invalid-labeled prefixes succeed 13.9% of the time, and at most 0.90 for the other five (Appendix Table 17).

What do other selectors capture? Table 2 compares seven selectors on the same candidates. On PRM800K, Qwen-PRM (85.66) captures about 89% of the MC value selector's gain over uniform (2.31 of 2.61 points), with 1.04 points between groups and 1.26 within groups. The human-label PRM splits its 1.73-point gain into 0.72 and 1.01 points. On ProcessBench, both PRMs gain mainly between groups. The human-label PRM gains 2.80 of its 2.94 points between groups, and Qwen-PRM gains 3.88 between groups with a within-group term of -0.01. This split fits its training label, positive when any of eight completions succeeds. Such a label separates branches that recover from those that never do, which carries most of the within-group gain on PRM800K. It does not separate branches with different success rates, which carry it on ProcessBench.

Shuffling Qwen-PRM's scores among candidates that share an official label removes exactly this within-group term. It lowers final correctness by 1.26 points on PRM800K and moves it by only 0.01 points on ProcessBench (Appendix Table 21).

Table 2: Fixed-candidate selection on PRM800K (357 states) and ProcessBench (322 states) under Qwen3 rollouts. Final correctness (%) uses independent evaluation rollouts, and label satisfaction (%) the selected candidate's official label. Bold marks the highest value in each column. Appendix Table 16 gives bootstrap standard deviations.

	PRM800K		ProcessBench	
Selector	Final ↑	Label ↑	Final ↑	Label ↑
Uniform	83.35	47.23	74.70	69.80
Validity selection	84.25	**70.31**	79.44	**90.99**
Validity filter → MC value	85.95	**70.31**	81.88	**90.99**
Human-label PRM	85.08	62.46	77.64	86.03
Qwen-PRM	85.66	62.98	78.56	87.89
Consensus value	84.98	49.10	80.04	77.14
MC value selection	**85.96**	48.78	**82.09**	77.82

Table 3: Final-correctness gain (points) from breaking MC-value ties on fixed candidates under Qwen3 rollouts, by source rollouts m per candidate. Tied rows give the share of selections with a tied top set. For PRM800K, $m = 8$ extends each source bank from four to eight rollouts per candidate and keeps the original evaluation banks. Appendix Table 11 adds intervals and two scorers.

		$m = 1$	$m = 2$	$m = 4$	$m = 8$
ProcessBench	Tied (%)	80.4	72.4	63.3	53.4
	Official labels	+1.31	+0.73	+0.37	+0.16
	Qwen-PRM	+0.97	+0.53	+0.24	−0.05
PRM800K	Tied (%)	96.5	92.4	86.0	75.4
	Official labels	+0.22	+0.03	−0.01	−0.13
	Qwen-PRM	+1.35	+0.98	+0.81	+0.05

Filtering for valid candidates before ranking by MC value keeps validity selection's label satisfaction and raises final correctness from 84.25% to 85.95% on PRM800K and from 79.44% to 81.88% on ProcessBench. If no candidate is valid, all candidates remain eligible. Relative to unfiltered MC value selection, the filter costs 0.01 and 0.22 points. By Appendix Proposition B.1, the cost is the loss when no valid candidate reaches the top, minus the gain of validity tie-breaking. Consensus value reads neither labels nor reference answers, yet keeps 62.5% of MC value selection's gain over uniform on PRM800K and 72.2% on ProcessBench (Appendix Table 18). It matches MC value selection whenever its consensus answer is correct (Corollary F.5) and is the primary score of §6.

Takeaway. Even perfect step labels lose to a few sampled continuations, because labels gain only between validity groups while sampling also ranks branches within them. Learned PRMs capture part of the within-group gain on PRM800K but not on ProcessBench.

6 What a PRM Adds on Top of Sampling

A tie is unresolved evidence. We pair a sampled primary score, MC value or consensus value, with an *auxiliary score* consulted only among the branches tied for the highest primary score, and resolve any remaining ties uniformly. Unlike the validity filter, which restricts the candidate set before ranking, a tie-breaker changes a choice only on ties. Its overall effect is the probability of a tie times its gain within ties, so it can be measured by itself (Proposition F.1). For MC value, one more source rollout per candidate is likewise worth exactly its expected gain on the current ties, since changes elsewhere cancel (Proposition F.2). The tied branches are exactly those that finite source rollouts could not separate, so they are not a random sample of the pool. In an exact construction, validity and recoverability are positively associated overall. Yet among branches whose source rollouts all succeed, the valid ones have lower recoverability, so a perfect validity checker favors the less recoverable branch. Validity tie-breaking then helps with two to four candidates and hurts with five or more (Example C.1), so a tie-breaker must be evaluated on the ties it breaks.

Official labels and Qwen-PRM help less with more sampling. Table 3 varies the number m of source rollouts per candidate on the fixed-candidate sets of §5. On ProcessBench, the official-label gain falls from 1.31 to 0.16 points, and all four intervals exclude zero. Both factors of the gain fall, since fewer choices are tied and the gain within ties drops from 1.63 to 0.30 points. Most of this decline comes from the recoverability gap between tied valid and invalid branches. Weighted by the invalid share as in Proposition F.1, this gap shrinks from 4.18 to 0.96 points, while the share of tied sets with both labels falls only from 39% to 32%. Qwen-PRM's gain moves to -0.05 on ProcessBench and falls from 1.35 to 0.05 points on PRM800K, where official labels add at most 0.22 points. Official labels predict this decline for scorers that act like validity checkers, such as Qwen-PRM on ProcessBench but not on PRM800K. The VGS value model (Wang et al., 2025), trained to predict success, still adds 0.46 points at $m = 8$ (Appendix Table 11).

A PRM priced in rollouts. With $f(m)$ the final correctness of MC value selection at m source rollouts, Proposition F.2 shows that $f(m{+}1) - f(m)$ is earned on the same ties that a tie-breaker resolves (Appendix Table 14), and Proposition F.4 bounds both gains by a term that falls exponentially in m. Drawing one more rollout only for the tied branches earns the same gain (Corollary F.3), so a PRM on these ties competes with at most one more rollout per tied branch. In final correctness, Qwen-PRM's gain on ProcessBench is close to that of one more rollout at $m = 1$, 2, and 4 (0.97, 0.53, and 0.24 points against 1.08, 0.51, and 0.23), and official labels are worth between one and two. On PRM800K at $m = 1$, Qwen-PRM's 1.35 points exceed the 1.11 of three more rollouts, while official labels are worth less than half of one (0.22 against 0.57). With one rollout per candidate, MC value selection with Qwen-PRM tie-breaking reaches 86.20% on PRM800K and 80.42% on ProcessBench, above Qwen-PRM alone (85.66% and 78.56%).

On fresh AIME 2024–2026 traces (Appendix E.2), with $K = 16$ candidates per selection point and eight evaluation rollouts each, Qwen-PRM breaking MC value ties changes final correctness by $+0.57$ points at $m = 1$ and -0.44 at $m = 16$. The tied share falls from 93.6% to 52.8%, and the change within ties moves from $+0.61$ to -0.83. On consensus value ties the endpoints are $+0.66$ and -0.35. Doubling the source rollouts from 16 to 32, scored with sixteen evaluation rollouts, raises the two primary scores' final correctness by 1.14 and 1.00 points, while Qwen-PRM's added gain stays within 0.2 points of zero (Appendix E.3).

Sequential selection from the original token history. We run 4 rounds with Qwen3-4B-Instruct-2507. Each round samples 4 extensions of up to 128 tokens. The MC value selector and consensus value use 4 source rollouts per candidate to select one extension. Qwen-PRM breaks their ties on mathematics, and a Qwen3-14B judge breaks them on GPQA Diamond (Rein et al., 2023). We resume the chosen extension's original tokens within a total 8,192-token budget, including the prefix.

After rounds 1, 2, and 4, we draw 4 evaluation rollouts, independent of the source rollouts used for selection. *Single* averages their individual correctness to estimate single-answer accuracy. *Voting* returns the most frequent of these four answers, choosing uniformly among ties. Direct generation uses the same model, initial prompt, per-trajectory token limit, and grading. Scorer-guided search continues the branch that Qwen-PRM, or the Qwen3-14B judge on GPQA Diamond, scores highest, and random branch continues a uniformly chosen one.

Table 4: Four-round final correctness (%) with Qwen3-4B-Instruct-2507. Rows specify branch selection. Single: single-answer accuracy. Voting: four-answer voting accuracy. Green: consensus value, with scorer tie-breaking added in the indented row. AIME averages all 90 questions equally. Appendix Table 7 gives MC value selection.

	MATH (200)		AIME 2024–26 (90)		GPQA Diamond (198)	
Method	Single	Voting	Single	Voting	Single	Voting
Direct generation	80.08	93.16	38.73	54.09	54.56	60.34
Random branch	78.88	89.75	38.06	50.93	54.67	59.85
Scorer-guided search	79.38	91.88	36.11	53.52	53.66	57.11
Consensus value	82.63	92.67	41.94	56.11	58.21	61.45
+ scorer tie-breaking	82.50	94.50	45.00	60.00	57.32	61.20

Complete search. Adding Qwen-PRM tie-breaking to consensus value changes voting accuracy by $+1.83$ points on MATH (Hendrycks et al., 2021) and $+3.89$ points on the 90 AIME 2024–2026 questions (Table 4). It changes single-answer accuracy by -0.12 and $+3.06$ points, with paired intervals and individual years in Appendix Tables 10 and 8. At the search's own setting of $K = 4$ and $m = 4$, Qwen-PRM adds 0.16 points per selection point on the fresh AIME traces ($[+0.02, +0.30]$), while in the complete search each choice also changes the prefixes that later rounds extend. Unlike on the fresh traces, at the search's own AIME 2026 selection points Qwen-PRM's gain on MC value ties moves from -0.09 to $+0.31$ points as m grows from 1 to 8 (Appendix F.5). Scorer-guided search, which needs no source rollouts, falls below direct generation in both Single and Voting on all three benchmarks.

Takeaway. On the ties that sampling leaves, Qwen-PRM is worth about one more continuation per branch on ProcessBench when continuations are few and less as sampling grows. At the search's budget it adds 0.16 points per selection point.

7 Related Work

What a process reward should measure. Prior work justifies policy advantages as process rewards (Jia et al., 2025) and ranks steps by Q-values (Li & Li, 2025). Others denoise rollout-derived step labels (Xie et al., 2026; Ding et al., 2025). Setlur et al. (2024) and Zhou et al. (2026) score a step or prefix by how much it raises a policy's success rate and argue that utility and correctness can diverge. Both build new scorers for a recoverability-like target, while we analyze existing scorers on fixed candidates and decompose their selection gains.

Combining process rewards with sampling. Lightman et al. (2023) find that PRM best-of-N beats majority voting on MATH. Math-Shepherd sums PRM scores within groups of solutions that share a final answer, which helps on MATH but hurts on GSM8K (Wang et al., 2024). Snell et al. (2024) select the answer whose solutions have the largest total PRM score. In these combinations the PRM weighs every candidate. We study the lexicographic case, in which sampling ranks the branches and the PRM only breaks their ties, as the sampling budget varies.

Evaluating scores by their induced decisions. Reward-model accuracy can correlate weakly with optimized-policy performance (Wen et al., 2025; Razin et al., 2026; Liu et al., 2025). Decision-focused learning trains predictors for downstream decisions (Wilder et al., 2019; Elmachtoub & Grigas, 2022; Mandi et al., 2024). Cinquin et al. (2025) find that PRMs approximate state values poorly and that PRM-guided tree search does not significantly beat best-of-N. Their diagnosis concerns the PRM's learned approximation, and ours shows with official labels that even perfect validity leaves a gap.

8 Limitations

MC value selection requires reference answers. Consensus value can favor a common wrong answer. The search uses one rollout policy with 90 to 200 questions per benchmark. OpenR1 and code use outcome-blind judges, so validity there is judge-relative. We study the PRM as a tie-breaker, the small-weight limit of adding its score to the sampled one (Appendix F), and leave larger weights untested.

9 Conclusion

Even official step labels lose to a few sampled continuations as a branch selector, because they cannot rank branches that share a label, and Qwen-PRM's tie-breaking on ProcessBench is worth about one continuation per branch at small budgets. Process scores should be judged by what they add beyond sampled continuations.

AI USE DISCLOSURE

LLMs assisted with literature search, research critique, derivations, code for experiments and figures, data processing, reporting, and manuscript drafting. The authors re-derived adopted suggestions, reran scripts on the recorded evidence, and take full responsibility. All figures are code-generated. Model outputs and outcome-blind judges serve as protocol components, as described in Section 3.

REPRODUCIBILITY STATEMENT

The supplementary material's `EVIDENCE.md` maps claims to compact arrays and recomputation scripts, Appendices G and D describe the fixed-candidate sets and the search protocol, and Appendix J states the scope of the evidence.

REFERENCES

Tristan Cinquin, Geoff Pleiss, and Agustinus Kristiadi. Limits of PRM-Guided Tree Search for Mathematical Reasoning with LLMs, 2025. URL https://arxiv.org/abs/2510.20272.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Yuyang Ding, Xinyu Shi, Juntao Li, Xiaobo Liang, Zhaopeng Tu, and Min Zhang. SCAN: Self-Denoising Monte Carlo Annotation for Robust Process Reward Learning, 2025. URL https://arxiv.org/abs/2509.16548.

Yuyang Ding, Chi Zhang, Juntao Li, Haibin Lin, and Min Zhang. FAPO: Flawed-Aware Policy Optimization for Efficient and Reliable Reasoning, 2026. URL https://arxiv.org/abs/2510.22543.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 2022. URL https://arxiv.org/abs/1710.08005.

Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu, Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 Open Series, 2024. URL https://doi.org/10.5281/zenodo.16998085. Skywork-o1-Open-PRM-Qwen-2.5-1.5B.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1, January 2025. URL https://github.com/huggingface/open-r1. Dataset: OpenR1-Math-220k, https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code, 2024. URL https://arxiv.org/abs/2403.07974.

Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Do We Need to Verify Step by Step? Rethinking Process Supervision from a Theoretical Perspective, 2025. URL https://arxiv.org/abs/2502.10581.

Hynek Kydlíček. Math-Verify: A robust mathematical expression evaluation system, 2025. URL https://github.com/huggingface/math-verify. Version 0.9.0.

Wendi Li and Yixuan Li. Process Reward Model with Q-Value Rankings, 2025. URL https://arxiv.org/abs/2410.11287.

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, 2023. URL https://arxiv.org/abs/2305.20050.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation, 2023. URL https://arxiv.org/abs/2305.01210.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. RM-Bench: Benchmarking Reward Models of Language Models with Subtlety and Style. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.16184.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve Mathematical Reasoning in Language Models by Automated Process Supervision, 2024. URL https://arxiv.org/abs/2406.06592.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shawn Gavin, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, David Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tyshawn Hsing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Tianyang Pang, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. SuperGPQA: Scaling LLM Evaluation across 285 Graduate Disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities. *Journal of Artificial Intelligence Research*, 2024. URL https://arxiv.org/abs/2307.13565.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card, 2025. URL https://arxiv.org/abs/2508.10925.

Shrey Pandit, Austin Xu, Xuan-Phi Nguyen, Yifei Ming, Caiming Xiong, and Shafiq Joty. Hard2Verify: A Step-Level Verification Benchmark for Open-Ended Frontier Math, 2025. URL https://arxiv.org/abs/2510.13744.

Qwen Team. Qwen2.5-Math-7B-PRM800K, 2025. URL https://huggingface.co/Qwen/Qwen2.5-Math-7B-PRM800K. Fine-tuned on human step annotations from PRM800K only.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D. Lee, and Sanjeev Arora. What Makes a Reward Model a Good Teacher? An Optimization Perspective, 2026. URL https://arxiv.org/abs/2503.15477.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding Progress: Scaling Automated Process Verifiers for LLM Reasoning, 2024. URL https://arxiv.org/abs/2410.08146.

Zikang Shan, Han Zhong, Liwei Wang, and Li Zhao. Bringing Value Models Back: Generative Critics for Value Modeling in LLM Reinforcement Learning, 2026. URL https://arxiv.org/abs/2604.10701.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Kaiwen Wang, Jin Peng Zhou, Jonathan Chang, Zhaolin Gao, Nathan Kallus, Kianté Brantley, and Wen Sun. Value-Guided Search for Efficient Chain-of-Thought Reasoning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.17373.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations, 2024. URL https://arxiv.org/abs/2312.08935.

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, 2023. URL https://arxiv.org/abs/2203.11171.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, Xing Yu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. Rethinking Reward Model Evaluation: Are We Barking up the Wrong Tree? In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05584.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. URL https://arxiv.org/abs/1809.05504.

Bin Xie, Bingbing Xu, Xueyun Tian, Yilin Chen, and Huawei Shen. Towards Robust Process Reward Modeling via Noise-aware Learning, 2026. URL https://arxiv.org/abs/2601.12748.

Wei Xiong, Hanning Zhang, Nan Jiang, and Tong Zhang. An Implementation of Generative PRM. GitHub repository, 2024. URL https://github.com/RLHFlow/RLHF-Reward-Modeling/tree/main/math-rm.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, 2023. URL https://arxiv.org/abs/2305.10601.

Fei Yu, Anningzhe Gao, and Benyou Wang. OVM, Outcome-supervised Value Models for Planning in Mathematical Reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 858–875. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.55. URL https://aclanthology.org/2024.findings-naacl.55/.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The Lessons of Developing Process Reward Models in Mathematical Reasoning, 2025. URL https://arxiv.org/abs/2501.07301.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying Process Errors in Mathematical Reasoning, 2025. URL https://arxiv.org/abs/2412.06559.

Yuhang Zhou, Yixin Cao, and Guangnan Ye. From Correctness to Utility: Gain-Based Prefix Evaluation for LLM Reasoning, 2026. URL https://arxiv.org/abs/2606.07190.

A Label-Ordering Sensitivity

PRM800K rates each step -1, 0, or $+1$. Table 5 compares three ways of turning these ratings into the validity ordering that validity selection reads.

Table 5: PRM800K label-ordering sensitivity (%, both directions averaged, 20,000 problem-cluster resamples). Binary validity ranks $\{0, +1\}$ above -1. Ordinal ranks $+1 > 0 > -1$. Positive-only ranks $+1$ above $\{0, -1\}$. Official $+1$ is the rate at which the selected candidate carries rating $+1$.

Selector	Final ↑	Label ↑	Official $+1$ ↑
Uniform	83.35	47.23	24.87
MC value selection	85.96	48.78	25.85
Validity selection (binary)	84.25	70.31	39.98
Validity selection (ordinal)	84.24	70.31	51.26
Validity selection (positive only)	84.13	64.48	51.26

Validity selection reads official ratings, and MC value selection reads reference answers.

MC value selection's final-correctness advantage is $+1.71$, $+1.72$, and $+1.83$ points against the binary, ordinal, and positive-only rules. Resolving all three rating levels therefore leaves the central final-correctness gap intact.

B Proofs and Verification for the Propositions

Proposition 1 supports the claim of the introduction and Section 5 that most of the gap between validity selection and MC value selection comes from branches that share a label. Proposition B.1 supports the claim that the validity filter keeps label satisfaction at almost no cost in final correctness. This appendix proves both, reports the gain split of Proposition 1 on both official-label evaluation sets, and verifies Proposition B.1 against the per-candidate data of the fixed-candidate sets. The verification scripts in the supplementary material record input hashes.

Proof of Proposition 1. Write $V_j = \bar{V}_{C_j} + (V_j - \bar{V}_{C_j})$ and expand $(w - u)^\top V$. The group-mean part is $\sum_j (w_j - \frac{1}{K})\bar{V}_{C_j} = (W_1 - \frac{n_1}{K})\bar{V}_1 + \big((1 - W_1) - \frac{n_0}{K}\big)\bar{V}_0 = (W_1 - \frac{n_1}{K})(\bar{V}_1 - \bar{V}_0)$, using $n_0 = K - n_1$ and $\sum_j w_j = 1$. The remainder is the within-group gain. For the validity selector $w_j = \mathbf{1}\{C_j = 1\}/n_1$ on mixed states, so $W_1 = 1$ and $w_j - \frac{1}{K}$ is constant within each group, making the within-group gain a sum of deviations from a group mean, which is zero.

Holding w fixed, adding η to every invalid-group recoverability shifts $\bar{V}_0$ by η and leaves all within-group deviations unchanged, so only the between-group gain moves, by $-(W_1 - \frac{n_1}{K})\eta$. For the validity selector on mixed states, $W_1 = 1$, giving $-\frac{n_0}{K}\eta$. The shift is restricted to values that remain in $[0, 1]$. This is an identity in V for fixed labels and selection weights, so it holds under any joint law with these quantities fixed. □

Gain split on the official-label evaluation sets. Table 6 applies Equation 3 to every state of both evaluation sets under Qwen3 rollouts, averaged over states and both directions. MC value selection gains 2.61 points over uniform on PRM800K and 7.40 on ProcessBench, and validity selection gains 0.90 and 4.74 points, entirely between groups. Qwen-PRM gains 2.31 points on PRM800K, and the human-label PRM gains 1.73 and 2.94 points. Of MC value selection's within-group gain, 1.79 of 2.16 points come from invalid candidates on PRM800K, and 1.57 of 2.69 points come from valid candidates on ProcessBench. The gap between MC value selection and validity selection equals MC value selection's within-group gain in single-label states plus its within-group gain in mixed-label

states, minus the amount by which validity selection's between-group gain exceeds its own. These terms are 1.43, 0.73, and 0.45 of 1.71 points on PRM800K and 1.96, 0.73, and 0.04 of 2.65 points on ProcessBench. On PRM800K, all candidates have the same label in 52.66% of states, 23% with only valid candidates and 30% with only invalid candidates, and MC value selection gains 0.62 points over uniform in the former and 4.32 in the latter.

Table 6: Gain over uniform selection (points) split by Proposition 1 into the between-group and within-group terms, on PRM800K (357 states) and ProcessBench (322 states) under Qwen3 rollouts. Each selector's two terms add to its gain over uniform up to rounding. The human-label PRM's ProcessBench row comes from an earlier three-decimal readout.

	PRM800K		ProcessBench	
Selector	Between	Within	Between	Within
MC value selection	$+0.445$	$+2.160$	$+4.704$	$+2.695$
Validity selection	$+0.899$	0	$+4.744$	0
Qwen-PRM	$+1.042$	$+1.263$	$+3.880$	-0.015
Human-label PRM	$+0.721$	$+1.009$	$+2.803$	$+0.141$

Proposition B.1 (Cost identity for the validity filter)**.** *Fix a state, recoverabilities V, binary labels C with valid set $\mathcal{V} = \{j : C_j = 1\}$, and a ranking score with top set $\mathcal{T}$. Write $V(R)$ for the expected recoverability of the candidate that rule R selects and $\bar{V}_A$ for the mean of V over a set A. The unconstrained rule U selects uniformly on $\mathcal{T}$. The filter F selects uniformly on the set $\mathcal{F}$ of valid candidates with the highest score when $\mathcal{V} \neq \emptyset$, and uniformly on $\mathcal{T}$ otherwise. The validity tie-breaker B selects uniformly on $\mathcal{T} \cap \mathcal{V}$ when this set is nonempty, and uniformly on $\mathcal{T}$ otherwise. Then*

$$V(U) - V(F) = \mathbf{1}\{\mathcal{T} \cap \mathcal{V} = \emptyset,\ \mathcal{V} \neq \emptyset\}\big(\bar{V}_{\mathcal{T}} - \bar{V}_{\mathcal{F}}\big) - \big[V(B) - V(U)\big].$$

In expectation over states and source draws, the first term is the exclusion probability times its conditional recoverability gap. The second is the gain of validity tie-breaking over uniform ties, which vanishes when the top set is a single candidate. The label satisfaction of F equals that of validity selection.

Proof of Proposition B.1. If $\mathcal{V} = \emptyset$, all three rules select uniformly on $\mathcal{T}$. If $\mathcal{T} \cap \mathcal{V} \neq \emptyset$, the highest score among valid candidates is the overall maximum, so $\mathcal{F} = \mathcal{T} \cap \mathcal{V}$ and $F = B$. Otherwise B selects on $\mathcal{T}$, which contains no valid candidate, and F selects on $\mathcal{F} \subseteq \mathcal{V}$. Hence $V(B) - V(F)$ equals the first term, and $V(U) - V(F) = [V(B) - V(F)] - [V(B) - V(U)]$. F selects a valid candidate whenever one exists, as validity selection does. □

The identity links the filter of Section 5 to the tie-breakers of Section 6. Outside the exclusion event the filter is validity tie-breaking on the top set, so whenever that tie-breaking does not hurt, the cost is at most the exclusion term.

Verification of Proposition B.1. Rebuilding the filtered selector from the per-candidate labels, source rollout counts, and evaluation-rollout rates reproduces the validity-filter and MC value rows of Table 2. The rebuilt cost is 0.009 points on PRM800K and 0.217 on ProcessBench. On ProcessBench the exclusion event has probability 4.97% and conditional gap 7.62 points, which contributes 0.378 points, and the tie-breaking term contributes -0.162, the negative of the official-label gain at $m = 8$ in Table 3. On PRM800K every term is at most 0.012 points.

C Finite Rollout Ties and the Selection Target

Candidate i has recoverability V_i and validity label C_i, and the pairs (V_i, C_i) are drawn independently from a fixed law. The m source outcomes of candidate i are independent Bernoulli(V_i) draws with success count X_i. MC value selection maximizes X_i, and an auxiliary score is consulted only among the candidates tied for the highest count. Because X_i takes only $m + 1$ values, ties are common. The example below shows that a perfect validity checker used on these ties can lower final correctness even when valid candidates are more recoverable overall. It supports the claim of the abstract, the introduction, and Section 6 that a perfect checker consulted only on the ties left

by sampling can lower final correctness even when its valid branches are more recoverable overall. Appendix F gives the accounting identities for these ties.

Example C.1 (Validity helps overall but hurts among tied candidates)**.** *For an exact synthetic example, give each candidate 4 source rollouts and sample*

$$(V, C) = \begin{cases} (0.6, 1) & \textit{with probability } 0.5 \\ (0.1, 0) & \textit{with probability } 0.45 \\ (0.9, 0) & \textit{with probability } 0.05. \end{cases}$$

Valid candidates have higher mean recoverability in the full population (0.60 versus 0.18), but lower mean recoverability among candidates whose source rollouts all succeed (0.60 versus 0.90). With four candidates, validity selection alone improves expected final correctness from 39.00% to 57.38%, and using validity on MC value ties raises it from 60.79% to 61.19%. As the candidate pool grows, MC value selection with uniform ties approaches 70.06%, while MC value selection with validity on its ties approaches 60.00%. The same auxiliary score, useful on its own, changes its contribution as the mix of candidates that reach a tie changes. The two large-pool limits hold because some valid candidate succeeds on all four source rollouts with probability tending to one. Validity tie-breaking then selects uniformly among such candidates, whose mean recoverability is $\mathbb{E}[CV^5]/\mathbb{E}[CV^4]$*, while MC value selection with uniform ties selects among all candidates with four successes, whose mean recoverability is* $\mathbb{E}[V^5]/\mathbb{E}[V^4]$*. Validity tie-breaking helps with two to four candidates and hurts with five or more, by 3.94 points at sixteen candidates (Figure 4).*

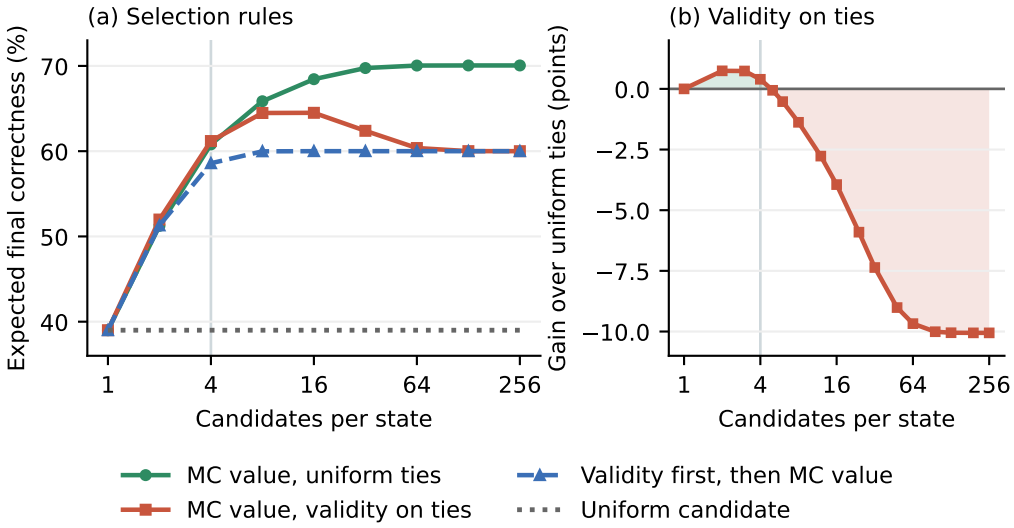

Figure 4: Exact values for Example C.1, in which valid candidates are more recoverable overall. Validity labels are perfectly observed, and each candidate has four source rollouts. (a) As the pool grows, validity on MC value ties approaches validity first, then MC value, at 60.00%, while MC value with uniform ties approaches 70.06%. (b) Validity tie-breaking gains up to 0.74 points with two to four candidates and loses from five candidates on.

D Original-Token Sequential Selection Protocol

We evaluate 200 MATH questions, all 90 AIME 2024–2026 questions, and all 198 GPQA Diamond questions (Rein et al., 2023). Each method continues from its own previously selected history for four rounds. Each round retains the selected candidate extension, and end-of-sequence histories remain terminal. Source rollouts estimate the value of each candidate, and evaluation rollouts measure final correctness after selection. We use four candidates per round and m source rollouts per candidate.

Random branch chooses among candidates uniformly. Scorer-guided search selects the candidate with the highest auxiliary score, which is the Qwen-PRM score on MATH and AIME and the Qwen3-14B judge score on GPQA Diamond. The MC value selector ranks candidates by their mean source

correctness, using $m = 1$ or $m = 4$ rollouts per candidate. Both configurations consult Qwen-PRM only among candidates tied for the highest source score, then choose uniformly among any remaining ties. Qwen-PRM is Qwen2.5-Math-PRM-7B with product aggregation over scored steps. We evaluate every selection rule with four evaluation rollouts at rounds 1, 2, and 4. Consensus value uses source answer frequencies to choose a candidate extension. When several answers share the highest frequency, it averages the matching fractions over them. Voting of direct generation instead returns the most frequent generated final answer.

Table 7: MC-value-selection diagnostic with reference-answer access. After four rounds with Qwen3-4B-Instruct-2507, Single and Voting use the same final-answer rules as Table 4. Values are accuracy (%).

Evaluation set	Single	Voting
MATH (200)	83.88	95.17
AIME 2024–2026 (90)	44.17	55.19

MC value selection ranks candidates by reference-answer correctness over four source rollouts per candidate. Qwen-PRM breaks source-score ties, with uniform choice among remaining ties. AIME averages all 90 questions equally.

The original-token experiments use temperature 0.7, top-p 0.98, and top-k 20. We first average the evaluation outcomes within each problem, then average problems equally. Direct generation uses uninterrupted completions with the same sampling settings and per-trajectory token limit. We use all 64 existing independent completions per problem to estimate Single of direct generation. To estimate Voting of direct generation, we average the vote over all four-element subsets of this bank, with uniform choice among tied most-frequent answers. The branch-selection rows instead vote over four continuations of one selected prefix, so their answers share that prefix. Both columns apply the same four-answer vote but sample the four answers differently, and the direct-generation estimate is not self-consistency over 64 answers. These are repeated measurements of the same sampling and voting rules.

We compute 95% confidence intervals by paired bootstrap resampling over problems. We resample problems jointly across methods using 20,000 bootstrap samples and take the percentile endpoints.

AIME by year. Table 8 reports each year's 30 questions. The main table averages all 90 questions equally, which also gives each year equal weight. Both AIME I and II are included in every year.

Table 8: AIME results by year (30 questions each), using the same four-round search and output rules as Table 4. MC value rows read reference answers, and the other rows do not.

		AIME 2024		AIME 2025		AIME 2026	
Selection rule	Tie-breaker	Single	Voting	Single	Voting	Single	Voting
Direct generation	None	43.23	60.72	34.69	48.22	38.28	53.33
Random branch	None	46.67	68.33	30.00	38.33	37.50	46.11
Scorer-guided search	Uniform	43.33	68.33	30.83	42.78	34.17	49.44
Consensus value	Uniform	52.50	71.11	32.50	46.11	40.83	51.11
Consensus value	PRM	52.50	68.89	37.50	51.67	45.00	59.44
Diagnostic: reads reference answers							
MC value selection ($m = 1$)	PRM	50.83	67.78	33.33	44.44	47.50	59.44
MC value selection ($m = 4$)	PRM	50.00	61.67	40.00	50.00	42.50	53.89

Complete-answer PRM baselines. On MATH and AIME 2024–2025, we score the same direct-generation completions with RLHFlow/Llama3.1-8B-PRM-Deepseek-Data (Xiong et al., 2024). This scorer is separate from the sequential Qwen-PRM. We follow the released double-newline step split and plus/minus chat scoring, retaining every step without truncation. Table 9 reports the released mean aggregation and the minimum, product, and last-step alternatives studied by Zhang et al. (2025). Products are ranked in log space. For $N \in \{4, 8\}$, we partition the same completion bank into fixed disjoint groups of N and average selection correctness across groups, then

problems. Best-of-N chooses the first maximum in the original candidate order. All rules use the original prediction-only answer extraction and correctness labels, and they share the step scores and candidate bank. With four completions, the best aggregation reaches 89.44% on MATH, below four-answer voting over direct generation (93.16%).

Table 9: RLHFlow PRM complete-answer best-of-N. Mean is the released implementation's aggregation. Minimum, product, and last-step aggregation use the same stored step scores. Values are final correctness (%).

Step aggregation	N	MATH	AIME 2024	AIME 2025
Mean	4	78.16	40.42	31.25
Mean	8	74.75	42.50	29.58
Minimum	4	86.22	43.54	34.58
Minimum	8	87.38	47.92	37.08
Product	4	88.09	51.25	39.38
Product	8	89.44	54.17	40.83
Last step	4	89.44	51.25	40.83
Last step	8	90.50	56.25	42.92

Tie-breaking across successive selection points. We replace PRM tie-breaking with uniform choice among candidates sharing the highest consensus value and repeat all four selection rounds. Table 10 reports the component contrast of Section 6, consensus value with PRM tie-breaking minus consensus value with uniform ties at round four, for Single and Voting. Both rules use the same search protocol but can visit different prefixes.

Table 10: Effect of PRM tie-breaking in complete four-round consensus-value search. Values are accuracy changes (percentage points) from uniform tie-breaking, with 95% paired question-bootstrap intervals (20,000 draws). The contrast matches the component gains reported in Section 6.

Evaluation set	Δ Single [95% CI]	Δ Voting [95% CI]
MATH (200)	-0.12 $[-3.25, +2.88]$	$+1.83$ $[-0.67, +4.50]$
AIME 2024–2026 (90)	$+3.06$ $[-0.83, +7.22]$	$+3.89$ $[-1.85, +10.00]$
AIME 2024 (30)	$+0.00$ $[-6.67, +6.67]$	-2.22 $[-12.22, +7.78]$
AIME 2025 (30)	$+5.00$ $[-0.83, +12.50]$	$+5.56$ $[-1.67, +15.00]$
AIME 2026 (30)	$+4.17$ $[-1.67, +11.67]$	$+8.33$ $[-2.22, +20.56]$

GPQA Diamond. This experiment is separate from the 64-question SuperGPQA fixed-candidate diagnostic. We use the official 198-question Diamond set, preserving the question and option text and deterministically shuffling the four options. The generator, sampling settings, four 128-token candidate extensions, four source rollouts, and total 8,192-token trajectory budget match the mathematical search. Answer extraction uses the last completed boxed single letter A–D for both consensus value and exact-match grading.

Qwen3-14B supplies a 0–100 step-validity score from the problem and generated prefix, without reference answers or rollout outcomes. We compare random selection, scorer-guided search, consensus value with uniform ties, and consensus value with judge ties. All four use independent evaluation rollouts. All questions pass the same original-token continuation checks. For direct generation, the first four completions alone give 54.92% Single and 61.45% Voting, against 54.56% and 60.34% from all 64 completions and all four-answer subsets.

E Tie-Breaking Budget Scans

E.1 Fixed-candidate sets with official labels

We reuse the fixed candidates, labels, and both source/evaluation banks of Section 5. A candidate's *source bank* is the set of rollouts available to a selector for that candidate. For each budget m, every candidate's source rollouts are drawn without replacement from its observed source bank, and

selection is averaged exactly over these draws. No continuation is generated or regraded. MC value is the primary score and reads reference answers. The auxiliary score is consulted only on MC value ties, and remaining ties are resolved uniformly. Skywork-PRM uses its native mean aggregation, and the VGS value model is the DeepSeek-VM-1.5B outcome value model of value-guided search (Wang et al., 2025). Intervals are pointwise and resample problems with 20,000 draws (Table 11). With one source rollout, MC value selection with uniform ties reaches 84.85% on PRM800K and 79.44% on ProcessBench (Table 14), and Qwen-PRM on its ties raises these values to 86.20% and 80.42%.

Table 11: Gain in final correctness (points) from breaking MC value ties with each scorer, with 95% problem-cluster intervals. *Within ties* is the gain on tied top sets, so each overall gain equals the tied share times it. For official labels, the *mixed share* is the percentage of tied sets that carry both labels and the *weighted gap* is $\mathbb{E}[\frac{n_0}{t}(\bar{V}_1 - \bar{V}_0) \mid \text{mixed tie}]$ in points, so the within-tie gain equals their product (Proposition F.1). For PRM800K, $m = 8$ extends each source bank from four to eight rollouts per candidate and keeps the original four-rollout evaluation banks.

ProcessBench	$m = 1$	$m = 2$	$m = 4$	$m = 8$
Tied (%)	80.4	72.4	63.3	53.4
Official labels	$+1.31$ $[+0.88, +1.75]$	$+0.73$ $[+0.44, +1.03]$	$+0.37$ $[+0.18, +0.57]$	$+0.16$ $[+0.01, +0.33]$
within ties	$+1.63$	$+1.01$	$+0.59$	$+0.30$
mixed share (%)	39.0	37.1	34.6	31.7
weighted gap	$+4.18$	$+2.73$	$+1.70$	$+0.96$
Qwen-PRM	$+0.97$ $[+0.35, +1.61]$	$+0.53$ $[+0.09, +0.97]$	$+0.24$ $[-0.08, +0.55]$	-0.05 $[-0.36, +0.24]$
Skywork-PRM	$+0.93$ $[+0.29, +1.55]$	$+0.66$ $[+0.22, +1.10]$	$+0.48$ $[+0.18, +0.79]$	$+0.22$ $[-0.09, +0.53]$
VGS value model	$+0.82$ $[+0.17, +1.47]$	$+0.71$ $[+0.27, +1.16]$	$+0.54$ $[+0.23, +0.85]$	$+0.46$ $[+0.16, +0.77]$
PRM800K	$m = 1$	$m = 2$	$m = 4$	$m = 8$
Tied (%)	96.5	92.4	86.0	75.4
Official labels	$+0.22$ $[-0.36, +0.78]$	$+0.03$ $[-0.40, +0.44]$	-0.01 $[-0.40, +0.34]$	-0.13 $[-0.39, +0.13]$
within ties	$+0.23$	$+0.04$	-0.01	-0.17
mixed share (%)	43.4	41.0	38.8	36.6
weighted gap	$+0.53$	$+0.09$	-0.04	-0.47
Qwen-PRM	$+1.35$ $[+0.39, +2.29]$	$+0.98$ $[+0.20, +1.75]$	$+0.81$ $[-0.08, +1.66]$	$+0.05$ $[-0.47, +0.57]$
Skywork-PRM	-0.38 $[-1.32, +0.52]$	-0.08 $[-0.87, +0.66]$	$+0.11$ $[-0.69, +0.89]$	-0.19 $[-0.78, +0.43]$

E.2 Fresh AIME traces

The fresh traces cover all 90 AIME 2024–2026 questions. For each question we sample one new trajectory from Qwen3-4B-Instruct-2507 with the sampling settings of the search and take selection points after 0, 128, 256, and 384 generated tokens. At each of these 360 points we sample 16 candidate extensions of up to 128 tokens and, for every candidate, 16 source and 8 independent evaluation rollouts within the 8,192-token total budget. For $K < 16$ or $m < 16$, candidate and source subsets are averaged over 2,048 shared random draws per point. These candidates have no official labels, so the auxiliary scores are Qwen-PRM, the human-label PRM, and PUM. Single is the mean correctness of the chosen candidate's eight evaluation rollouts. Intervals resample questions with 20,000 draws, keeping each question's four points together (Table 12).

With MC value as the primary score, the Qwen-PRM endpoints have 95% intervals $[-0.30, +1.43]$ at $m = 1$ and $[-1.16, +0.29]$ at $m = 16$, and its change within ties moves from $+0.61$ to -0.83 points. With consensus value they are $[-0.23, +1.57]$ and $[-1.01, +0.34]$. Scoring only the last complete answer box in both source and evaluation rollouts gives endpoints of $+0.63$ and 0.00 with MC value and $+0.74$ and $+0.06$ with consensus value. At $K = 4$ and $m = 4$, the setting of the complete search, Qwen-PRM tie-breaking changes Single by $+0.17$ ($[+0.02, +0.31]$) with MC value and by $+0.16$ ($[+0.02, +0.30]$) with consensus value. The corresponding Voting changes are $+0.18$ ($[-0.02, +0.39]$) and $+0.16$ ($[-0.02, +0.33]$).

E.3 Doubling the source rollouts on fresh AIME traces

The 32-rollout extension of the fresh AIME traces covers all 90 questions, all four selection depths, and 360 selection points. Each candidate has 32 source and 16 evaluation rollouts. The 16-rollout

Table 12: Change in Single (percentage points) from auxiliary tie-breaking on fresh AIME 2024–2026 traces with $K = 16$ candidates, by source rollouts m per candidate. All four selection depths are pooled.

Primary score	Auxiliary score	$m = 1$	$m = 2$	$m = 4$	$m = 8$	$m = 16$
MC value	Tied (%)	93.6	87.8	79.7	68.2	52.8
	Qwen-PRM	+0.57	+0.35	+0.10	+0.04	−0.44
	Human-label PRM	+0.03	0.00	−0.16	−0.24	−0.52
	PUM	+0.23	+0.18	+0.15	+0.12	+0.07
Consensus value	Tied (%)	97.1	84.9	73.2	59.7	39.4
	Qwen-PRM	+0.66	+0.37	+0.12	+0.05	−0.35
	Human-label PRM	+0.12	+0.05	−0.15	−0.23	−0.54
	PUM	+0.22	+0.19	+0.11	+0.06	−0.02

condition averages 2,048 random 16-subsets of the 32 source rollouts per point, and both conditions share the same 16 evaluation rollouts, so the comparison changes only the sampling budget of the primary score. Doubling the source rollouts raises Single by 1.14 points with MC value ($[+0.66, +1.64]$) and by 1.00 with consensus value ($[+0.57, +1.46]$), and the tied share falls by 10.6 and 12.3 points. Every auxiliary gain stays at most 0.2 points from zero (Table 13). With these sixteen evaluation rollouts, the $m = 16$ endpoints of Qwen-PRM are -0.09 with MC value and -0.11 with consensus value, against -0.44 and -0.35 with the eight evaluation rollouts of Table 12.

Table 13: Single (%) of the primary score and change in Single (percentage points) from auxiliary tie-breaking at 16 and 32 source rollouts per candidate, on the 90 fresh AIME questions with $K = 16$ and a shared bank of 16 evaluation rollouts.

Primary score	m	Primary	Tied (%)	Qwen-PRM	Human-label PRM	PUM
MC value	16	43.55	55.0	−0.09	−0.11	0.00
	32	44.69	44.4	+0.13	+0.02	+0.04
Consensus value	16	42.97	44.5	−0.11	−0.10	0.00
	32	43.98	32.2	+0.20	+0.09	+0.04

F TIES, THE SOURCE BUDGET, AND WHAT AN AUXILIARY SCORE CAN ADD

Section 6 varies two things, the number m of source rollouts per candidate and the auxiliary score consulted on ties. This appendix gives exact accounting identities for both and verifies them on the recorded per-candidate arrays. Proposition F.1 is the factorization that Section 6 uses to split the official-label gain into the share of tied sets with both labels and the recoverability gap inside them. Proposition F.2 proves the claim of the abstract and the introduction that one more reference-graded source rollout per candidate is worth exactly its expected gain on the current ties. Proposition F.4 bounds both gains by a term that falls exponentially as the source budget grows, and Corollary F.5 carries the analysis to consensus value, the primary score of the search. Throughout, a state has K candidates with values V_j and labels $C_j \in \{0, 1\}$. In the propositions V_j is the recoverability of Section 2. In the verification it is the evaluation-bank rate. Source counts X_j determine the tied top set $\mathcal{T} = \arg\max_j X_j$ with $t = |\mathcal{T}|$, and $\bar{V}_A$ denotes the mean of V over a set A. MC value selection with uniform ties selects uniformly on $\mathcal{T}$ and has value $\bar{V}_{\mathcal{T}}$.

F.1 THE GAIN OF A TIE-BREAKER IS A DECOMPOSITION ON THE TIED SET

Proposition F.1 (Tie-breaking gain on the tied set)**.** *Fix a state and a realized tied top set $\mathcal{T}$ of any primary score. Let a tie rule put weights $w_j \geq 0$ with $\sum_{j\in\mathcal{T}} w_j = 1$ on $\mathcal{T}$. Write $n_1 = \sum_{j\in\mathcal{T}} C_j$, $n_0 = t - n_1$, and $W_1 = \sum_{j\in\mathcal{T},\, C_j=1} w_j$, and let $\bar{V}_1, \bar{V}_0$ be the mean values of the valid and of the invalid members of $\mathcal{T}$, with the mean of an empty group set to zero. Then*

$$\sum_{j\in\mathcal{T}} w_j V_j - \bar{V}_{\mathcal{T}} = \Big(W_1 - \frac{n_1}{t}\Big)(\bar{V}_1 - \bar{V}_0) + \sum_{j\in\mathcal{T}} \Big(w_j - \frac{1}{t}\Big)(V_j - \bar{V}_{C_j}).$$

For the validity tie-breaker, uniform among the valid members of $\mathcal{T}$ when $n_1 \geq 1$ and uniform on $\mathcal{T}$ otherwise, the second term vanishes and the gain equals $\frac{n_0}{t}(\bar{V}_1 - \bar{V}_0)$ when $0 < n_1 < t$ and zero otherwise. Averaged over source draws, the overall gain of any tie rule equals the tie probability times its expected gain conditional on a tie.

Proof. The tied set with its labels and weights is a state with t candidates whose uniform rule is $u_j = 1/t$, and Equation (3) of Proposition 1 applied to this state is the displayed identity. For the validity tie-breaker on a mixed tied set, $w_j = 1/n_1$ on valid members and 0 otherwise, so $W_1 = 1$ and $w_j - 1/t$ is constant within each label group. The second term is then a sum of deviations from group means, which is zero. When $\mathcal{T}$ carries one label only, $W_1 = n_1/t$ and $w_j = 1/t$, so both terms vanish. Outside ties the rule has no effect, which gives the factorization. □

The proposition holds for consensus value as the primary score without change, because only the tied set enters. For a binary label it turns the within-tie gain of Table 11 into two measurable factors, the share of tied sets that carry both labels and the recoverability gap between the valid and the invalid members of those sets. For a continuous score it separates a between-group term, which any label-like ordering can earn, from a within-group term, which only a score that ranks inside label groups can earn.

F.2 One more source rollout is worth exactly its expected tie resolution

Let the source draws of candidate j form an exchangeable sequence $\xi_{j,1}, \xi_{j,2}, \ldots$ of $\{0,1\}$ variables, independent across candidates, and let $X_j^{(m)} = \sum_{i \leq m} \xi_{j,i}$. Independent Bernoulli($V_j$) rollouts and uniform draws without replacement from a finite bank are both of this form. Because the draws are exchangeable within a candidate and independent across candidates, the success probability of a candidate's next draw depends on the history only through its own count, $q_j(x) = \Pr[\xi_{j,m+1} = 1 \mid X_j^{(m)} = x]$ for every x with $\Pr[X_j^{(m)} = x] > 0$. Let $\mathcal{T}_m$ be the tied top set at budget m at common count M, and define the resolution gain

$$R_m(\mathcal{T}) = \sum_{\emptyset \neq A \subseteq \mathcal{T}} \prod_{j \in A} q_j(M) \prod_{j \in \mathcal{T} \setminus A} \big(1 - q_j(M)\big)\, \big(\bar{V}_A - \bar{V}_{\mathcal{T}}\big),$$

the expected change in value when one more draw per tied candidate resolves the tie, with $R_m(\mathcal{T}) = 0$ for a singleton. Write $f(m) = \mathbb{E}[\bar{V}_{\mathcal{T}_m}]$ for the value of MC value selection with uniform ties at budget m.

Proposition F.2 (Value of one more source rollout)**.** *For every value vector $V \in \mathbb{R}^K$ and every $m \geq 0$, with $m < n$ for a finite bank of size n,*

$$f(m+1) - f(m) = \mathbb{E}\big[R_m(\mathcal{T}_m)\big] = \Pr\big[|\mathcal{T}_m| \geq 2\big]\, \mathbb{E}\big[R_m(\mathcal{T}_m) \,\big|\, |\mathcal{T}_m| \geq 2\big].$$

For two candidates with independent Bernoulli(V_j) draws, $R_m(\{1,2\}) = \frac{1}{2}(V_1 - V_2)^2$, so that

$$f(m+1) - f(m) = \tfrac{1}{2}\,(V_1 - V_2)^2\, \Pr\big[X_1^{(m)} = X_2^{(m)}\big].$$

Proof. Write $x = X^{(m)}$, $\xi = (\xi_{j,m+1})_j$, and $y = x + \xi = X^{(m+1)}$. Let $\mathcal{T} = \mathcal{T}(x)$ at count M and $\mathcal{S} = \{j : x_j = M - 1\}$. Then $f(m+1) - f(m) = \mathbb{E}[\bar{V}_{\mathcal{T}(y)} - \bar{V}_{\mathcal{T}(x)}]$. Split the outcomes by whether some tied candidate succeeds, $E_1 = \{\exists j \in \mathcal{T} : \xi_j = 1\}$, or all tied candidates fail, $E_2 = E_1^c$.

On E_1 the maximum of y is $M+1$ and $\mathcal{T}(y) = A := \{j \in \mathcal{T} : \xi_j = 1\}$, since every other candidate has $y_j \leq M$. Conditional on x, the draws ξ_j for $j \in \mathcal{T}$ are independent with success probabilities $q_j(M)$, so $\mathbb{E}[(\bar{V}_{\mathcal{T}(y)} - \bar{V}_{\mathcal{T}})\mathbf{1}_{E_1} \mid x] = R_m(\mathcal{T})$.

On E_2 the maximum of y stays M and $\mathcal{T}(y) = \mathcal{T} \cup B$ with $B = \{j \in \mathcal{S} : \xi_j = 1\}$, so the value changes only when $B \neq \emptyset$. These changes cancel in expectation. Group the outcomes (x, ξ) in $E_2 \cap \{B \neq \emptyset\}$ by the final count vector y, whose top set $\mathcal{T}' = \mathcal{T}(y)$ has size $t' \geq 2$ at count M', which equals M on E_2, and by the draws ξ_j of the candidates outside $\mathcal{T}'$. Within a group the outcome is determined by B, a nonempty proper subset of $\mathcal{T}'$, because the members of $\mathcal{T}(x)$ fail and so lie outside B. Candidates in $\mathcal{T}' \setminus B$ have $x_j = M'$ and $\xi_j = 0$, and candidates in B have $x_j = M' - 1$ and $\xi_j = 1$. Every such B yields a valid outcome of the group, because candidates

outside $\mathcal{T}'$ have $x_j \le y_j \le M' - 1$, so $\mathcal{T}(x) = \mathcal{T}' \setminus B$ and all of $\mathcal{T}(x)$ fails. Exchangeability of the first $m+1$ draws of candidate j gives

$$\Pr\big[X_j^{(m)} = M' - 1,\ \xi_{j,m+1} = 1\big] = \Pr\big[X_j^{(m+1)} = M'\big]\,\frac{M'}{m+1},$$
$$\Pr\big[X_j^{(m)} = M',\ \xi_{j,m+1} = 0\big] = \Pr\big[X_j^{(m+1)} = M'\big]\,\frac{m+1-M'}{m+1},$$

because given M' successes among $m+1$ exchangeable draws the last draw is a success with probability $M'/(m+1)$. The probability of an outcome in the group therefore depends on B only through $|B|$. For fixed $b = |B| \in \{1, \dots, t'-1\}$ the value changes of the group sum to

$$\sum_{|B|=b} \big(\bar{V}_{\mathcal{T}'} - \bar{V}_{\mathcal{T}'\setminus B}\big) = \binom{t'}{b}\bar{V}_{\mathcal{T}'} - \sum_{|A|=t'-b} \bar{V}_A = 0,$$

since each element of $\mathcal{T}'$ lies in the same number of subsets A of a given size, so the average of $\bar{V}_A$ over those subsets is $\bar{V}_{\mathcal{T}'}$. Summing over b, over the outside draws, and over y gives $\mathbb{E}[(\bar{V}_{\mathcal{T}(y)} - \bar{V}_{\mathcal{T}})\mathbf{1}_{E_2}] = 0$, which proves the identity. The factorization follows because R_m vanishes on singletons.

For two candidates with $q_j = V_j$ the three nonempty sets A contribute $V_1(1-V_2)\frac{V_1-V_2}{2} + V_2(1-V_1)\frac{V_2-V_1}{2} + V_1V_2 \cdot 0 = \frac{1}{2}(V_1 - V_2)^2$. □

Corollary F.3 (Resampling only the tied candidates)**.** *Under the conditions of Proposition F.2, draw one more source rollout only for the candidates in $\mathcal{T}_m$ when $|\mathcal{T}_m| \ge 2$, and select uniformly among those that succeed, or uniformly on $\mathcal{T}_m$ if none does. This rule attains $f(m+1)$ with $\mathbb{E}\big[|\mathcal{T}_m|\,\mathbf{1}\{|\mathcal{T}_m| \ge 2\}\big] \le K$ additional rollouts per state.*

Proof. Given the first m draws, the rule changes the value by $\bar{V}_A - \bar{V}_{\mathcal{T}_m}$ when the success set $A \subseteq \mathcal{T}_m$ is nonempty and leaves it unchanged otherwise, so its expected gain is $R_m(\mathcal{T}_m)$, which vanishes when $|\mathcal{T}_m| = 1$. Proposition F.2 gives $f(m) + \mathbb{E}[R_m(\mathcal{T}_m)] = f(m+1)$. □

The rule is itself a tie-breaker that consults one fresh rollout per tied candidate, so a PRM consulted on the same ties competes with this many rollouts rather than with one more rollout for every candidate.

What the identity says. Proposition F.1 and Proposition F.2 place the auxiliary score and further sampling on the same events. An auxiliary score realizes the tie probability times its within-tie gain, and one more source rollout per candidate realizes the tie probability times the expected resolution gain. Outside tied sets neither changes the expected outcome, even though one more rollout can change the selected branch by letting a runner-up catch up, because those changes cancel exactly. The comparison between consulting a PRM and drawing more rollouts therefore reduces to comparing, on the same tied sets, the PRM's within-tie gain with $R_m(\mathcal{T})$. On a two-candidate tie a perfect auxiliary gains $\frac{1}{2}|V_1 - V_2|$ and one more rollout gains $\frac{1}{2}(V_1 - V_2)^2$, a fraction $|V_1 - V_2|$ of it.

Because source counts are integers, ranking by $X_j + \varepsilon S_j$ with $\varepsilon > 0$ and $\varepsilon(\max_j S_j - \min_j S_j) < 1$ selects exactly as MC value selection with S on its ties, so every result in this appendix also covers weighted combinations with a small enough weight on the auxiliary score.

F.3 Both channels vanish as the source budget grows

Proposition F.4 (Auxiliary gain and rollout value under a growing source budget)**.** *Let the rollouts be independent Bernoulli(V_j), let $V^\star = \max_j V_j$ and $\mathcal{M}^\star = \{j : V_j = V^\star\}$, and suppose $\mathcal{M}^\star \ne [K]$ with $\Delta = V^\star - \max_{j \notin \mathcal{M}^\star} V_j > 0$. For any tie rule on MC value ties, the gain G_m over uniform ties and the increment $f(m+1) - f(m)$ of Proposition F.2 satisfy*

$$\max\big(|G_m|,\ |f(m+1) - f(m)|\big) \le \mathbb{E}\Big[\big(\max_{j\in\mathcal{T}_m} V_j - \min_{j\in\mathcal{T}_m} V_j\big)\,\mathbf{1}\{\mathcal{T}_m \not\subseteq \mathcal{M}^\star\}\Big]$$
$$\le (K - |\mathcal{M}^\star|)\,e^{-m\Delta^2/2}.$$

Proof. When $\mathcal{T}_m \subseteq \mathcal{M}^\star$, every tied candidate has value $V^\star$, so any tie rule and any resolution give the same value and both quantities vanish. Otherwise each is bounded by the range of V on $\mathcal{T}_m$, which is at most 1. The event $\mathcal{T}_m \not\subseteq \mathcal{M}^\star$ requires $X_j \geq X_{i^\star}$ for some $j \notin \mathcal{M}^\star$ and a fixed $i^\star \in \mathcal{M}^\star$. The difference $X_j - X_{i^\star}$ is a sum of m independent terms in $[-1, 1]$ with mean at most $-\Delta$, so Hoeffding's inequality gives $\Pr[X_j \geq X_{i^\star}] \leq e^{-m\Delta^2/2}$. A union bound over $j \notin \mathcal{M}^\star$ completes the proof. □

The tie probability itself need not vanish. When $|\mathcal{M}^\star| \geq 2$, the candidates in $\mathcal{M}^\star$ tie at every budget if $V^\star = 1$ and with probability of order $m^{-1/2}$ if $0 < V^\star < 1$. Only the gains carry the exponential bound. With a finite source bank the budget stops at the bank size, the counts become the bank counts, and ties among candidates with equal bank counts persist, so the auxiliary's value converges to its full-bank tie gain rather than to zero. This is the regime of the last column of Table 11.

F.4 Consensus value as the primary score

Propositions F.1 and B.1 hold for consensus value without change, because only the tied set and the labels enter. Proposition F.2 requires the success indicator of each rollout to be a function of that rollout alone. Agreement with the pooled consensus answer $\hat{a}$ depends on all rollouts of the state, so the identity does not apply to consensus value as stated. Table 18 reports the share of states with a wrong consensus answer, between 4.62% and 48.76%.

Corollary F.5 (Consensus value when the consensus answer is correct)**.** *Suppose agreement with $\hat{a}$ and correctness are judged by the same answer equivalence, and let ρ_m be the probability that the consensus answer formed from m source rollouts per candidate is wrong. On the event that it is correct, consensus value and MC value give every candidate the same source count, so under a common tie rule they select the same branch. Write $f^{\mathrm{cons}}(m)$ and $f^{\mathrm{MC}}(m)$ for the values of the two selectors under that rule at budget m. Hence, for any common tie rule,*

$$\left|f^{\mathrm{cons}}(m) - f^{\mathrm{MC}}(m)\right| \leq \rho_m \Big(\max_j V_j - \min_j V_j\Big),$$

and with uniform ties

$$f^{\mathrm{cons}}(m+1) - f^{\mathrm{cons}}(m) \geq \mathbb{E}\big[R_m(\mathcal{T}_m)\big] - \rho_m - \rho_{m+1}.$$

Proof. A source rollout agrees with a correct $\hat{a}$ exactly when it is correct, so on that event the two count vectors coincide, and so do the top sets and the selection distributions under a common tie rule. The two values therefore differ only on the event that $\hat{a}$ is wrong, where each is a convex combination of the V_j and their difference lies between $\min_j V_j - \max_j V_j$ and $\max_j V_j - \min_j V_j$. For the second claim, write $f^{\mathrm{cons}}(m+1) - f^{\mathrm{cons}}(m)$ as $f^{\mathrm{MC}}(m+1) - f^{\mathrm{MC}}(m)$ plus the change in $f^{\mathrm{cons}} - f^{\mathrm{MC}}$, apply Proposition F.2 to the first term, and bound the second by $\rho_m + \rho_{m+1}$ using the first claim and $\max_j V_j - \min_j V_j \leq 1$. □

Averaged over states, both bounds hold with ρ_m replaced by its average. The main text uses the exact coupling on the event that the consensus answer is correct. The original-token search averages consensus value over tied most-frequent answers (Appendix D), so there the coupling holds when the most frequent answer is unique and correct.

F.5 Verification on the recorded arrays

The checks reuse the fixed candidates, official labels, both source/evaluation bank directions, and the hypergeometric averaging of Table 11, which averages over all size-m subsets of the source bank drawn without replacement. PRM800K has 357 states with a source bank of four rollouts per candidate, and ProcessBench has 322 states with a source bank of eight. Values are the evaluation-bank rates, and every quantity is averaged over states and both directions. Proposition F.1 holds at every event on both sets, and Proposition F.2 holds at every state, direction, and budget on both sets, on the fresh AIME traces of Appendices E.2 and E.3, and on the AIME 2026 selection points below, in each case to within 2×10^{-15}.

Proposition F.1. On the official label, the within-tie gain equals the between-group term on the tied set, and Table 11 lists its two factors. On ProcessBench the share of tied sets that carry both labels moves from 39.0% to 31.7% as m grows from 1 to 8, while the label-weighted recoverability gap inside those sets moves from 4.18 to 0.96 points. The decline of the official-label gain in Table 3 is therefore mostly a shrinking gap inside tied sets, with a smaller contribution from fewer mixed ties. On PRM800K the label-weighted gap is 0.53 points at $m = 1$ and -0.04 at $m = 4$. For the continuous scorers on ProcessBench, Qwen-PRM earns a positive between-group term at every budget, 1.35 points at $m = 1$ and 0.34 at $m = 8$, while its within-group term is negative at every budget and equals -0.44 points at $m = 8$, which is why its total is negative there. The VGS value model keeps both terms positive, 0.43 and 0.42 points at $m = 8$, and its within-tie gain stays near 0.85 points at $m = 4$ and $m = 8$, which matches its flat column in Table 11.

Proposition F.2 on the fixed-candidate sets. With values equal to the evaluation-bank rates and next-draw probabilities $(k_j - M)/(n - m)$ from the source bank, Table 14 lists $f(m)$ and its increments. The aggregate increment is positive at every budget on both sets, and the values at the full bank are those of Table 2.

Table 14: Proposition F.2 on the fixed-candidate sets. $f(m)$ is the value of MC value selection with uniform ties in percent, and the increment $\Delta f(m) = f(m{+}1) - f(m)$, which equals $\mathbb{E}[R_m(\mathcal{T}_m)]$, is in percentage points.

Evaluation set		$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
PRM800K	$f(m)$	84.85	85.42	85.74	85.96				
	$\Delta f(m)$	+0.572	+0.316	+0.224					
ProcessBench	$f(m)$	79.44	80.52	81.03	81.34	81.56	81.75	81.92	82.09
	$\Delta f(m)$	+1.077	+0.508	+0.311	+0.226	+0.190	+0.170	+0.171	

Proposition F.2 on the search's own AIME 2026 selection points. We fix the candidates that the complete consensus value + Qwen-PRM search reaches on AIME 2026 at rounds 1, 2, and 4, giving 90 selection points over 30 questions with four candidates each. We add four source rollouts per candidate to the four that the search used, which doubles each source bank from four to eight, and score every choice with eight independent evaluation rollouts per candidate that do not participate in selection. Drawing m source rollouts without replacement from the eight and averaging exactly, MC value selection rises from 41.19% at $m = 1$ to 43.02% at $m = 8$ in Single value under the original answer parsing, while the tied share moves from 84.9% to 52.2%. Qwen-PRM's tie-breaking gain on these MC value ties moves from -0.09 points at $m = 1$ to $+0.31$ at $m = 8$ for the Single value, the mean of its $m = 8$ gains of $+1.60$, -0.66, and 0.00 points in rounds 1, 2, and 4.

Proposition F.4. The bound concerns true recoverabilities, so we report the quantities it controls. The mean range of evaluation rates inside tied sets moves from 21.5 to 14.8 points on PRM800K as m grows from 1 to 4 and from 11.9 to 4.8 points on ProcessBench as m grows from 1 to 8, and the headroom $\max_{\mathcal{T}} V - \bar{V}_{\mathcal{T}}$ moves from 8.5 to 5.9 and from 5.7 to 2.3 points. Between 79% and 97% of tied sets consist of candidates that succeeded on all m source rollouts, which is where the finite bank stops separating candidates.

G Experimental Details

The supplementary material contains compact per-candidate arrays, manifests, hashes, validation scripts, and figure-generation scripts for the central comparisons. Recomputation commands and expected outputs are listed in its `README.md` and `EVIDENCE.md`.

G.1 Fixed-candidate evaluation sets

PRM800K. The official-label evaluation set contains 357 exact-prefix states from 73 held-out MATH problems, with five candidates per state. Candidate construction applies terminal-marker and reference-answer-literal filters and requires complete step boundaries and available ratings. The literal filter applies to answers with at least two characters that do not already appear in the problem.

Table 15 reports the full construction funnel from 1,873 observed states to the 357 retained. Official three-level ratings define step validity. Qwen3 uses four source and four evaluation rollouts per candidate in both directions. The policy-family replication uses eight gpt-oss rollouts per candidate in both directions. In every direction, one set fixes the pick and the other scores its final correctness.

Natural OpenR1. The natural-prefix evaluation set contains one state from each of 405 problems and seven candidates per state. Four Qwen3 rollouts per candidate estimate and evaluate recoverability in both directions. The dual-target evaluation applies an outcome-blind judge to the same exact candidate strings. The judge sees neither rollout outcomes nor scorer values. One candidate judgment remains unparsable, so the judge-label comparison uses 404 problems. The learned Qwen-PRM comparison uses the 300 problems whose candidates fit within its context window. Over all 405 problems, the MC value selector raises final correctness from 46.18% under uniform selection to 47.84%.

ProcessBench. The dual-target evaluation set groups distinct responses by exact problem and treats the first three supplied steps as candidate branches. It retains 322 problem groups and 747 candidates ($K = 2$–4), excludes candidates containing a boxed answer, and uses the official annotation to identify branches whose three steps remain valid. Every candidate receives two independently served sets of eight rollouts under Qwen3 and two under gpt-oss, with source and evaluation roles swapped across directions.

Code and scientific choice. The code evaluation contains 223 independent problems: 63 HumanEval+, 85 MBPP+, and 75 LiveCodeBench. Each has five fixed program prefixes, eight rollouts per candidate in both directions under both rollout policies, and two outcome-blind judge draws. The source draw's confidence in validity ranks candidates, and the opposite draw's binary label evaluates label satisfaction. Construction and screening read neither final outcomes nor validity labels. The SuperGPQA evaluation analogously fixes 64 independent problems and five root-reasoning candidates before collecting eight rollouts per candidate in both directions and two judge draws.

Across all evaluation sets, exact ties are resolved uniformly and both directions are averaged before problem-cluster bootstrap resampling.

Table 15: Exhaustive PRM800K exact-prefix construction. Counts before the final row are nested admission milestones. Exclusion tags overlap and therefore do not sum to the total excluded count.

Construction milestone	States
Observed dataset-row/step states	1,873
Nonempty prefixes	1,767
Boundary-complete prefixes	1,568
At least five admissible candidates	442
All state and candidate criteria	**357**
Excluded by at least one criterion	1,516

Overlapping exclusion tag	Count
Fewer than five eligible candidates	1,431
Incomplete prefix	199
Reference-answer literal in prefix	171
Empty prefix	106
Completed prefix	33
Candidate filters: incomplete boundary	708
Candidate filters: completed prefix	323
Candidate filters: reference literal	288
Candidate filters: missing rating / flagged	110 / 6

G.2 Scientific-choice matched targets

On 64 SuperGPQA problems with 320 fixed candidates, the MC value selector ($m = 8$), evaluated in both directions, improves final correctness over uniform by 18.38 points ($[+12.62, +24.72]$). Validity selection with both judge draws improves label satisfaction by 31.72 points ($[+26.25, +37.19]$) and joint value by 8.37 points ($[+4.98, +11.97]$).

G.3 Full selector rates

Table 16: Selector rates on evaluation rollouts under Qwen3 rollouts (%, point estimate ± problem-cluster bootstrap standard deviation) for five of the selectors in Table 2. MC value selection reads reference answers, and validity selection reads official step labels.

	PRM800K		ProcessBench	
Selector	Final ↑	Label ↑	Final ↑	Label ↑
Uniform	83.35 ± 2.65	47.23 ± 3.26	74.70 ± 1.74	69.80 ± 1.82
Validity selection	84.25 ± 2.75	70.31 ± 3.01	79.44 ± 1.77	90.99 ± 1.59
Validity filter → MC value	85.95 ± 2.63	70.31 ± 3.01	81.88 ± 1.75	90.99 ± 1.59
Qwen-PRM	85.66 ± 2.75	62.98 ± 3.34	78.56 ± 1.89	87.89 ± 1.82
MC value selection	85.96 ± 2.59	48.78 ± 3.41	82.09 ± 1.71	77.82 ± 1.94

H Additional Fixed-Candidate Diagnostics

H.1 Concrete branches and rollout-policy dependence

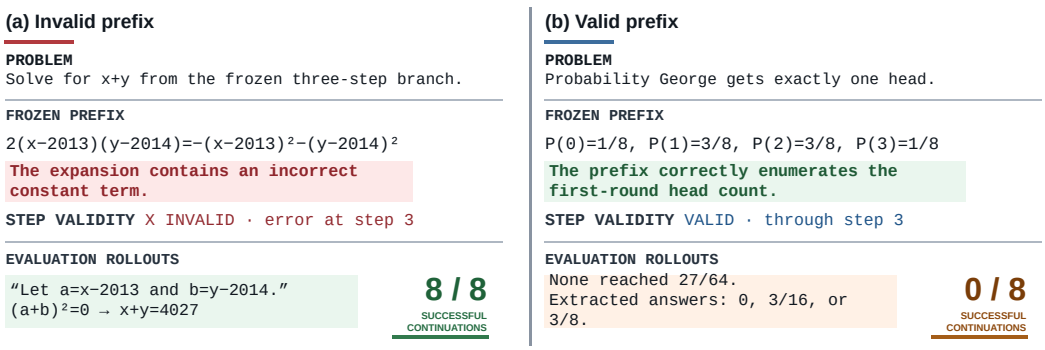

Figure 5: Two ProcessBench examples from different problems. The left three-step prefix contains an annotated error but all eight evaluation rollouts succeed. The right prefix is valid through step three but none of its eight evaluation rollouts succeeds. These illustrate the off-diagonal cells of Figure 3.

Table 17: Six rollout policies on the same PRM800K evaluation set: 357 states from 73 problems. Invalid-branch success averages within the 275 states containing invalid-labeled candidates. Validity-selection gain is the final-correctness difference from uniform over all 357 states. Both source/evaluation directions are averaged.

Rollout policy	Invalid-branch success (%)	Validity-selection gain (points)
Qwen3-4B-Base	13.86	+2.03
R1-Distill-1.5B	49.84	+0.45
Qwen3-4B-Instruct-2507 (main)	79.91	+0.90
Qwen3-32B	82.42	+0.25
gpt-oss-20B	88.17	+0.28
Qwen3-14B	89.18	+0.15

The association is not monotone. The main policy, Qwen3-4B-Instruct-2507, recovers more often than R1-Distill-1.5B yet gains more from labels, and gpt-oss-20B has slightly higher recovery and validity-selection gain than Qwen3-32B. This comparison varies the rollout policy while keeping the candidates and labels fixed. With labels fixed, validity selection's weights are fixed, so by Proposition 1 each policy's validity-selection gain averages $\frac{n_0}{K}(\bar{V}_1 - \bar{V}_0)$ over mixed-label states, counting single-label states as zero. The gain therefore depends on how often valid branches recover as well as invalid ones, so it need not fall monotonically with invalid-branch success.

H.2 Consensus value without a verifier

Consensus value needs no reference answer. It ranks candidates by how often their source rollouts match a shared answer, using answer frequencies (Wang et al., 2023). For a state with candidates

$z_1, \ldots, z_K$ and m source rollouts per candidate, we pool the parsed final answers from all Km source rollouts. We choose a most frequent answer as the consensus answer $\hat{a}$, resolving answer ties uniformly with a fixed seed. Candidate j receives score

$$\widetilde{V}_j = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{\mathrm{ans}(y_{j,i}) = \hat{a}\}. \tag{4}$$

A rollout without a parseable answer casts no vote and counts as disagreement. A candidate with no parseable answer has consensus value zero, and a state in which every candidate has score zero falls back to uniform selection, as any exact tie does. No state is excluded on answer grounds. These fixed-candidate diagnostics use one consensus answer when source answer frequencies tie. The original-token protocol averages consensus value over all tied most-frequent answers (Appendix D).

We evaluate the selected candidates on disjoint rollouts scored by the task verifier, with both rollout directions and 20,000 problem-cluster bootstrap samples. Table 18 reports PRM800K (357 states) and ProcessBench (322 states) under Qwen3, ProcessBench under gpt-oss-20B (322 states), and natural OpenR1 (404 states, one state lacking a parse-valid validity label).

Table 18: Consensus value against the MC value selector (all columns in percent). The first column is consensus value's final-correctness gain over uniform with a 95% problem-cluster interval, and the retained share divides it by the MC value selector's gain. Pseudo-label accuracy is the per-rollout rate at which matching the consensus answer coincides with verifier-based correctness. Wrong consensus is the share of states whose consensus answer is incorrect.

Evaluation set	Consensus value − uniform	Share retained	Pseudo-label acc.	Wrong consensus
PRM800K / Qwen3	$+1.63$ $[+0.72, +2.64]$	62.5	96.32	4.62
ProcessBench / Qwen3	$+5.35$ $[+4.02, +6.73]$	72.2	90.17	14.89
ProcessBench / gpt-oss-20B	$+2.01$ $[+1.23, +2.85]$	95.7	95.19	5.92
OpenR1 / Qwen3	$+0.60$ $[-0.20, +1.41]$	36.2	62.92	48.76

An incorrect consensus answer makes consensus value reward the wrong outcome. More rollouts can estimate the match rate with a fixed consensus answer more precisely while preserving this bias. Pseudo-label accuracy is above 90% on the three official-label evaluation sets and 62.92% on OpenR1, where uniform selection is correct on 46.05% of evaluation rollouts and the consensus answer is wrong in 48.76% of states. The retained share falls accordingly.

I Additional Selector Results

I.1 Annotation-aligned localization

Table 19 scores each public PRM on ProcessBench's first-error localization task.

Table 19: Annotation-aligned ProcessBench localization (%) on its 3,400 traces. Each public PRM uses its released prompt format and a raw-score threshold calibrated on GSM8K, then transferred unchanged. F1 is the harmonic mean of exact first-error accuracy and accuracy on fully correct traces. Skywork-PRM is the 1.5B model. For reference, published ProcessBench F1 is 82.4/77.6/67.5/66.3 for Qwen-PRM (Zhang et al., 2025) and 59.0/48.0/19.3/19.2 for Skywork-PRM-1.5B (Zheng et al., 2025). With a fixed 0.5 cutoff instead of the GSM8K-calibrated one, our saved Qwen-PRM scores give 81.90/77.22/67.42/64.85.

Split	Qwen-PRM first error	Qwen-PRM F1	Skywork-PRM first error	Skywork-PRM F1
GSM8K (calibration)	74.88	**82.65**	52.66	**63.11**
MATH (held out)	70.03	**74.06**	39.56	**49.88**
OlympiadBench (held out)	53.86	59.23	14.83	19.39
Omni-MATH (held out)	56.65	57.56	13.97	19.80

I.2 FULL DUAL-TARGET CONTRASTS

Each row below compares selectors on identical candidates. Final correctness is the MC value selector minus the named comparison selector. Label satisfaction and joint value reverse that subtraction, so positive entries follow the predicted dissociation.

Evaluation set	Comparison selector	Final ↑	Label ↑	Joint ↑
PRM800K (Qwen3)	Official labels	**+1.71 [0.46, 3.20]**	**+21.53 [16.98, 26.43]**	**+17.83 [13.52, 22.28]**
PRM800K (gpt-oss)	Official labels	**+1.04 [0.03, 2.13]**	**+19.27 [14.61, 24.16]**	**+15.65 [10.94, 20.80]**
ProcessBench (Qwen3)	Qwen-PRM	**+3.53 [1.67, 5.61]**	**+10.07 [6.21, 13.95]**	**+4.58 [1.32, 7.79]**
ProcessBench (Qwen3)	Official labels	**+2.65 [1.43, 3.97]**	**+13.17 [9.96, 16.56]**	**+6.90 [4.35, 9.59]**
ProcessBench (gpt-oss)	Official labels	**+0.80 [0.05, 1.60]**	**+17.56 [14.12, 21.09]**	**+11.30 [8.41, 14.35]**
ProcessBench (gpt-oss)	Qwen-PRM	**+1.11 [0.03, 2.39]**	**+14.45 [10.58, 18.33]**	**+9.10 [5.75, 12.40]**
OpenR1 (Qwen3)	Judge labels	**+1.50 [0.17, 2.90]**	**+30.32 [26.42, 34.21]**	**+12.72 [10.01, 15.62]**

Table 20: Dual-target selector contrasts (percentage points, problem-cluster family-wise 95% intervals over the three selection targets within each row). Bold intervals exclude zero. The OpenR1 row uses one outcome-blind judge label per candidate on the 404 problems with parseable judgments.

Effect sizes change with the rollout policy, as expected for policy-dependent recoverability.

I.3 SAME-LABEL SCORE PERMUTATION

Within each candidate set, we permute a scorer's continuous scores among the candidates that share an official binary label and select by the highest permuted score with uniform ties. We average exactly over all permutations. The permutation keeps each label group's scores, so it removes exactly the scorer's within-group term of Proposition 1 (Table 21).

Table 21: Final correctness (%) of continuous scorers before and after permuting their scores within label groups, with the difference (points), its 95% problem-cluster interval, and the interval with Bonferroni correction over the five comparisons.

Evaluation set	Scorer	Original	Permuted	Difference	95% interval	Bonferroni
PRM800K	Qwen-PRM	85.66	84.40	+1.26	[+0.17, +2.36]	[−0.18, +2.73]
PRM800K	Skywork-PRM	82.88	83.77	−0.89	[−2.03, +0.25]	[−2.39, +0.61]
ProcessBench	Qwen-PRM	78.56	78.58	−0.01	[−0.98, +0.94]	[−1.31, +1.27]
ProcessBench	Skywork-PRM	77.06	77.50	−0.44	[−1.51, +0.59]	[−1.86, +0.89]
ProcessBench	VGS value model	76.09	76.05	+0.03	[−1.01, +1.04]	[−1.34, +1.35]

I.4 CODE-DOMAIN BREAKDOWN

The 223-problem pooled evaluation contains 63 HumanEval+, 85 MBPP+, and 75 LiveCodeBench problems. Qwen3's final-correctness direction is consistent across all three benchmarks. The full gpt-oss dissociation is clearest on MBPP+. Figure 2 reports the pooled result. Per-benchmark estimates appear below.

Table 22: Code-domain pooled result and per-benchmark breakdown (percentage points). Each row uses three-target family-wise 95% problem-cluster intervals. These intervals are not simultaneous across rows. MC value denotes MC value selection, and judge score denotes validity selection with one judge draw's continuous scores.

Dataset (N)	Qwen3		gpt-oss	
	MC value − judge score final ↑	judge score − MC value label ↑	MC value − judge score final ↑	judge score − MC value label ↑
Pooled (223)	**+18.39 [+14.49, +22.56]**	**+1.60 [+0.25, +3.18]**	**+1.90 [+0.37, +3.70]**	**+3.94 [+2.44, +5.62]**
HumanEval+ (63)	+19.89 [+12.13, +28.53]	−0.26 [−1.06, +0.00]	+0.76 [−0.39, +2.15]	+3.72 [+1.31, +6.79]
MBPP+ (85)	+19.68 [+13.33, +26.82]	+2.43 [−0.16, +5.29]	+4.00 [+0.75, +8.11]	+6.15 [+3.46, +9.20]
LiveCodeBench (75)	+15.68 [+9.48, +22.76]	+2.21 [−0.22, +5.57]	+0.49 [−1.75, +2.95]	+1.63 [−0.61, +4.20]

J Detailed Scope

Fixed evaluation sets and finite rollout evidence. All confidence intervals condition on the realized candidate sets, rollout sets, and search trajectories. They quantify uncertainty over sampled problem clusters, not variation from regenerating candidates, policies, or rollouts. Recoverability is defined at the stated rollout policy, prompt format, verifier, and token budget. Changing these conditions changes the measured recoverability. Exact ties are common at the measured rollout budgets and are resolved uniformly.

Validity targets. PRM800K and ProcessBench use official labels. Natural OpenR1, SuperGPQA, and code use outcome-blind model judges because comparable official labels are unavailable. These judges never observe rollout outcomes or scorer values, but their labels remain judge-relative. The two code-judge draws agree on 1,038/1,115 candidate labels (93.09%). Code selection ranks by one draw's continuous scores even when the two binary judgments agree. The pooled code result is therefore evidence for the declared validity target rather than a substitute for expert annotation.

Label-reading selectors and constraints. Validity selectors that read official labels measure how far an exact measurement of step validity determines the recoverability ranking. The validity-filter $\rightarrow$ MC value selector measures the price of enforcing validity, 0.01 points on PRM800K and 0.22 on ProcessBench for the tested policies and candidates. The tie counterexample of Example C.1 holds under its stated sampling assumptions.

Budget scans. The tie-breaking budget scans use the PRM800K and ProcessBench fixed-candidate sets with official labels (Appendix E.1) and two sets of unlabeled AIME traces, the fresh 2024–2026 traces (Appendices E.2 and E.3) and the search's own AIME 2026 selection points (Appendix F.5). The source budget stops at the recorded bank, which holds eight rollouts per candidate on PRM800K, ProcessBench, and the AIME 2026 selection points, and 32 on the fresh AIME traces.

Model and domain coverage. The fixed-candidate dissociation spans mathematical reasoning, code generation, and scientific multiple choice under Qwen3 and gpt-oss rollout policies. Public PRM comparisons remain concentrated in mathematical reasoning and use Qwen-family-generated candidates. The six rollout policies of Table 17 vary checkpoint and training jointly. Recoverability estimates must be revalidated when the rollout policy or token budget changes. Our FP32 scores for the human-label PRM differ from the example step scores on its model card by up to 0.03, although the example's validity classification and score order are unchanged. These scores enter only the human-label PRM rows, and the official-label, Qwen-PRM, and PUM results do not read them.

Sequential selection. The original-token study uses Qwen3-4B-Instruct-2507 on MATH, AIME 2024–2026, and GPQA Diamond, with four selection rounds and a fixed total trajectory budget. These experiments compare inference-time selection rules over four rounds, and complete-answer PRM comparisons cover MATH and AIME 2024–2025. Training process supervisors and deeper search remain open.