

Target-Local Verifier Choice in Best-of- K Reasoning Selection

Xin Li Hao Jiang Weisi Lin
Nanyang Technological University, Singapore
<https://lixin.ai/TLVC/>

Abstract

Best-of- K pipelines sample K candidate solutions, score them with a verifier, and keep the top-ranked answer—typically reusing one verifier across all generators. We test that reuse assumption on a broad mathematical-reasoning panel: 34 generators, $K=64$ candidates per problem, seven primary verifiers, plus two cross-lineage verifiers as robustness checks. The strongest fixed verifier, a 7B process reward model, is not the best verifier for every generator. Label-free statistics of the candidate texts predict which verifier class wins on each target, and the two reliable budgeted policies—process/outcome score fusion and a small-label-budget per-target arbiter—beat the fixed default by roughly two points on two separately sampled candidate banks; a no-label router is seed-sensitive. We frame the result as a configuration- and verifier-family-conditioned deployment diagnostic rather than a universal routing rule; improved selection did not transfer to student fine-tuning in an exploratory check.

1 Introduction

Reasoning ability is a central target for modern large language models, and reward models or verifiers are one of the main interfaces for improving it. They can train models toward better traces, but they also serve a simpler inference-time role: select the best answer from a set of sampled candidates (Cobbe et al., 2021; Wang et al., 2022; Lightman et al., 2024; Snell et al., 2024; Brown et al., 2024). In Best-of- K selection, a system samples K candidate solutions, scores them, and returns the verifier’s top-ranked candidate.

Most verifier work asks how to build a stronger verifier or how much test-time sampling to spend. We ask a different deployment question. When a service uses many generators, can one strong process reward model (PRM) be calibrated once

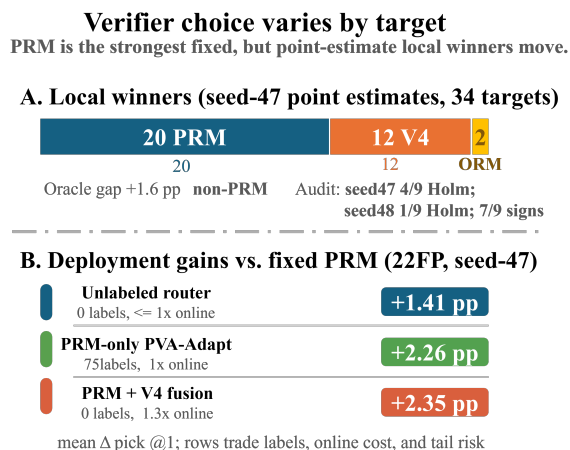


Figure 1: **Target-local verifier allocation.** Fixed process reward model (PRM) scoring is the strongest global default, but some target candidate distributions prefer outcome verifiers. The lower panel shows budgeted gains over fixed PRM; seed-48 preserves the scored and labelled gains while weakening the no-label router.

and reused for all of them, or does the best verifier depend on the served candidate distribution? Throughout, “target generator” denotes the distribution induced by a generator, prompt template, decoding settings, and answer-extraction protocol.

We test this on a matched 34-generator $\times$ 7-primary-verifier math panel at $K=64$, with two cross-lineage verifiers added as robustness checks. Fixed PRM-7B remains the strongest global verifier (43.5% mean pick@1), but the best local verifier is not constant: the per-target single-verifier oracle leaves fixed PRM with +1.6 percentage-point (pp) regret on the full panel. The local winners split across PRM, V4 outcome verifiers, and scalar outcome reward models (ORMs), so the effect is not a search for a weaker global baseline. We treat the winner counts as a mismatch diagnostic rather than as individually significant claims: direct paired tests support the high-margin non-PRM cases, and a second candidate-bank check

preserves the scored and labelled policy gains while weakening the no-label router.

The mismatch is partly visible before running every verifier. Candidate sets differ in answer concentration, solution length, boxed-answer rate, and max-cluster dominance. These statistics can encode style as well as reasoning quality, so we read them diagnostically rather than causally. They classify the coarse PRM/V4 label under target leave-one-out (target-LOO; 19/21 first-pass; 27/32 combined), survive family and architecture holdouts, and preserve 83–91% of winner classes under two alternative prompt templates.

We convert the diagnostic into budgeted policies evaluated by panel-average deltas against fixed PRM. With no target labels, a gated router gains +1.41 pp; with extra scoring, PRM+V4-2B fusion gains +2.35 pp; with $m=75$ labelled probes, a PRM-only probe-based verifier-allocation arbiter (PVA-Adapt) gains +2.26 pp at fixed-PRM test cost. The same-panel American Invitational Mathematics Examination (AIME) 2026 holdout gives +3.88 pp. The gains are modest and the router has a −3.33 pp worst target loss. The scored fusion and labelled PVA-Adapt are therefore our primary recommendations; the router is a seed-sensitive no-label option, not a default. We also test two edges of the claim: a small code stress check is suggestive, while a Qwen3-8B Instruct supervised fine-tuning (SFT) study finds no positive transfer from better selected traces.

The diagnostic is conditioned on the complete generator $\times$ prompt $\times$ decoding configuration, rather than on the generator name alone, and should be recomputed when that configuration changes.

We make three contributions:

1. **A target-local verifier allocation study.** We build a matched 34-generator $\times$ 7-primary-verifier math panel showing that fixed PRM-7B is the strongest global verifier but still leaves +1.6 pp to a per-target single-verifier oracle.
2. **A label-free diagnostic and budgeted policies.** Four candidate-text features predict the coarse PRM/V4 regime and support a gated router, PRM+V4 fusion, and labelled PRM-only PVA-Adapt.
3. **Robustness and limits.** We report family/architecture holdouts, prompt-template stress tests, AIME 2026 calibration holdout, a second candidate-bank check, a small code

pilot, and an exploratory non-positive SFT check.

2 Related Work

Our work sits at the intersection of process verification, test-time selection, and routing. The distinction is the deployment object: prior work usually asks how to train a stronger verifier, how to spend more test-time compute on a prompt, or how to route prompts to generators. We ask which verifier should serve a *target generator’s candidate distribution* under a fixed reranking budget.

Process verifiers and reward-model evaluation.

Process reward models (PRMs) were introduced as step-level verifiers for mathematical reasoning (Lightman et al., 2024); later work reduced the annotation burden with automatic process labels (Wang et al., 2024), progress-style process rewards (Setlur et al., 2025), weak supervision (Sun et al., 2025), and generative or reasoning-driven verifiers (Khalifa et al., 2025; Zhao et al., 2026; She et al., 2025). The Qwen2.5-Math-PRM report (Zhang et al., 2025) provides the public PRM-7B verifier we use as the strongest global fixed baseline.

Recent benchmarks sharpen the evaluation problem. ProcessBench asks models to identify the earliest erroneous step in competition-level math solutions (Zheng et al., 2025); PRMBench broadens process-level evaluation to fine-grained error types and reports substantial weaknesses in current PRMs (Song et al., 2025). RewardBench 2 documents brittleness and downstream-evaluation gaps for general reward models (Malik et al., 2025), while Skywork-Reward-V2 scales open reward-model training across preference domains (Liu et al., 2025). These papers improve or stress-test the verifier itself. Our question is complementary: even when PRM-7B is the best single fixed verifier on average, it is not the best verifier for every generator distribution.

Test-time selection and compute allocation.

Best-of- N selection with an ORM or PRM is a standard way to convert sampling compute into accuracy (Cobbe et al., 2021; Lightman et al., 2024; Hosseini et al., 2024). Self-consistency clusters answers without a learned verifier (Wang et al., 2022); in our panel it remains a useful label-free baseline but does not close the fixed-PRM gap (Section M). Jain et al. (2024) also use lightweight statistics over sampled generations for reranking;

we use candidate-text statistics to diagnose which verifier should serve a target distribution. Recent test-time-scaling work studies how repeated sampling scales coverage (Brown et al., 2024), how to allocate inference compute more efficiently than naive Best-of- N (Snell et al., 2024), and how to use a generative model to compare an entire candidate set (Toshniwal et al., 2025).

The closest compute-allocation paper is Damani et al. (2025): it predicts per-prompt difficulty and chooses how much decoding or search effort to spend. We instead hold the candidate set fixed and allocate *verifier identity*. This difference matters operationally: our zero-label router can be run after candidates are sampled, before verifier scoring, using only candidate-distribution statistics.

Routing and model selection. Large language model (LLM) routing routes prompts to generators under quality/cost tradeoffs. FrugalGPT introduced LLM cascades for cost reduction (Chen et al., 2023), and RouteLLM learns strong-vs-weak model routers from preference data (Ong et al., 2024). Recent work extends this line to unseen-model routing with learned model feature vectors (Jitkrittum et al., 2025), large-scale router evaluation (Huang et al., 2025), multi-round routing and aggregation (Zhang et al., 2026), and expected-reward prediction for prompt-to-generator routing (Hasanaliyev et al., 2026). Reward model routing is closer in object space: Wu and Lu (2025) routes among reward models during alignment, and Xu et al. (2026) routes uncertain reward-model decisions to a stronger LLM judge.

Our setting differs along three axes. First, the routed object is the verifier used for mathematical candidate reranking, not the generator answering the prompt and not an expensive judge fallback. Second, the routing unit is a target generator distribution rather than an individual prompt or reinforcement-learning-from-human-feedback query. Third, the cold-start router uses no labels and no preflight verifier scores to choose the verifier: it reads only the candidate texts. Table 27 gives the full axis-by-axis comparison.

Verifier-selected data for post-training. Verifier-filtered or reward-filtered candidates can serve as self-training data. V-STaR trains verifiers for self-taught reasoning (Hosseini et al., 2024); ReST-EM iterates sampling, filtering, and fine-tuning (Singh et al., 2023); Setlur et al. (2024) report gains from RL on synthetic data; and Chen

et al. (2026) tests how math PRMs transfer to code. Our SFT result is deliberately negative: in our setup, better rerank-pick@1 selection does not predict better student utility (Section 5.3). This does not contradict positive self-training results; it identifies a boundary between *choosing the correct candidate now* and *choosing data that a smaller student can imitate*.

Positioning. Our contribution is a verifier-allocation study. We hold the candidate generation protocol fixed, vary verifier identity, and ask when a strong global PRM should be reused, fused with an outcome verifier, or adapted with a small labelled probe set. On a 22-generator policy panel and a 34-generator extended panel, the best fixed verifier is not always the best local verifier; the observed local regime is predictable from candidate texts; and target-local candidate arbitration improves further under small label budgets while exposing a separate SFT-utility limit.

3 Setup and Target-Local Verifier Mismatch

Evaluation protocol. A target is the served candidate distribution induced by generator, prompt, decoding, and extraction protocol. We evaluate 34 such targets: a 22-generator first-pass panel used for the main policy comparison, plus 12 second-pass generators used to test whether the pattern persists after the diagnostic is fixed. All generators produce $K=64$ candidates per problem at $T=0.7$, $\text{top-}p=0.95$, $\text{max_new} = 4096$, seed 47.

The first-pass panel covers 13 model families: Qwen3.5-Base/Instruct, Qwen3 (0.6B–32B), Phi-4 (mini and mini-reasoning), Llama-3.2, SmoLLM3, Falcon-H1, Granite-3.1, DSR1-Distill-Qwen, InternLM3, OLMo-2, Hermes-4, and QwQ. The second-pass extension adds Qwen2.5-Instruct scales, Qwen2.5-Math-7B-Instruct, Mistral-Nemo-12B, Phi-4 14B, Yi-1.5-34B-Chat, Qwen3-30B-A3B-Instruct-2507, DSR1-Distill-Llama-70B, and GPT-OSS-120B. We refer to the 12-generator second-pass extension as *Wave A* and its cross-template re-run (T1 open CoT; T2 structured steps) as *Wave B*.

All results are conditioned on the generator $\times$ prompt $\times$ decoding $\times$ extraction configuration. The label-free diagnostic characterizes a configuration, not a generator in isolation, and should be recomputed whenever the prompt or decoding configuration changes.

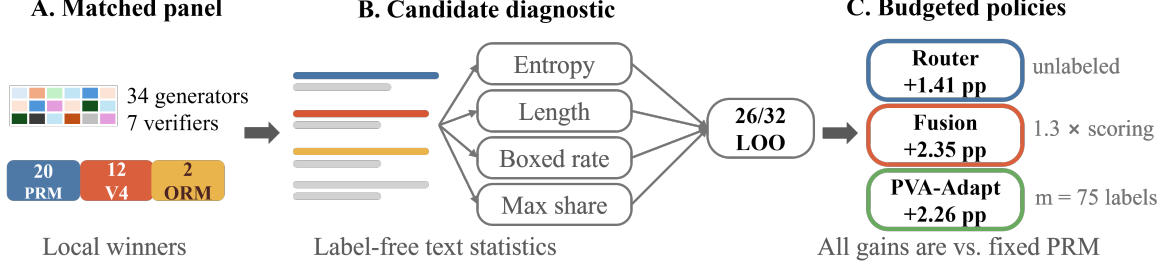


Figure 2: **Overview.** We measure target-local verifier mismatch, diagnose it from candidate texts, and evaluate budgeted response policies. Full costs and CIs are in Table 1.

Local-winner labels are point estimates on this fixed candidate bank. The deployment tables use the 22 first-pass targets; binary routing uses the 21 first-pass and 32 combined PRM/V4-routable targets, excluding ORM-class targets from the router denominator. The treatment of calibration problems for labelled policies is specified with their protocol in Section 4. Paired CIs condition on the seed-47 candidate draw and on one verifier family per class; the seed-48 check in Table 1 is a robustness check rather than a population-level seed CI.

The math eval mix is frozen across all generators: AIME 2024 (30), AIME 2025 (30), AIME 2026 (30), OlympiadBench-math-hard (110), and MATH-500 (100). The verifier pool has three classes, with one representative family per class: (i) *PRM class* — Qwen2.5-Math-PRM-7B, a process reward model with step-level signals (Zhang et al., 2025); (ii) *V4 class* — V4-2B/4B/9B Bradley-Terry outcome reward verifiers trained on within-problem math candidate pairs with Qwen3.5-size backbones; (iii) *ORM class* — ORM-2B/4B/9B, scalar outcome reward models. All seven primary verifiers consume the full candidate trace; selection is argmax. Section P lists the exact V4/ORM pair construction, backbones, and training configs.

The selection metric is top-1 pick@1 (mean over 300 problems and paired-bootstrap 95% confidence interval, CI95, against fixed PRM-7B). Cost is dominated by verifier forward passes: with PRM-7B normalised to 1, the 2B/4B/9B verifiers cost $\approx 0.3\times, 0.6\times, 1.3\times$. The PRM-only PVA-Adapt variant in Section 4 preserves $1\times$ deployment cost.

Fixed-verifier regret. On the merged 34-generator panel a practitioner who deploys fixed PRM across all targets loses +1.6 pp of pick@1 relative to a per-target single-verifier oracle (45.1% vs. 43.5%, panel-weighted mean); against an in-sample per-problem oracle the gap is +9 pp. On

the 22-target first-pass slice alone, the fixed-PRM reference is 43.02% and the per-target oracle gap is +1.85 pp: the per-target single-verifier oracle averages 44.86%, and the per-problem oracle upper bound is 52.47% (+9.45 pp). PRM-7B is the strongest single fixed verifier on either slice, so the regret is against the *best* fixed alternative.

Because this per-target oracle takes a maximum over seven noisy primary verifier estimates, it can be upward-biased. On the 22-target slice, its +1.85 pp is a reference for the policy class that commits to one verifier per target. The +9.45 pp per-problem oracle is a separate candidate-level hindsight reference: the router recovers about 15% of that headroom, while fusion and PVA-Adapt recover about 24–25%. Fusion and PVA-Adapt are outside the one-verifier-per-target policy class and can therefore exceed the +1.85 pp reference without contradiction. All deployable policies are evaluated directly against fixed PRM. The 22-target local-oracle distribution is 13 PRM, 8 V4, and one ORM-boundary target.

Winner-vs-runner-up tests establish 8 Holm-significant cells, all PRM-class; the deployment-relevant comparison is more direct for the nine non-PRM point-estimate winners. Comparing each such winner against fixed PRM gives 4/9 Holm-significant contrasts under the primary family-wise correction (6/9 raw and 6/9 Benjamini–Hochberg (BH) sensitivity) (Table 14). Thus the non-PRM labels are partly statistically supported, partly descriptive, and the downstream claim is evaluated primarily through panel-average gain over fixed PRM.

A targeted second candidate-bank check reranks the same nine seed-47 non-PRM winners directly against PRM under seed 48. The direction preserves on 7/9 targets, but only 4/9 are positive and raw-significant, 1/9 survives Holm, and 2/9

survive BH; one boundary target flips significantly toward PRM (Table 15). We treat this as evidence that the high-margin cases partly replicate while candidate-bank variance is material.

We also rerun the full 22-target first-pass deployment table on seed 48: point-estimate winner classes become 15/6/1 PRM/V4/ORM, the per-target oracle gap is +1.52 pp, PRM+V4-2B fusion gains +1.95 pp, and PRM-only PVA-Adapt gains +1.93 pp; the no-label gated router weakens to +0.92 pp with CI95 crossing zero (Table 1; full table in Table 16). Thus the scored and labelled policies survive a second candidate-bank draw, while cold-start routing is more seed-sensitive.

Second-pass extension and template robustness.

The 12 second-pass generators split 8/3/1 for PRM/V4/ORM winners; mean per-target single-verifier oracle gain over fixed PRM on these targets is approximately +1.04 pp. A specific sub-pattern: every reasoning-tuned or mixture-of-experts (MoE) large generator we add — Qwen3-30B-A3B-Instruct-2507 (ORM-class), DSR1-Distill-Llama-70B (V4-class), GPT-OSS-120B (V4-class) — is non-PRM, while all five Qwen2.5-Instruct dense scales (1.5B–72B) stay PRM-class. The merged panel has 11 V4-class and 2 ORM-class/boundary winners; 11/34 counts V4-class winners only.

For cross-template robustness, Wave B re-runs all 12 second-pass generators under two alternative system prompts (T1 open chain-of-thought, CoT; T2 structured steps). It preserves the single-verifier-oracle winner class on 10/12=83% of targets under T1 and 10/11=91% evaluable targets under T2; all evaluable PRM-class T0 winners remain PRM under both alternative prompts. The remaining consistent template-driven flip is Mistral-Nemo-12B. The frozen diagnostic had predicted PRM for this target ($p_{V4} \approx 10^{-4}$), while its original observed winner was V4 by only 1 pp; both alternative templates move the observed winner to PRM, toward the frozen prediction. We therefore do not attribute this flip to the length feature alone.

Cross-family scale sweeps. Within the 22-target panel and auxiliary Phi stress banks, three family-scale views move in distinct directions. **Qwen3.5-Base** (0.8B $\rightarrow$ 9B) moves from PRM (McNemar $p=0.018$ at 0.8B) to V4-class; the 4B and 9B V4 winners beat fixed PRM by +6 and +8.33 pp and both survive the direct winner-vs-PRM Holm comparison.

Qwen3 (0.6B $\rightarrow$ 32B) moves in the *opposite*

direction: V4-class at 1.7B–4B, crossover near 8B, PRM at 32B (+8.33 pp, $p=0.0015$).

Phi is a three-bank stress view rather than a clean one-generation scale law: Phi-3-mini, Phi-3-medium, and Phi-4 all stay PRM-class, with margin shrinking from +13.67 pp to +5.33 pp. We therefore use the scale sweeps descriptively, not as small- n logistic-slope inference. Of the 22 McNemar winner-vs-runner-up tests 10/22 are significant uncorrected at $\alpha=0.05$, 8/22 survive Holm–Bonferroni, and 10/22 survive Benjamini–Hochberg. Within this panel, the three sweeps are inconsistent with the simplest size-only explanation; per-family tables, post-training and decoding-shift controls are in Sections C to D.

One candidate-distribution hypothesis is that family-specific post-training changes the served distributions differently with scale. Qwen3.5-Base keeps its boxed-answer rate near 47–55% while answer entropy falls from 1.54 to 0.20–0.27 as it moves from PRM to V4. Qwen3 instead shifts from 51% boxed and roughly 7.7k tokens at 4B to 85–97% boxed and roughly 2.2k–1.1k tokens at 8B–32B while moving from V4 to PRM. Similar pass@64 at the two large endpoints (70.7% vs. 71.0%) weakens a simple ability-only explanation. These associations are correlational, and the low-margin Qwen3-0.6B endpoint changes regime across candidate banks.

4 Budgeted Responses Under Explicit Costs

We turn the mismatch into a budgeted response menu indexed by label budget and online verifier cost (Figure 2). PVA denotes probe-based verifier allocation: PVA-Calibrate chooses a verifier class from labelled probes, while PVA-Adapt trains a target-local candidate arbiter. Our primary recommendations are no-label extra-scoring fusion (+2.35/ +1.95 pp on candidate Banks 1/2 at $1.3\times$ cost) and labelled-probe PRM-only adaptation (+2.26/ +1.93 pp with $m=75$ labels at $1\times$). The no-label one-verifier router (+1.41/ +0.92 pp, $\leq 1\times$) is a seed-sensitive option, not a default recommendation: its Bank 2 interval crosses zero. Table 1 is the primary paired-bank deployment comparison; appendix tables provide sensitivity analyses (Table 3 isolates the full-7 labelled-probe m-curve).

Fusion is a primary mean-gain policy but is not uniformly safe: it wins on 16/22 targets, loses on

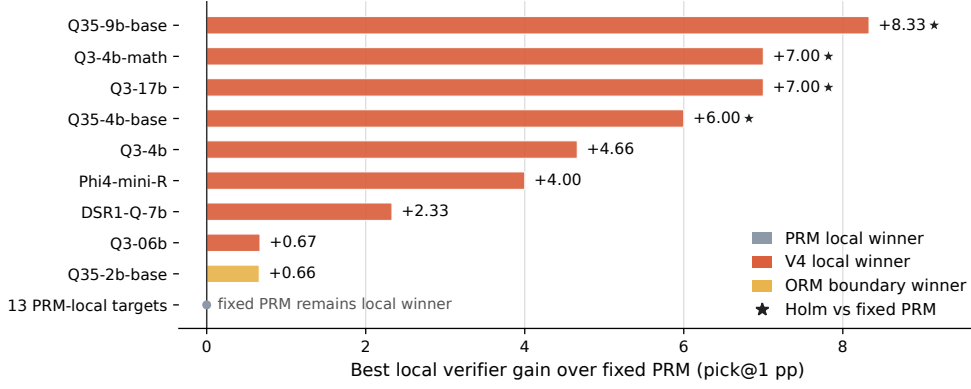


Figure 3: **Local verifier winners on the 22-target panel.** Thirteen PRM-local targets are collapsed into one grey row; the eight V4 winners and one ORM-boundary winner are shown individually. Direct comparison details are in App. B; the seed-48 policy check is summarized in Table 1.

6/22, and its worst target loss is -3.33 pp. Because fusion mixes PRM and V4 scores per candidate rather than committing to one verifier per target, it is not bounded by the per-target single-verifier oracle gap. We treat all policies as deployment options to validate per target rather than autonomous rollout controllers.

The policy definitions are fixed before evaluation: fusion is the z-scored PRM+V4-2B policy, the router threshold is selected by nested target cross-validation, PVA variants use the fixed hyperparameters in the appendix, and AIME calibration excludes AIME 2026 problems.

4.1 A four-feature candidate-text diagnostic

For each PRM/V4-routable target, we compute four statistics from the $K=64$ candidate texts before verifier scoring: boxed-rate, log length, answer-cluster entropy, and max-cluster share. An L_2 -regularised target-LOO classifier predicts the point-estimate PRM/V4 regime with $19/21=90.5\%$ accuracy on the original first-pass panel and $27/32=84.4\%$ on the combined panel, five errors and a 6.1 pp drop after adding the second-pass targets (Wilson CI95 $[71.1, 97.3]\%$ and $[68.2, 93.1]\%$, respectively) (Figure 4 and section E); the first-pass permutation null gives $p=0.0003$, and the combined-panel V4 prior is $11/32=34.4\%$.

The classifier is used as a serving heuristic: its labels are seed-47 point-estimate regimes rather than significance-screened truth, and the deployment claim is evaluated separately through policy gains in Table 1.

This signal is not a single first-pass number: combined target-LOO is 27/32, fam-

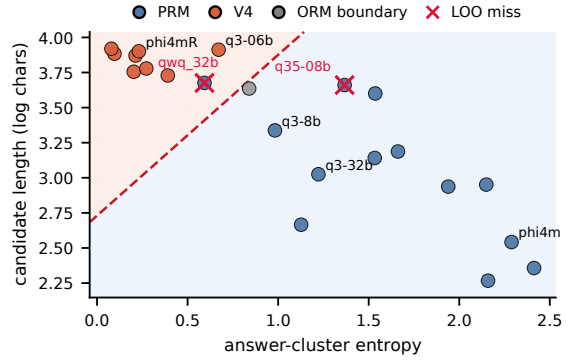


Figure 4: **Candidate-text diagnostic.** This entropy-length slice visualizes the fitted four-feature classifier with boxed-rate and max-share fixed to their medians. Points are target averages; target-LOO accuracy is 19/21 first-pass and 27/32 combined.

ily/architecture holdouts are 11/12 and 20/21, dropping boxed-rate or length leaves 20/21, parser/style controls give 19–20/21, and a metadata-only size/training-tag router is much weaker (13/21, -0.67 pp). Cross-template winner class is preserved on 10/12 under T1 and 10/11 evaluable targets under T2; equivalently, 2/12 and 1/11 targets flip, all at original winner margins ≤ 1.67 pp. The features may still proxy coherence, extraction regularity, or difficulty rather than causality; full ablations are in App. D and App. P.

4.2 Cold-start router: protocol and Pareto

For a new target, the router draws $B=300$ unlabeled problems, generates $K=64$ candidates, computes target-level statistics, and routes to PRM or V4-2B only when confidence exceeds the nested cross-validation gate threshold (modal 0.7); oth-

Policy	Labels	Test cost	Bank 1 Δ [CI95]	Bank 2 Δ [CI95]	Worst Bank 1/2
fixed PRM	0	1 $\times$	0.00	0.00	0.00/0.00
<i>No labels</i>					
cluster+PRM top-3	0	1 $\times$	+0.89 [+0.27, +1.56]	+1.03 [+0.50, +1.59]	−1.33/ − 1.00
gated router	0	$\leq 1\times$	+1.41 [+0.21, +2.76]	+0.92 [−0.22, +2.19]	−3.33/ − 5.33
PRM+V4-2B fusion	0	1.3$\times$	+2.35 [+1.09, +3.53]	+1.95 [+0.73, +3.17]	−3.33/ − 4.00
<i>75 labelled probes; offline calibration</i>					
PVA-Calibrate	75	1 $\times$	+1.12 [+0.14, +2.31]	+0.78 [−0.06, +1.81]	−1.31/ − 1.38
PVA-Adapt PRM-only	75K	1$\times$	+2.26 [+1.11, +3.42]	+1.93 [+0.84, +3.04]	−1.87/ − 2.12
PVA-Adapt full-7	75K	5.4 $\times$	+2.02 [+0.95, +3.05]	+1.72 [+0.65, +2.83]	−2.39/ − 1.96
<i>Hindsight reference (not deployable)</i>					
per-target single-verifier oracle	test labels	1 $\times$	+1.85	+1.52 [+0.56, +2.64]	0.00/0.00

Table 1: **Paired candidate-bank policy results on the 22 first-pass targets.** Δ and Worst are mean and worst-case changes in pick@1 vs. fixed PRM (pp); brackets are target-bootstrap CI95 and do not estimate population-level seed variance. Bank 1 uses seed 47 and Bank 2 seed 48. The Bank 2 router entry is a fresh nested target-LOO refit on Bank 2, not a frozen Bank 1 $\rightarrow$ Bank 2 transfer. Labels are problem-level calibration probes; 75K denotes answer-matched candidate labels. Bold rows are the primary recommendations.

Available budget	Recommended policy	Evidence and caution
No validation; risk-sensitive	fixed PRM-7B	Strongest fixed verifier and safest default.
No labels; $\leq 1\times$ scoring	gated router	+1.41/ + 0.92 pp on Banks 1/2, but seed-sensitive; worst loss −3.33/ − 5.33 pp.
No labels; 1.3 $\times$ scoring	PRM+V4-2B fusion	Primary no-label policy: +2.35/ + 1.95 pp, with positive CI95 on both banks.
$m=75$ labels; 1 $\times$ scoring	PVA-Adapt PRM-only	Primary labelled policy: +2.26/ + 1.93 pp and the smallest observed worst-target losses among adaptive policies.

Table 2: **Deployment guidance.** Results are reported as Bank 1/Bank 2 changes from fixed PRM. Adaptive policies should be deployed with monitoring and rollback.

erwise it falls back to PRM. This is label-free and score-free, but not sample-free; smaller- B and smaller- K deployment budgets remain future work. A within-bank subsampling check in Section E shows that the diagnostic label prediction itself is stable down to smaller B/K settings.

Table 1 separates three budgets. Calibration labels are problem-level gold answers; PVA-Adapt turns them into mK candidate labels by answer matching. For the labelled-policy holdout protocol, see Section 4.4. Fusion and routing target different budgets: extra test-time scoring versus one-verifier deployment. Calibration scoring is a one-time cost on probes; test cost is the online verifier scoring cost on deployment candidates, normalised to fixed PRM.

The router changes 9/21 targets (+3.30 pp on changed targets, +1.41 pp overall) and leaves 12/21 at PRM. The two routed losses are Qwen3-0.6B (−2.67 pp) and QwQ-32B (−3.33 pp). Meta-data and answer-cluster baselines trail the router/fusion policies (App. L); per-target diagnostics are in Section F. The gate reduces the ungated worst loss from −11.67 pp to −3.33 pp by abstaining. On Bank 2, a fresh nested-LOO re-

fit gains +0.92 pp ([−0.22, +2.19] pp) with worst loss −5.33 pp; it is not a frozen Bank 1 $\rightarrow$ Bank 2 transfer. Raising the gate to $\tau=0.9$ removed every observed loss only because it never routed and therefore degenerated to fixed PRM; this is a risk-gain observation, not a guarantee. Remaining losses motivate a staged rollout: start with fixed PRM, use the diagnostic to flag targets, validate any switch on labelled probes, and retain rollback. The two Bank 2 worst cases have different failure modes: the router’s −5.33 pp occurs on the 0.67 pp-margin Qwen3-0.6B boundary target, which changes regime across banks, whereas fusion’s −4.00 pp occurs on Hermes-4-14B, where fixed PRM leads by 10 pp on both banks and mixing in V4-2B dilutes the stronger score.

4.3 PVA-Calibrate: a class-level rule from m problem labels

PVA-Calibrate uses m calibration probes to pick one verifier class. It computes per-verifier calibration pick@1, collapses by class ($a_C = \max_{V_i \in C} \hat{a}(V_i)$), and leaves PRM only if a non-PRM class beats it by shrinkage margin $\delta=2$ pp. Test cost stays 1 $\times$.

4.4 PVA-Adapt: a target-local candidate arbiter

PVA-Adapt converts the same probes into answer-matched mK candidate labels and trains a target-local L_2 -regularised candidate scorer. Each calibration problem contributes labels for all K candidates by exact-answer matching; test means exclude those calibration problems. For the $m=75$ policies, probes are sampled per target and candidate bank and the reported test mean uses the remaining $300-m$ problems; AIME 2026 instead calibrates only on the 270 non-AIME-2026 problems. Features include verifier scores/ranks/margins plus candidate text/cluster features (definitions and C sweep in Section H).

On the 22-target panel, full-7 PVA-Adapt is barely positive at $m=20$ ($[+0.07, +2.25]$ pp); more comfortable gains start at $m=30$ and grow through $m=100$. PVA-Calibrate is weaker and its CI excludes zero only at $m=75$ and 100. The m-curve uses full-7 verifier-score features; the PRM-only deployment variant is reported separately below and in Table 1.

For deployment comparisons, the $+2.26$ pp PRM-only result in Table 1 is the $1\times$ -cost result; the separate full-7 probe-budget curve in Table 3 describes how the labelled arbiter scales with m .

Calib. probes	PVA-Calibrate Δ [CI95]	PVA-Adapt full-7 Δ [CI95]
20	-0.05 $[-1.16, +0.95]$	$+1.17$ $[+0.07, +2.25]$
30	$+0.33$ $[-0.76, +1.45]$	$+1.50$ $[+0.42, +2.56]$
50	$+0.87$ $[-0.18, +1.95]$	$+1.85$ $[+0.77, +2.91]$
75	$+1.13$ $[+0.12, +2.14]$	$+2.02$ $[+0.98, +3.03]$
100	$+1.26$ $[+0.30, +2.21]$	$+2.24$ $[+1.18, +3.29]$

Table 3: **Label-budget curve.** PVA-Calibrate and full-7 PVA-Adapt gains vs. fixed PRM on the 22-target seed-47 panel (50 calibration resamples; paired-bootstrap CI95). Values are pp.

The paired-bank results formerly repeated in a separate seed table are now consolidated in Table 1: fusion and both PVA-Adapt variants retain positive intervals on Bank 2, whereas the router interval crosses zero.

4.5 Budget-aware variant: PRM-only at $1\times$ test cost

The PRM-only variant keeps PRM scores plus four label-free candidate features, so test cost matches fixed PRM while recovering $+2.26$ pp at $m=75$. The full 7-verifier variant is a fixed reference, not a tuned ensemble; a 20-seed subset sweep ranges

from $+2.14$ to $+2.58$ pp. Extra verifier-score features can add calibration variance at $m=75$, so the PRM-only variant is the most deployment-efficient result rather than a claim that fewer verifier features are intrinsically better; target-local adaptation and candidate features drive the gain.

5 Held-out and Boundary Checks

5.1 AIME 2026 held-out slice

In the strict split, the 270 non-AIME-2026 problems supply calibration probes, and the 30 AIME 2026 problems form the deterministic test set for all 22 targets.

Selector	Within-pool	Held-out
PVA-Calibrate	$+2.45$ $[+1.04, +3.96]$	$+2.45$ $[+0.77, +4.37]$
PVA-Adapt full	$+3.88$ $[+2.11, +5.74]$	$+3.88$ $[+1.56, +6.15]$
oracle	$+9.55$ $[+7.42, +11.82]$	$+5.30$ $[+3.33, +7.42]$

Table 4: **AIME 2026 held-out slice.** Δ pick@1 vs. fixed PRM-7B on the 30 AIME 2026 problems, in percentage points ($m=75$, 50 calibration seeds); subscripts are bootstrap 95% CIs. The held-out split calibrates on the other 270 problems.

PVA-Adapt full retains a $+3.88$ pp gain over fixed PRM (CI95 $[+1.56, +6.15]$). The selected policies match the within-pool run, while oracle headroom drops to $+5.30$ pp. This is a same-panel leakage check, not new-generator or private-benchmark transfer; per-family detail is in Section L.

5.2 Out-of-domain code pilot

As a boundary check on whether target-local mismatch is math-specific, we ran a code pilot over five *cells*, each a (target, benchmark) pair from HumanEval (HE) and MBPP, reranked by the four code-trained outcome verifiers (three V4C scale variants and one ORMC variant), plus fixed PRM-7B. In all five cells the winner is non-PRM and survives Holm/BH correction (Table 5); margins ($+10.98$ to $+21.95$ pp) far exceed the math panel, and Qwen3.5-9B even changes winner between HE and MBPP. A four-target HumanEval+ follow-up is inconclusive, so the pilot shows only that mismatch is not math-exclusive, not a code-domain transfer claim.

5.3 Exploratory SFT-utility boundary check

Table 1 and the AIME 2026 check measure selector correctness; a separate question is whether better-selected traces make better SFT data. As an ex-

Target / bench	Winner	Δ (pp)	n
Q35 / HE	V4C-9B	+10.98 _[+2.44,+19.51]	164
Q35 / MBPP	V4C-4B	+8.95 _[+5.45,+12.84]	257
DSR1 / HE	V4C-4B	+8.54 _[+4.27,+13.41]	164
OSS / HE	ORMC-4B	+17.07 _[+7.93,+26.22]	164
Phi4R / HE	V4C-4B	+21.95 _[+13.41,+30.49]	164

Table 5: **Out-of-domain code pilot.** Winner-vs-PRM Δ pick@1 on HumanEval (HE) or MBPP, in percentage points; subscripts are bootstrap 95% CIs. All five rows pass Holm/BH within this pilot; V4C/ORMC are code-trained variants.

ploratory check, we fine-tune a Qwen3-8B Instruct student on full PVA-Adapt-selected traces ($m=75$) and evaluate pick@1 on 210 non-floor problems. In this setup, no SFT variant shows positive transfer, and the full-7-verifier arbiter is significantly *negative* (-6.19 pp, CI95 $[-11.90, -0.95]$); the intervals are wide. We therefore treat this only as an underpowered, non-positive boundary result, not evidence that improved selection can never benefit SFT (Section 6). Choosing the correct answer now is a different objective from choosing data a small student can imitate.

6 Conclusion

We have argued that verifier choice in Best-of- K selection is a *target-local allocation problem*, not a fixed configuration. On a matched 34-generator $\times$ 7-verifier math panel, PRM-7B is the strongest single fixed verifier, yet a per-target single-verifier oracle still leaves +1.6 pp of pick@1, with local winners split systematically across process, outcome, and scalar reward models. This mismatch is partly visible in label-free candidate-text statistics before any verifier is run. Rather than a new universal verifier, we propose a budgeted response menu over an existing one: a label-free gated router (+1.41 pp), PRM+V4 score fusion (+2.35 pp), and a labelled PRM-only arbiter (PVA-Adapt, +2.26 pp at fixed online cost), with +3.88 pp on the AIME 2026 held-out slice. A second candidate-bank draw keeps the scored and labelled policies positive while weakening the router, so we recommend keeping the strong PRM default and spending extra scoring or labels only on targets the diagnostic flags. The boundaries are as informative as the gains. The improvements are modest, the router carries a -3.33 pp worst-target loss, and the evidence remains configuration- and verifier-family-conditioned despite two cross-

Setting	Protocol	Δ (pp)
22-target	PVA-Adapt full	+ 2.02 _[+0.98,+3.03]
AIME 2026	PVA-Adapt full	+ 3.88 _[+1.56,+6.15]
8B SFT	PVA-Calibrate	-1.43 _[-4.29,+1.43]
8B SFT	PVA-Adapt full-7v	-6.19 _[-11.90,-0.95]
8B SFT	oracle data	-4.76 _[-10.48,+0.95]

Table 6: **Exploratory selection gain versus SFT utility.** Selection rows use full PVA-Adapt with $m=75$; SFT rows use a Qwen3-8B Instruct student. Δ in percentage points; subscripts are bootstrap 95% CIs.

lineage robustness checks. An exploratory SFT check also found no positive transfer from selector-better traces. Closing this gap calls for population-level evidence—more candidate-bank seeds, generator families, and independent verifier lineages—and allocation policies that recover more of the +9 pp per-problem oracle headroom that remains open.

Limitations

Magnitude and deployment risk. The deployable gains over fixed PRM-7B are small (+1.4–2.4 pp on the 22-target panel) and recover only part of the in-sample oracle gaps. The gated cold-start router still has a -3.33 pp worst target loss. We therefore view the policies as validated target-local adaptations, not categorical replacements for fixed PRM; real deployments would need labelled checks and rollback paths.

Scope and seed variance. The 34-generator panel remains small relative to production fleets. Its primary results use one canonical decoding setup ($T=0.7$, $\text{top-}p=0.95$, $K=64$); the $T=1.0$ and small Qwen3.5 controls do not cover the full decoding space. The code pilot is only a small out-of-domain check, and the router is reported at $B=300$, $K=64$; the appendix B/K sweep subsamples the fixed bank rather than validating a new smaller-budget router. Primary CIs condition on the seed-47 candidate bank. Seed-48 keeps the scored/labelled policies positive but weakens the no-label router, so more candidate-bank draws are still needed for population-level seed CIs.

Verifier coverage. The primary seven-verifier pool contains one PRM family, one Bradley–Terry outcome-verifier family, and one scalar ORM family derived from Qwen2.5/Qwen3 bases. The cross-lineage check adds two Llama-3.1-based verifiers (Skywork-Reward-V2 and RLHFlow’s Llama3.1-

8B PRM), and shows that an all-Qwen verifier roster is not necessary for target-local mismatch. It does not establish verifier-family universality: other training distributions, generative pairwise judges, or unified-criteria reward models may shift the local-winner landscape. We also do not make a benchmark-level decontamination claim for verifier training data, so the result remains verifier-family-conditioned rather than class-universal.

Diagnostic interpretation. The four text features may partly encode style and formatting rather than verifier compatibility itself. Holdouts, parser controls, and metadata-only baselines reduce direct shortcut explanations, but they do not prove causality. Candidate-set coherence, problem difficulty, and extraction regularity remain plausible latent variables.

Training utility. The exploratory check in Section 5.3 shows no detectable SFT-utility improvement from selector-better traces in our setup. The intervals are wide, so this is an underpowered non-positive result, not evidence that selection can never help SFT.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. We used a large language model only for language editing. All technical content, experiments, analyses, references, and conclusions were verified by the authors.

References

- Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.
- Zhengyu Chen, Yudong Wang, Teng Xiao, Ruochen Zhou, Xuesheng Yang, Wei Wang, Zhifang Sui, and Jingang Wang. 2026. From mathematical reasoning to code: Generalization of process reward models in test-time scaling. 40(36):30368–30376.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2025. Learning how hard to think: Input-adaptive allocation of lm computation. In *International Conference on Learning Representations*, volume 2025, pages 102783–102802.
- Kenan Hasanaliyev, Silas Alberti, Jenny Hamer, Dheeraj Rajagopal, Kevin Robinson, Jasper Snoek, Victor Veitch, and Alexander Nicholas D’Amour. 2026. Expected reward prediction, with applications to model routing. *arXiv preprint arXiv:2603.20217*.
- Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*.
- Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*.
- Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2024. Lightweight reranking for language model generations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6960–6984.
- Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, and 1 others. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *arXiv preprint arXiv:2504.16828*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. 2024:39578–39601.
- Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and 1 others. 2025. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.
- Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

- Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. RL on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37:43000–43031.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. Rewarding progress: Scaling automated process verifiers for llm reasoning. 2025:60808–60838.
- Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. R-prm: Reasoning-driven process reward modeling. pages 13438–13451.
- Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, and 1 others. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. pages 25299–25346.
- Lin Sun, Chuang Liu, Xiaofeng Ma, Tao Yang, Weijia Lu, and Ning Wu. 2025. Freeprm: Training process reward models without ground truth process labels. *arXiv preprint arXiv:2506.03570*.
- Shubham Toshniwal, Ivan Sorokin, Aleksander Ficek, Ivan Moshkov, and Igor Gitman. 2025. Genselect: A generative approach to best-of-n. *arXiv preprint arXiv:2507.17797*.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. pages 9426–9439.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Xinle Wu and Yao Lu. 2025. Reward model routing in alignment. *arXiv preprint arXiv:2510.02850*.
- Wei Xiong, Hanning Zhang, Nan Jiang, and Tong Zhang. 2024. An implementation of generative prm. <https://github.com/RLHFlow/RLHF-Reward-Modeling>.
- Zhenghao Xu, Qin Lu, Qingru Zhang, Liang Qiu, Ilgee Hong, Changlong Yu, Wenlin Yao, Yao Liu, Haoming Jiang, Lihong Li, and 1 others. 2026. Ask a strong llm judge when your reward model is uncertain. *Advances in Neural Information Processing Systems*, 38:74639–74664.
- Haozhen Zhang, Tao Feng, and Jiaxuan You. 2026. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:141233–141265.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. pages 10495–10516.
- Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and 1 others. 2026. Genprm: Scaling test-time compute of process reward models via generative reasoning. 40(41):34932–34940.
- Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Processbench: Identifying process errors in mathematical reasoning. pages 1009–1024.

Appendix map. The appendix is organised as an evidence trail rather than a table dump. Appendix A gives the full matched verifier panel behind the main regret claim. Appendix B reports the cross-lineage verifier check. Appendices C–D cover scale, post-training, and decoding controls. Appendices E–F audit the four-feature diagnostic and cold-start router.

Notation and denominator map. The *first-pass panel* has 22 targets, of which 21 are binary PRM/V4-routable after excluding one ORM-boundary target. *Wave A* is the 12-target second-pass extension; *Wave B* is its cross-template rerun under T1 (open chain-of-thought) and T2 (structured steps). The merged panel has 34 targets, 32 of them binary-routable after two ORM-boundary exclusions. The seed-48 direct audit contains the 9 first-pass non-PRM point-estimate winners. Bank 1 is the canonical seed-47 candidate draw and Bank 2 is the separately sampled seed-48 draw. K is candidates per problem, m is the labelled calibration-problem budget, and B is the unlabeled probe-problem budget.

The headline math panel uses 300 problems per target; the AIME 2026 check is a 30-problem problem-held-out slice within the same target-generator panel. Throughout the appendix, *point-estimate regime* means a winner under Bank 1, while McNemar, Holm, and bootstrap intervals mark stronger paired-statistical claims.

A Full 22-generator by 7-verifier evaluation panel

Reading. This matrix is the provenance table for the main-panel claim, not an additional experiment. Its purpose is to show that the fixed-PRM baseline is a strong choice globally while still leaving local regret: PRM-7B has the best mean over targets, yet the row-wise oracle moves away from PRM on 9/22 first-pass targets. The main text uses the aggregates; this table exposes the per-target cells so that the regime labels can be audited.

B Cross-lineage verifier check

We add the two non-Qwen verifier checkpoints requested during review to the frozen 34-target banks (300 problems per target, $K=64$). Skywork-Reward-V2-Llama-3.1-8B is scored as a sequence classifier (Liu et al., 2025). RLHFlow/Llama3.1-8B-PRM-Mistral-Data follows its official math-RM protocol, taking the mean of the step rewards produced by `prm_evaluate.py` (Xiong et al., 2024). No candidates, correctness labels, or original seven-verifier scores are regenerated for this comparison.

Fixed verifier	Bank 1 mean pick@1	Bank 2 mean pick@1
PRM-7B	43.5%	43.3%
Skywork-Reward-V2	42.7%	42.6%
RLHFlow Llama3.1-8B PRM	37.6%	37.3%

Table 8: Cross-lineage fixed-verifier means on both candidate banks. The original PRM-7B remains the point-estimate best fixed verifier; Skywork-V2 is second on both banks.

Check	Result
Bank 1 strict/tied leaders	17 PRM, 8 Skywork, 5 V4, 1 RLHFlow, 1 ORM; 2 additional targets tie for the lead
New strict local winners	9/34 (8 Skywork, 1 RLHFlow), plus the 2 ties above
Bank 2 Skywork recurrence	Of the 7 Bank 1 Skywork strict-win cells covered by the reported paired summary, 5 remain strict wins and a sixth becomes a tie (6/7 non-losses)
Expanded ora-cle/reference	Max-over-nine per-target oracle: +2.1 pp over fixed PRM on the 34-target Bank 1 panel; nine-pool diagnostic target-LOO: 29/34 (base rate 52.9%)

Table 9: Tie-aware cross-lineage summary and candidate-bank replication. The oracle is a hindsight reference and retains the same max-selection caveat as the original max-over-seven oracle.

These results rule out an all-Qwen verifier roster as a necessary condition for target-local mismatch, but do not establish universality over verifier families or training distributions. In particular, RLHFlow is below PRM-7B on 31/34 targets and has one strict local win, on Mistral-Nemo, which is consistent with rather than proof of a training-distribution match.

Family	Target	Param	n	PRM-7B	V4-2B	V4-4B	V4-9B	ORM-2B	ORM-4B	ORM-9B	Oracle v	Oracle	PRM regret
Qwen3.5	qwen35_08b_base	0.8B-Base	300	32.67	26.33	27.33	27.00	17.33	21.67	21.67	PRM-7B	32.67	+0.00
Qwen3.5	qwen35_2b_base	2B-Base	300	41.67	39.33	39.33	39.67	34.33	37.00	42.33	ORM-9B	42.33	+0.66
Qwen3.5	qwen35_4b_base	4B-Base	300	51.67	57.67	55.67	54.33	49.67	54.33	54.00	V4-2B	57.67	+6.00
Qwen3.5	qwen35_9b_base	9B-Base	300	50.67	59.00	57.00	55.00	51.67	52.67	54.33	V4-2B	59.00	+8.33
Qwen3.5	qwen35_08b	0.8B-Instruct	300	34.00	22.33	21.67	24.00	15.33	19.00	19.33	PRM-7B	34.00	+0.00
Qwen3	qwen3_06b	0.6B	300	38.00	35.33	37.67	38.67	32.33	36.33	35.67	V4-9B	38.67	+0.67
Qwen3	qwen3_17b	1.7B	300	47.67	52.67	54.67	50.33	45.33	49.33	51.00	V4-4B	54.67	+7.00
Qwen3	qwen3_4b	4B	300	51.67	56.33	56.00	54.33	52.00	52.00	54.33	V4-2B	56.33	+4.66
Qwen3	qwen3_4b_math	4B-Math	300	48.00	55.00	52.67	51.00	48.33	47.67	50.67	V4-2B	55.00	+7.00
Qwen3	qwen3_8b	8B	300	49.67	44.33	45.33	45.33	44.67	43.00	42.33	PRM-7B	49.67	+0.00
Qwen3	qwen3_32b	32B	300	50.67	42.00	41.67	42.33	40.33	42.00	39.00	PRM-7B	50.67	+0.00
Phi-4	phi4_mini	mini	300	36.00	14.67	18.33	16.67	17.33	15.00	10.00	PRM-7B	36.00	+0.00
Phi-4	phi4_mini_reasoning	mini-reasoning	300	50.33	53.00	51.33	54.33	48.00	50.33	53.33	V4-9B	54.33	+4.00
Llama-3.2	llama32_1b	1B	300	13.33	6.67	7.00	8.33	6.00	3.67	7.33	PRM-7B	13.33	+0.00
SmolLM3	smollm3_3b	3B	300	47.67	31.33	36.00	37.33	32.33	30.67	31.00	PRM-7B	47.67	+0.00
Falcon-H1	falcon_h1_7b	7B	300	41.67	38.00	39.33	41.00	36.67	35.33	39.33	PRM-7B	41.67	+0.00
Granite-3.1	granite_31_8b	8B	300	29.67	13.00	14.67	14.33	17.00	12.67	14.00	PRM-7B	29.67	+0.00
DSR1-Distill	deepseek_r1_qwen_7b	Qwen-7B	300	53.67	55.67	56.00	55.00	41.67	47.33	49.00	V4-4B	56.00	+2.33
InternLM3	internlm3_8b	8B	300	41.33	25.67	25.33	32.00	24.67	22.00	25.00	PRM-7B	41.33	+0.00
OLMo-2	olmo2_13b	13B	300	30.33	18.33	17.00	20.00	15.00	14.67	15.67	PRM-7B	30.33	+0.00
Hermes-4	hermes4_14b	14B	300	50.33	34.33	36.33	40.33	33.67	33.67	35.33	PRM-7B	50.33	+0.00
QwQ	qwq_32b	32B	300	55.67	52.33	55.00	54.67	44.67	49.67	50.67	PRM-7B	55.67	+0.00

Table 7: **Full 22-generator $\times$ 7-verifier rerank pick@1 matrix** on the heldout math eval mix ($n=300$ problems, $K=64$ candidates per problem, $T=0.7$, seed 47). Per-cell values are top-1 pick@1 in percent; the per-row oracle cell is bolded. Family breakdown: Qwen3.5-Base (4), Qwen3.5-Instruct (1), Qwen3 (6), Phi-4-mini (2), and 9 single-target families (Llama-3.2, SmolLM3, Falcon-H1, Granite-3.1, DSR1-Distill, InternLM3, OLMo-2, Hermes-4, QwQ). Aggregates: fixed PRM-7B mean = 43.02%; per-target oracle mean = 44.86%; mean PRM regret = +1.85 pp; PRM-7B is the strongest single fixed verifier across all 22 targets (next best fixed: V4-9B at 38.91%). Oracle winner distribution: PRM-7B 13/22; V4-class 8/22; ORM-class 1/22.

C Per-family scale-sweep tables

Generator	n	Winner	Δ over runner-up	McNemar p
qwen3.5-0.8B-Base	300	PRM-7B	+5.33 pp	0.018
qwen3.5-2B-Base	300	ORM-9B	+0.67 pp	0.860 (boundary)
qwen3.5-4B-Base	300	V4-2B	+6.00 pp vs PRM (within V4 +2.00 pp)	0.114 (V4-class)
qwen3.5-9B-Base	300	V4-2B	+8.33 pp vs PRM (within V4 +2.00 pp)	0.181 (V4-class)

Table 10: **Qwen3.5-Base 4-target scale sweep (Section 3)**. Empirical winner verifier, margin over runner-up, and McNemar p . The 0.8B target is PRM-class with $p=0.018$; 4B–9B targets are V4-class; 2B is a ± 0.7 pp regime boundary.

Generator	n	Winner	Δ over runner-up	McNemar p
qwen3-0.6B	300	V4-9B	+0.67 pp	0.855 (boundary)
qwen3-1.7B	300	V4-4B	+7.00 pp vs PRM	0.181
qwen3-4B	300	V4-2B	+4.67 pp vs PRM	1.000 (V4-class)
qwen3-4B-Math	300	V4-2B	+7.00 pp vs PRM	0.096
qwen3-8B	300	PRM-7B	+4.33 pp	0.061 (borderline)
qwen3-32B	300	PRM-7B	+8.33 pp	0.0015

Table 11: **Qwen3 6-target scale sweep (Section 3)**. Direction is opposite to Qwen3.5-Base.

Reading. The three scale sweeps are intentionally not pooled into a single size law. Qwen3.5-Base moves from PRM to V4 as scale grows, Qwen3 moves in the opposite direction, and Phi stays PRM-class while its margin shrinks. This is why the main paper treats generator family and candidate distribution as part of the target, rather than fitting a single monotone scaling explanation.

Generator	Params	Winner	Δ over runner-up (V4-9B)	McNemar p
Phi-3-mini-4k-instruct	3.8B	PRM-7B	+13.67 pp	$< 10^{-4}$
Phi-3-medium-4k-instruct	14B	PRM-7B	+9.67 pp	$< 10^{-3}$
Phi-4 (full)	14B	PRM-7B	+5.33 pp	0.024

Table 12: **Phi 3-bank stress test (Section 3)**. All three Phi targets are PRM-class with $p < 0.05$. Margin shrinks with scale (+13.67 $\rightarrow$ +5.33 pp) but no crossover within the 3.8–14B range. **Phi-3-mini $\rightarrow$ Phi-3-medium** is a clean scale step within one model generation; **Phi-4** (14B) is the same size as Phi-3-medium but a different generation and serves as a generation-control, not a continuation of Phi-3’s scale trajectory.

One candidate-distribution hypothesis is that scale changes the two families differently. Qwen3.5-Base keeps its coarse output format similar while answer-cluster entropy falls sharply through 4B, yielding more coherent banks on which outcome-style verifiers win. Qwen3 instead changes format between 4B and 8B: boxed-answer rate rises, length falls, and entropy recovers as the winner returns to PRM. Similar pass@64 at the two large Qwen3 endpoints (70.7% versus 71.0%) weakens a simple ability-only explanation. This account is correlational, not causal, and the low-margin Qwen3-0.6B endpoint changes regime on Bank 2.

Target	Boxed	Geo.-mean length	Answer entropy	Regime
Qwen3.5-0.8B-Base	47%	~ 4.0k	1.54	PRM
Qwen3.5-2B-Base	49%	~ 4.3k	0.84	ORM (boundary)
Qwen3.5-4B-Base	53%	~ 5.7k	0.20	V4
Qwen3.5-9B-Base	55%	~ 6.0k	0.27	V4
Qwen3-0.6B	47%	~ 8.2k	0.67	V4
Qwen3-4B	51%	~ 7.7k	0.10	V4
Qwen3-8B	85%	~ 2.2k	0.98	PRM
Qwen3-32B	97%	~ 1.1k	1.22	PRM

Table 13: **Candidate-text feature trajectories for the Qwen3.5 and Qwen3 scale sweeps.** Boxed is the fraction of candidates with a boxed-answer marker; length is the geometric mean candidate length; entropy is computed over extracted-answer clusters.

Multiple-comparison correction across all 22 targets. We compute exact two-sided McNemar winner-vs-runner-up p -values for all 22 heldout targets in the panel (one test per target). With $m=22$ tests, family-wise error rate control is non-trivial, and the load-bearing claim of Section 3 is the family-level *cross-direction* pattern (Qwen3.5 forward, Qwen3 reverse, Phi stable) — not that every individual cell beats its runner-up at $p < 0.05$ uncorrected. After Holm-Bonferroni correction at $\alpha=0.05$, 8/22 tests remain significant; Benjamini-Hochberg at the same level holds 10/22 (Wilson interval covering the 10 uncorrected hits). For deployment, Table 14 adds the direct audit against fixed PRM for the nine non-PRM point-estimate winners.

Target	Winner	Δ vs. PRM	b	c	Holm p
qwen3.5-9B-Base	V4-2B	+8.33	30	5	0.0002
qwen3-4B-Math	V4-2B	+7.00	27	6	0.0026
qwen3-1.7B	V4-4B	+7.00	28	7	0.0036
qwen3.5-4B-Base	V4-2B	+6.00	23	5	0.0055
Phi-4-mini-reasoning	V4-9B	+4.00	16	4	0.0591
qwen3-4B	V4-2B	+4.67	21	7	0.0591
DSR1-Distill-Qwen-7B	V4-4B	+2.33	25	18	1.0000
qwen3-0.6B	V4-9B	+0.67	16	14	1.0000
qwen3.5-2B-Base	ORM-9B	+0.67	17	15	1.0000

Table 14: **Deployment-relevant winner-vs-PRM McNemar audit for the nine non-PRM point-estimate winners in the 22-target panel.** This complements the winner-vs-runner-up audit: for V4-class rows, the runner-up can be another V4 verifier, so direct comparison to fixed PRM is the relevant deployment test. Six of nine contrasts pass raw/BH $p < 0.05$; four pass Holm.

b : winner correct / PRM wrong; c : winner wrong / PRM correct.

Targeted seed-48 direct audit. To probe candidate-bank sensitivity without rerunning the full 34×7 panel, we regenerated the nine first-pass seed-47 non-PRM point-estimate winner targets with seed 48 and reranked each seed-47 winner directly against PRM-7B. This is narrower than a full local-winner recomputation: it asks whether

the original deployment contrast survives a second candidate bank.

Full seed-48 first-pass policy replay. After the targeted direct audit, we completed a full seed-48 rerank of the 22 first-pass targets with the same seven verifiers and recomputed the deployment rows without overwriting the canonical seed-47 artifacts. This replay uses the same target set, problem set, $K=64$, and policy code as Table 1, but a new candidate-bank draw. It is a second-bank robustness check, not a seed-level CI.

Target	seed-47 winner	Δ_{47}	Δ_{48}	b	c	Holm p	BH q
qwen3-4B-Math	V4-2B	+7.00	+8.33	28	3	0.00004	0.00004
qwen3-1.7B	V4-4B	+7.00	+4.67	20	6	0.0748	0.0406
qwen3-0.6B	V4-9B	+0.67	-5.00	9	24	0.0947	0.0406
qwen3.5-4B-Base	V4-2B	+6.00	+3.67	17	6	0.2081	0.0643
qwen3-4B	V4-2B	+4.67	+4.00	20	8	0.2081	0.0643
Phi-4-mini-reasoning	V4-9B	+4.00	+2.67	13	5	0.3850	0.1444
qwen3.5-9B-Base	V4-2B	+8.33	+3.00	19	10	0.4081	0.1749
qwen3.5-2B-Base	ORM-9B	+0.67	-2.33	11	18	0.5299	0.2980
DSR1-Distill-Qwen-7B	V4-4B	+2.33	+1.67	24	19	0.5424	0.5424

Table 15: **Seed-48 direct replay of seed-47 non-PRM winners against PRM.** Δ_{47} is the seed-47 winner-vs-PRM margin from Table 14; Δ_{48} reranks the same seed-47 winner on a separately sampled seed-48 candidate bank. Seven of nine signs preserve, but directional support weakens: 4/9 positive and raw-significant, 1/9 positive Holm, 2/9 positive BH. The qwen3-0.6B boundary row flips significantly toward PRM.

b : seed-47 winner correct / PRM wrong on seed 48; c : seed-47 winner wrong / PRM correct on seed 48.

Policy	Δ	CI95 Worst	W/T/L
Gated router	+0.92	[-0.22, +2.19]	-5.33 5/15/1
PRM+V4-2B fusion	+1.95	[+0.73, +3.17]	-4.00 16/1/5
PVA-Adapt PRM-only	+1.93	[+0.84, +3.04]	-2.12 15/0/7
PVA-Adapt full	+1.72	[+0.65, +2.83]	-1.96 17/0/5
Per-target verifier oracle	+1.52	[+0.56, +2.64]	+0.00 7/15/0

Table 16: **Seed-48 replay of the first-pass deployment rows.** Mean deltas are vs. fixed PRM-7B on the 22 first-pass targets. Winner classes on this seed are 15/6/1 PRM/V4/ORM. Scored and labelled rows remain positive; the no-label router is weaker and its CI crosses zero.

The seed-48 replay uses the same policy definitions as Table 1.

D Post-training controls and decoding-shift controls

Pair A	Pair B	A winner / regime	B winner / regime
Qwen3.5-0.8B-instruct	Qwen3.5-0.8B-Base	PRM-7B / PRM	PRM-7B / PRM
Qwen3-4B (chat)	Qwen3-4B-Math	V4-2B / V4	V4-2B / V4
Phi-4-mini (instruct, short CoT, reasoning high \boxed{ })	Phi-4-mini (instruct, long CoT, reasoning low \boxed{ })	PRM-7B / PRM	V4-class / V4 (flips)

Table 17: **Within-architecture light post-training pairs preserve regime; reasoning-style post-training that substantially changes candidate format CAN flip regime (Section 3).**

Decoding-shift controls. On the Qwen3.5-Base 4-size sweep, we additionally generated candidate banks at $T \in \{0.3, 1.0\}$ (same $K=64$) and $K=16$ (same $T=0.7$) and reranked them with the 4-verifier core. Three of the four target sizes preserve their winner regime across these configurations. The exception is the low-margin Qwen3.5-2B-Base ORM-boundary target (0.67 pp at the canonical setting), which switches to PRM at both $T=0.3$ and $T=1.0$; other margin changes are at most 3 pp.

We then expand the $T=1.0$ check to all 22 first-pass targets ($n=300$, $K=64$) and all seven primary

verifiers.

$T=1.0$ comparison with canonical Bank 1 Preserved	
Winner class	19/22
Deployment split (PRM vs. non-PRM)	20/22
Exact winning verifier	18/22
Class when original margin ≥ 3.67 pp	11/11

Table 18: **Aggregate all-target decoding check at $T=1.0$.** Every winner-class change occurs at an original margin of at most 2.0 pp.

The three winner-class changes are Qwen3.5-2B-Base (0.66 pp original margin; the same boundary cell switches to PRM in the four-target control), Qwen3-0.6B (0.67 pp), and Qwen3-1.7B (2.0 pp; a within-outcome-class V4-to-ORM change). The fourth exact-verifier change remains within its original winner class.

Reading. The post-training and decoding controls separate two cases. Light within-architecture changes preserve the winner class, while a reasoning-style change that substantially alters candidate format can flip it; decoding changes can likewise move low-margin boundary cells. This supports the paper’s cautious framing: the regime is tied to the candidate distribution produced by a target generator under a prompt/template and decoding configuration, not to model size alone.

E Diagnostic predictor: feature definitions and LOO coefficients

Features (all computed from candidate texts only).

- `boxed_rate` — fraction of candidates containing a boxed-answer marker.
- `cand_len_log_mean` — mean of $\log_{10}(|\text{cand}| + 1)$ over K candidates.
- `cluster_entropy_mean` — Shannon entropy of the answer-cluster distribution (clusters defined by extracted final answer string equiva-

lence).

- `cluster_max_share_mean` — fraction of candidates falling into the largest answer cluster.

Feature	Coefficient
<code>cluster_entropy_mean</code>	−1.37
<code>cand_len_log_mean</code>	+0.82
<code>boxed_rate</code>	−0.73
<code>cluster_max_share</code>	+0.34

Table 19: L_2 -regularised logistic-regression coefficients on z-scored features (sign: − favours PRM regime, + favours V4 regime). Trained leave-one-target-out across the 21 PRM/V4-routable generators. LOO accuracy 19/21 = 90.5%. The two failures are qwen3.5-0.8B-instruct (predicted V4, true PRM; boundary case with thin margins) and QwQ-32B (predicted V4, true PRM; long-CoT reasoning model where scale interacts).

Leakage audit. Earlier drafts of the predictor exposed verifier-score-derived features (oracle—PRM gap, pass@ K , verifier disagreement, per-verifier score means) which are NOT candidate-text-only and were therefore excluded. The 4-feature no-leak predictor still achieves 90.5% LOO.

Feature/model ablation. We additionally rerun target-LOO probes after removing or isolating candidate-text features, and after swapping the linear probe for simple nonlinear classifiers. The purpose is diagnostic rather than model selection: with only 21 routable targets, nonlinear rows are treated as audit-only evidence. The two most obvious output-style features are not sufficient explanations of the result: removing boxed-answer rate or removing candidate length does not reduce target-LOO accuracy, while boxed-answer rate alone is much weaker.

Setting	Probe	Features used	Correct/ n	Misses
all features	logistic	boxed + length + entropy + max-share	19/21	qwen3.5-0.8B; QwQ-32B
drop boxed-rate	logistic	length + entropy + max-share	20/21	QwQ-32B
drop length	logistic	boxed + entropy + max-share	20/21	QwQ-32B
only boxed-rate	logistic	boxed	15/21	six targets
only length	logistic	length	18/21	qwen3.5-0.8B; qwen3.5-0.8B-Base; QwQ-32B
only cluster entropy	logistic	entropy	19/21	qwen3-0.6B; QwQ-32B
all features	kNN-3	boxed + length + entropy + max-share	19/21	qwen3.5-0.8B; QwQ-32B
all features	random forest	boxed + length + entropy + max-share	20/21	QwQ-32B
all features	GBDT	boxed + length + entropy + max-share	21/21	none

Table 20: **Diagnostic predictor ablation under leave-one-target-out.** All rows use the same 21 PRM/V4-routable targets as Table 19. Nonlinear probes are audit-only because n is small.

Permutation and coefficient-stability audit. Because the binary diagnostic has only 21 target units, we audit stability along three axes: a 10,000-trial label permutation null, a 10,000-resample bootstrap over the 21 targets, and an adversarial-flip sensitivity sweep on the least-confident predictions.

Quantity	Value
Observed target-LOO accuracy	19/21
10,000-bootstrap 95% CI on LOO	[18, 21]/21 (median 20)
10,000-permutation $p(\text{correct} \geq 19)$	0.0003
Permutation null max / 99th / 95th percentile	20 / 16 / 15 correct
Adversarial flip: $k=5$ least-confident flipped	17/21 remaining
Adversarial flip: $k=6$ flipped	14/21 (just below $\alpha=0.05$ binomial)
Coefficient sign (positive-rate over 10,000 bootstrap fits)	
boxed_rate (sign in observed fit: −)	100.0% negative
cand_len_log_mean (+)	100.0% positive
cluster_entropy_mean (−)	100.0% negative
cluster_max_share_mean (+)	82.2% positive
Coarse leave-family-out accuracy (all families)	18/21

Table 21: **Stability audit for the 4-feature diagnostic** (10,000 bootstrap + permutation; class-balance-preserving label permutation).

Within-bank B/K subsampling. The main cold-start router uses $B=300$ unlabeled problems and $K=64$ candidates per problem. To separate diagnostic stability from new candidate-bank variance, we also subsample the fixed seed-47 candidate bank: for each target, draw B problems and K candidates per problem, recompute the four text features, and rerun target-LOO classification. Labels remain the original full-bank point-estimate PRM/V4 regimes, so this is not a new-generation robustness test and does not by itself validate a smaller-budget deployment router.

B problems	$K=16$	$K=32$	$K=64$
50	89.33 ± 2.45	91.10 ± 2.10	92.81 ± 2.39
100	90.14 ± 1.40	91.10 ± 1.61	92.81 ± 2.39
150	90.24 ± 1.04	90.71 ± 1.04	92.95 ± 2.39
300	90.48 ± 0.00	90.48 ± 0.00	90.48 ± 0.00

Table 22: **Within-bank diagnostic stability under B/K subsampling.** Entries are mean target-LOO accuracy ± s.d. over 100 resamples on the 21 routable first-pass targets.

Why we do not use scale-slope inference. The scale sweeps above are descriptive. Qwen3.5-Base has only three non-boundary scale points and Qwen3 has six, so fitting a logistic slope to regime labels would be high-variance and near-separable. We therefore use the sweeps only as a qualitative check against a single monotone size explanation.

Parser-level style controls. Table 20 removes or isolates feature columns but does not change how features are parsed. We therefore add a stricter parser-level diagnostic: strip explicit `\boxed{ }` wrappers and common final-answer boilerplate before measuring length, and recompute answer-cluster features with a box-agnostic final-answer parser. This is not as strong as regenerating every target under a prompt-normalized template, but it directly tests whether the reported probe collapses once the most explicit boxed-answer cue is removed.

Control	Features used	Correct/n	Misses
format-only baseline	boxed rate + raw length	18/21	qwen3.5-0.8B; qwen3.5-0.8B-Base; QwQ-32B
style-stripped length only	stripped length	18/21	qwen3.5-0.8B; qwen3.5-0.8B-Base; QwQ-32B
boxed parser clusters only	boxed parser entropy + max-share	20/21	QwQ-32B
box-agnostic clusters only	box-agnostic entropy + max-share	11/21	eight targets
style-stripped length + box-agnostic clusters	stripped length + box-agnostic clusters	20/21	QwQ-32B
style-stripped, no boxed rate	stripped length + box-agnostic clusters + extraction coverage	19/21	Falcon-H1-7B; QwQ-32B

Table 23: **Parser-level style controls for the candidate-text diagnostic.** All rows are target-LOO on the same 21 PRM/V4-routable targets. The result is intentionally reported as mixed: box-agnostic clusters alone are weak, while style-stripped features that exclude boxed-rate remain predictive.

Family- and architecture-level holdout. A natural concern with the 90.5% target-LOO number is that the 21 routable generators are not independent: 5 Qwen3.5 sizes, 6 Qwen3 sizes, and 2 Phi-4-mini variants share architecture and post-training pipelines, so leave-one-target-out keeps same-family neighbours in the training fold. To answer this directly, we rerun the same 4-feature L_2 -regularised probe under two strictly stronger validation protocols.

Protocol	Held-out group	Correct/n	Accuracy	Wilson 95% CI
target-LOO	(each routable target in turn)	19/21	90.5%	[71.1, 97.3]
family-LOO	leave-Qwen3.5-out (4 targets)	3/4	75.0%	[30.1, 95.4]
	leave-Qwen3-out (6 targets)	6/6	100.0%	[61.0, 100.0]
	leave-Phi-4-mini-out (2 targets)	2/2	100.0%	[34.2, 100.0]
	pooled (multi-family)	11/12	91.7%	[64.6, 98.5]
architecture-LOO	leave-Qwen-arch-out (11 targets)	10/11	90.9%	[62.3, 98.4]
	leave-Phi-arch-out (2 targets)	2/2	100.0%	[34.2, 100.0]
	leave-rest-out (8 targets)	8/8	100.0%	[67.6, 100.0]
	pooled (all archs)	20/21	95.2%	[77.3, 99.2]

Table 24: **Family- and architecture-level holdout for the 4-feature predictor.** The two stricter protocols hold out an entire family group (or architecture cluster) rather than a single target, so no same-family neighbour ever leaks into the training fold. Pooled family-LOO and architecture-LOO accuracies are at or above the target-LOO number; Wilson 95% CIs are wider because n is smaller. The family-LOO miss is qwen3.5-0.8B-Instruct; the architecture-LOO miss is QwQ-32B. Both are the target-LOO boundary cases flagged in Table 19.

Reading. The limited holdout checks reduce the concern that the 4-feature candidate-text probe only memorises same-family neighbours: holding out the entire Qwen3 family (6 of 21 targets) keeps 6/6 predictions correct, and holding out the entire Qwen-architecture cluster (11 of 21 targets) keeps 10/11 correct, with QwQ-32B as the only miss. This is evidence against a pure family-cluster shortcut, not a proof of a causal candidate-distribution mechanism. A 10000-resample bootstrap on the family-LOO pooled accuracy gives a 95% CI of [75.0, 100.0]% (median 91.7%); the wider CIs are an inherent limitation of $n \leq 21$ generators rather than a sign of brittleness.

Why the regimes emerge: failure-mode breakdown. A separate concern is whether the PRM $\rightarrow$

V4 $\rightarrow$ ORM regime structure is just a label fit on the diagnostic, or whether it has a structural correctness-pattern signature. For each (target, alt-verifier) cell in the first-pass 22-target panel we compute the joint pick correctness of (PRM-7B, alt-verifier) on the same $K=64$ candidate banks and split into four modes: $BR\%$ both right, $P_{\text{only}}\%$ PRM-right / alt-wrong, $V_{\text{only}}\%$ alt-right / PRM-wrong, $BW\%$ both wrong. The gain $\Delta_{\text{alt vs. PRM}} = V_{\text{only}} - P_{\text{only}}$.

Regime (n)	Alt-verifier	$BR\%$	$P_{\text{only}}\%$	$V_{\text{only}}\%$	$BW\%$
PRM ($n=13$)	V4-2B	25.15	14.31	3.26	57.28
	V4-4B	26.15	13.31	3.46	57.08
	V4-9B	27.41	12.05	3.62	56.92
	ORM-9B	24.03	15.44	2.95	57.59
V4 ($n=8$)	V4-2B	45.42	3.54	7.67	43.38
	V4-4B	45.71	3.25	6.92	44.12
	V4-9B	46.04	2.92	5.58	45.46
	ORM-9B	44.38	4.58	5.92	45.12

Table 25: **Failure-mode breakdown by empirical regime,** averaged across the targets in each regime, all $K=64$ max-new 4096. The $P_{\text{only}}/V_{\text{only}}$ ratio reverses cleanly between regimes: in PRM-regime targets, PRM-only-right is $\approx 4\times$ alt-only-right; in V4-regime targets, alt-only-right is $\approx 2\times$ PRM-only-right. Concomitant rise in $BR\%$ (25 $\rightarrow$ 46%) and drop in $BW\%$ (57 $\rightarrow$ 44%) show that V4-regime banks have more total agreement signal, not less.

Reading. The regime structure is not just a property of the diagnostic features — it is also a structural property of the joint (PRM, alt-verifier) correctness pattern on the same candidates. In PRM-regime targets the candidate distribution is sparse on correct answers ($BR = 25\%$, $BW = 57\%$); PRM’s step-level penalty catches the rare correct candidate that V4/ORM outcome-rerankers miss ($P_{\text{only}} = 12\text{--}15\%$ vs. $V_{\text{only}} = 3\%$). In V4-regime targets the candidate distribution has a coherent right-answer cluster that any process-aware verifier latches onto ($BR = 46\%$, $BW = 44\%$); the residual disagreement disproportionately favours V4-style outcome reasoning over PRM-style step grading ($V_{\text{only}} = 6\text{--}8\%$ vs. $P_{\text{only}} = 3\%$). The diagnostic’s four candidate-text features (boxed rate, length, cluster entropy, max-cluster share) are the observable correlates of this candidate-distribution structure.

F Per-target cold-start router diagnostics (full table)

Target	True regime	$P(V4)$	gate	Deployed	Routed	Δ vs. PRM
qwen3.5-0.8B-instruct	PRM	0.506	0.70	PRM-7B	no	0.00
qwen3.5-0.8B-Base	PRM	0.262	0.75	PRM-7B	no	0.00
qwen3.5-4B-Base	V4	0.779	0.70	V4-2B	yes	+6.00
qwen3.5-9B-Base	V4	0.742	0.70	V4-2B	yes	+8.33
qwen3-0.6B	V4	0.704	0.65	V4-2B	yes	-2.67
qwen3-1.7B	V4	0.842	0.70	V4-2B	yes	+5.00
qwen3-4B	V4	0.860	0.70	V4-2B	yes	+4.67
qwen3-4B-Math	V4	0.862	0.70	V4-2B	yes	+7.00
qwen3-8B	PRM	0.221	0.75	PRM-7B	no	0.00
qwen3-32B	PRM	0.062	0.70	PRM-7B	no	0.00
Llama-3.2-1B-instruct	PRM	0.007	0.70	PRM-7B	no	0.00
SmolLM3-3B	PRM	0.025	0.70	PRM-7B	no	0.00
Phi-4-mini	PRM	0.002	0.70	PRM-7B	no	0.00
Phi-4-mini-reasoning	V4	0.831	0.70	V4-2B	yes	+2.67
Falcon-H1-7B	PRM	0.042	0.70	PRM-7B	no	0.00
Granite-3.1-8B	PRM	0.001	0.70	PRM-7B	no	0.00
DeepSeek-R1-Distill-Qwen-7B	V4	0.717	0.70	V4-2B	yes	+2.00
InternLM3-8B	PRM	0.013	0.70	PRM-7B	no	0.00
OLMo-2-13B	PRM	0.005	0.70	PRM-7B	no	0.00
Hermes-4-14B	PRM	0.031	0.70	PRM-7B	no	0.00
QwQ-32B	PRM	0.803	0.80	V4-2B	yes	-3.33
Aggregate						+1.41

Table 26: **Full 21-target diagnostic for the gated cold-start router (Section 4.2).** For each target: true regime, predicted $P(V4)$, nested-CV threshold, deployed verifier, routed (max probability above threshold) or fallback, Δ vs. fixed PRM. Aggregates: 7 strict wins (range +2.00 to +8.33 pp), 12 abstentions at $\Delta = 0$ by design, 2 losses (max observed -3.33 pp).

Failure-mode analysis. The two routed losses split by cause: **Qwen3-0.6B** ($\Delta = -2.67$ pp): router predicts regime correctly (V4-class — the per-target oracle verifier is V4-9B), but the canonical default V4-2B is suboptimal within the V4 class on this boundary-regime target. A within-class router (default to per-fold-best V4 rather than fixed V4-2B) could rescue this; LOO experiments confirm the aggregate is identical. **QwQ-32B** ($\Delta = -3.33$ pp): router mis-classifies (true regime is PRM, predicted V4 at $P_{V4} = 0.80$); long-CoT reasoning candidates fool the predictor at large scale despite gate = 0.80. Both observed losses are within ~ 3 pp magnitude.

G Routing-neighbour comparison

Paper	Routed object	Unit	Input features	Supervision	Deployment objective
Chen+23 (FrugalGPT)	LLM generators	per-query	prompt/cascade feat.	task labels	cost-quality cascade
Ong+24 (RouteLLM)	LLM generators	per-query	prompt feat.	preference data	cost-quality routing
Wu+25 (RM Routing)	reward models	per-query	query emb.	offline prefs	RLHF reward
Xu+25 (Judge Uncertain)	RM $\rightarrow$ judge	per-decision	RM uncertainty	RM-conf. calib.	alignment
Hasanaliyev+26 (ERP)	LLM generators	per-prompt	prompt feat.	expected reward	cost-adj. reward
Jidkrattum+25 (UniRoute)	LLMs	per-query	LLM feat. vectors	expected quality	cost-quality Pareto
Damani+25 (Hard-to-Think)	compute	per-prompt	difficulty estimate	task reward	accuracy under budget
Zhang+25 (Router-R1)	LLM calls	multi-round	model descriptors + context	RL rewards	cost-aware aggregation
Ours (gated cold-start)	verifiers	per-target	candidate-dist. stats	none at deploy	rerank-pick@1 at fixed cost

Table 27: **Axis-by-axis comparison** of our gated cold-start router against the closest routing neighbours. The main difference is the routed object and routing unit: our router allocates verifier identity for a target generator’s candidate distribution, using candidate-text statistics and no deployment labels.

H PVA-Adapt regularisation sweep

C	Δ vs. PRM	CI95
0.03 (strong L_2)	+2.62 pp	[+1.59, +3.61]
0.10 (canonical)	+2.14 pp	[+1.11, +3.16]
0.30	+1.92 pp	[+0.88, +2.96]
1.00 (weak L_2)	+1.72 pp	[+0.69, +2.75]

Table 28: **PVA-Adapt regularisation sweep at $m=75$** (20 seeds, paired-bootstrap CI95). Headline canonical is $C=0.1$ (+2.14 pp); strong L_2 ($C=0.03$) gives +2.62 pp as a tuned-sensitivity reference.

Per-family decomposition (50 seeds, $m=75$, $C=0.1$) is dominated by Qwen3.5-Base (+3.63 pp Adapt, 4 targets), Qwen3 (+3.55 pp Adapt, 6 targets), and DSR1-distill (+5.42 pp, single target with large class margin). Stable-PRM-favouring single-target families (SmolLM, OLMo, Hermes, Granite, Llama-3, InternLM) net 0 to +1.22 pp Adapt with zero oracle gain.

I PVA-Adapt boundary rescue on the 10-target scale sweep

Target	Regime	Δ vs. PRM	Δ vs. per-target oracle
qwen3.5-0.8B-Base	PRM	+2.01 pp	+2.01
qwen3.5-2B-Base	boundary (ORM)	+4.19 pp	+3.77 (rescue)
qwen3.5-4B-Base	V4	+3.42 pp	-2.71
qwen3.5-9B-Base	V4	+4.90 pp	-3.40
qwen3-0.6B	boundary (V4)	+5.38 pp	+4.69 (rescue)
qwen3-1.7B	V4	+5.61 pp	-1.31
qwen3-4B	V4	+2.97 pp	-1.67
qwen3-4B-Math	V4	+3.76 pp	-3.43
qwen3-8B	PRM	+1.95 pp	+1.95
qwen3-32B	PRM	+1.61 pp	+1.61

Table 29: **PVA-Adapt at $m=75$, $C=0.1$ on the 10 scale-sweep targets (Section 4.5).** PVA-Adapt beats fixed PRM on every target by +1.6 to +5.6 pp; on the two regime-boundary targets, PVA-Adapt’s candidate-level arbitration exceeds the *single-verifier* per-target oracle (qwen3.5-2B-Base +3.77 pp; qwen3-0.6B +4.69 pp) via per-problem regime mixing.

J Per-problem oracle within the observed verifier pool

Reference	pick@1	Gap vs. fixed PRM
fixed PRM-7B	43.02%	—
per-target single-verifier oracle	44.86%	+1.85 pp
per-problem oracle (within-pool upper bound)	52.47%	+9.45 pp

Table 30: **Three reference points on the 22-generator panel.** The per-target *single-verifier* oracle commits to one verifier per generator (a candidate-level mixer like PVA-Adapt can exceed it). The per-problem oracle is a within-pool upper bound: it picks the best of 7 verifiers *per problem* with hindsight; no selector restricted to choosing among those seven verifier top picks can exceed it. Full-7 PVA-Adapt at $m=75$ recovers about 21% of this per-problem gap (+2.02 pp out of +9.45 pp); the remaining $\sim +7.4$ pp is the identified open problem of Section 6.

K Budget-aware PVA-Adapt: full Pareto

Several appendix rows are intentionally close in value but not the same estimate. Table 31 maps the headline-adjacent gains to their protocol so the body and appendix should not be read as conflicting.

Gain	Protocol	Seeds	Where / distinction
+2.35 pp	PRM+V4-2B z -fusion	—	no labels; fixed score fusion
+2.26 pp	PRM-only PVA-Adapt	50	Table 1 deployment headline
+2.02 pp	full-7 PVA-Adapt	50	Table 1 reference row
+2.27 pp	PRM-only PVA-Adapt	20	budget-aware subset sweep
+2.43 pp	PVA-Adapt with PRM+V4-2B features	20	feature-subset Pareto, not fusion
+2.14 pp	canonical/smaller subset sweeps	20	regularisation or 5-verifier reference

Table 31: **Provenance map for headline-adjacent deployment gains.** All rows report Δ vs. fixed PRM-7B on the 22-target panel at $m=75$ when labels are used; score fusion uses no labels. The 20-seed rows are subset sweeps, not replacements for the 50-seed Table 1 headlines.

Variant	$ V $	Test cost	Δ vs. PRM	Observation
PRM-only PVA-Adapt	1	1×	+2.27 pp	fixed-PRM test cost
PRM + V4-2B	2	1.3×	+2.43 pp	—
PRM + V4-9B	2	2.3×	+2.58 pp	best efficient point
PRM + V4-2B + ORM-2B	3	1.6×	+2.24 pp	—
PRM + V4-9B + ORM-9B	3	3.6×	+2.49 pp	—
PRM + all V4 (2B/4B/9B)	4	3.2×	+2.41 pp	—
5 verifiers (drop largest each class)	5	2.8×	+2.14 pp	—
all 7 (full PVA-Adapt)	7	5.4×	+2.21 pp	5.4× reference

Table 32: **Budget-aware PVA-Adapt Pareto at $m=75$ (20 seeds, 22 targets).** Test cost is relative verifier forward-pass cost per test candidate (parameter-count weighted; PRM-7B normalised to 1). The headline gain is essentially unchanged across the Pareto (+2.14 to +2.58 pp) as long as PRM-7B is present and the four label-free candidate features are available — PVA-Adapt’s gain is driven by target-local adaptation + candidate features, not multi-verifier ensembling.

A 50-seed rerun of the PRM-only row gives +2.26 pp (CI95 [+1.11, +3.42]) and is the value reported in Table 1.

L AIME 2026 per-family decomposition

The AIME 2026 result (Table 4) decomposes by candidate family. Large wins on V4-favouring families: DSR1-Distill +13.47 pp, QwQ +13.00 pp, Qwen3.5-Base +6.88 pp, Phi-4 +4.60 pp. Single-target small losses on the 3 stable-PRM families: SmolLM -3.93 pp, Qwen3.5-instruct -2.73 pp, Hermes -1.20 pp. The pattern is identical to the full-test-set decomposition (Section H), consistent with the regime predictions of the allocation pattern.

M Label-free baselines: SC and answer-cluster arbiters

Method	Mean pick@1	Δ vs. PRM	W/T/L
SC vote (mode answer)	41.03%	-1.98 pp	9/0/13
SC vote + PRM tiebreak	41.47%	-1.55 pp	10/1/11
cluster + PRM top-3	43.91%	+0.89 pp	12/2/8
fixed PRM-7B (reference)	43.02%	0.00	—
fixed V4-9B	38.91%	-4.11 pp	8/0/14

Table 33: **Label-free baselines on the 22-target held-out panel.** Self-consistency (SC) vote and answer-cluster arbiters compared to fixed PRM-7B and fixed V4-9B. Wins / Ties / Losses count targets where the method strictly beats / equals / loses to fixed PRM-7B at the per-target level. Plain SC vote underperforms; PRM-arbitrated cluster top-3 recovers a small mean gain. The single-verifier fixed-V4-9B alternative loses to fixed PRM-7B by -4.11 pp mean, confirming PRM-7B is the best global single-verifier choice (motivating the regret framing in Section 3).

Reading. Self-consistency over answer clusters (Wang et al., 2022) alone does not catch up to fixed PRM-7B on this panel (-1.98 pp); answer-cluster arbitration with PRM as tie-breaker is modestly above SC alone but still below fixed PRM. The cluster-+-PRM-top-3 arbiter — which restricts the verifier’s argmax to candidates drawn from the three most populated answer clusters — yields a small but real +0.89 pp gain on 12/22 targets, suggesting that combining cluster structure with a process verifier does carry signal. None of these label-free baselines closes the +1.85 pp gap to the per-target oracle, and the strongest single fixed verifier across the seven is PRM-7B itself (no alternative single verifier beats it on the mean). Routable allocation methods (Sections 4.3 and 4.4) recover +1.12 to +2.21 pp over fixed PRM-7B.

Method	Cost	Mean Δ	CI95	Max loss
metadata size/style router	1.0×	-0.62 pp	[-1.56, 0.00]	-6.66 pp
metadata family-majority router	1.0×	-1.87 pp	[-4.67, +0.46]	-21.33 pp
PRM+V4-2B rank fusion	1.3×	+0.70 pp	[-0.94, +2.12]	-8.67 pp
PRM+V4-2B z -fusion	1.3×	+2.35 pp	[+1.09, +3.53]	-3.33 pp
PRM+V4-9B z -fusion	2.3×	+2.11 pp	[+1.32, +2.85]	-1.67 pp
all-7 z -fusion	5.4×	+0.38 pp	[-1.62, +2.27]	-8.00 pp

Table 34: **Metadata and score-fusion deployment baselines.** Metadata routers use leave-one-target-out training over known generator metadata only; score-fusion baselines use verifier scores at test time but no target labels. CI95 is a target bootstrap over mean Δ vs. fixed PRM.

Reading. The metadata-only routers are intentionally simple controls for the claim that candidate texts add information beyond known generator identity, size, and post-training tags; both are negative. Score fusion spends extra verifier compute on every problem rather than choosing one verifier per target: PRM+V4-2B z -fusion is the strongest no-label extra-compute control (+2.35 pp at 1.3×), while the cold-start router is a one-verifier policy.

Protocol	Metadata router	Correct/ n	Mean Δ	Max loss	V4 routes
target-LOO	size/training tags	13/21	-0.67 pp	-11.67 pp	6
target-LOO	family + size/training	11/21	-2.10 pp	-21.33 pp	8
family-LOO	size/training tags	12/21	-0.57 pp	-11.67 pp	3
family-LOO	family + size/training	11/21	-0.86 pp	-11.67 pp	2

Table 35: **Metadata-only router baselines under target- and family-LOO.** Features include parameter scale, training/style tags, and a coarse architecture flag; the family+size variant additionally exposes family ID. Both variants are much weaker than the candidate-text probe and have negative deltas.

N Exploratory SFT-utility seed replication and 4B student

Selector	pass@1	Δ vs. PRM	CI95
PRM (baseline)	46.67%	—	—
PVA-Calibrate (seed 42)	48.10%	+1.43 pp	[-2.86, +5.71]
PVA-Calibrate (3-seed pooled)	—	-1.43 pp	[-4.29, +1.43]
PVA-Adapt PRM-only	45.71%	-0.95 pp	[-6.19, +4.29]
PVA-Adapt full-7v	40.48%	-6.19 pp	[-11.90, -0.95]
oracle-selected	41.90%	-4.76 pp	[-10.48, +0.95]

Table 36: **SFT-utility on a Qwen3-8B Instruct student ($n=210$ non-floor eval; GSM8K-100, MATH-500-30, OB-hard-30, AIME 2024-10, 2025-10, 2026-30).** Selection-quality gains (Tables 3 and 4) do not produce detectable positive SFT utility; the oracle-selected SFT student has a negative point estimate relative to the PRM-selected student, with CI crossing zero.

Seed replication. The seed-42 PVA-Calibrate SFT result is the positive outlier: seed 43 is flat (0.00 pp) and seed 44 is negative (-5.71 pp), giving the 3-seed pooled estimate in Table 36 (-1.43 pp, CI95 crossing zero). The non-positive direction is also observed on a Qwen3-4B In-

struct student. This small study is exploratory and does not support a general negative SFT claim. One possible explanation is that selector-favoured chains differ in how easily the student can imitate them, but we do not identify that mechanism here. Rerank-pick@1 and student-imitation are different targets, related but not the same.

Wave E: 3-selector SFT on Qwen2.5-Math-7B-Instruct. Three selector-paired SFTs (PRM-7B, ORM-4B, V4-4B) on Qwen2.5-Math-7B-Instruct, $n=800$ traces each, 1 epoch single-seed raw transformers.Trainer; evaluated on the same 300-problem hard-math set (AIME 2024/30 + AIME 2025/30 + AIME 2026/30 + MATH-500/100 + OlympiadBench_math_hard/110, all at $K=64$, max-new 4096). The PRM-class selector ordering (PRM > ORM > V4) is direction-preserved but *none of the three deltas reach statistical significance*; pass@64 reverses the ordering (V4-SFT highest), so SFT-time and inference-time interpretations of the same selector ordering diverge at the candidate-set vs. single-pick level.

Selector	Greedy	Δ vs. base	CI95	McNemar p	pass@64
Base (Qwen2.5-Math-7B-Inst)	41.33%	—	—	—	67.00%
PRM-7B-selected SFT	43.00%	+1.67	pp [-1.67, +5.00]	0.4244	67.67%
ORM-4B-selected SFT	42.00%	+0.67	pp [-2.33, +3.67]	0.8238	67.67%
V4-4B-selected SFT	39.67%	-1.67	pp [-5.00, +1.67]	0.4421	69.33%

Table 37: **Wave E: math-specialized SFT base, 3 selectors.** Greedy pick@1 on 300 hard-math problems vs. base Qwen2.5-Math-7B-Instruct.

Wave E meta-PVA: 3×7 verifier-on-SFT’d-bank grid. We rerank each of the three SFT’d Qwen2.5-Math-7B variants with all 7 verifiers from the main panel. The greedy column shows the SFT’d model’s own no-rerank pick@1; remaining cells show the SFT’d model’s $K=64$ candidate bank reranked by the named verifier. PRM-7B is the decisive winner on *all three* SFT’d variants, with +6.33 to +8.66 pp over the SFT’d model’s greedy. The inference-time PRM-class winner regime for Qwen2.5-Math-7B-Instruct (the 12-target second-pass extension) is preserved through SFT regardless of which selector was used to choose the SFT training data.

SFT’d model	greedy	V4-2B	V4-4B	V4-9B	ORM-2B	ORM-4B	ORM-9B	PRM-7B
Qwen2.5-Math-7B + PRM-SFT	43.00	41.00	40.33	42.33	36.00	36.67	39.67	49.33
Qwen2.5-Math-7B + V4-SFT	39.67	40.33	41.67	43.67	37.33	36.67	40.00	48.33
Qwen2.5-Math-7B + ORM-SFT	42.00	39.33	40.00	41.67	35.67	38.00	—	49.00
Δ PRM-7B vs. greedy	—	—	—	—	—	—	—	+6.33 / +8.66 / +7.00

Table 38: **Wave E meta-PVA: 3×7 verifier-on-SFT’d-bank grid.** Each row is a Qwen2.5-Math-7B variant after 800-trace SFT with the named selector; each cell is pick@1 after reranking that variant’s $K=64$ candidates with the column verifier. All numeric cells are $n=300$; ORM-9B on ORM-SFT was not run and is excluded from row/column claims. PRM-7B wins on all three rows; observed ORM-class verifiers consistently underperform greedy.

O Full cost arithmetic

Test-time cost (per test candidate). With PRM-7B normalised to 1, the empirical verifier-cost breakdown is V4/ORM-2B ≈ 0.3 , -4B ≈ 0.6 , and -9B ≈ 1.3 . Full 7-verifier deployment is several-fold more expensive than PRM alone; PRM-only PVA-Adapt costs $1 \times$. The four label-free candidate features are $\mathcal{O}(K \log K)$ overhead, negligible relative to verifier forward passes.

Calibration-time cost. The m -probe budget consumes $m \cdot K \cdot |V|$ verifier forward passes (one per candidate per verifier) plus a single logistic-regression fit per target ($\mathcal{O}(mK \cdot 41^2)$ operations). At $m=75$, $K=64$, $|V|=7$ this is $\approx 33k$ forward passes per target — amortised over the deployment lifetime.

P Verifier provenance

The fixed-PRM baseline is the public Qwen2.5-Math-PRM-7B checkpoint. The V4 and ORM rows are one additional verifier lineage at three backbone sizes each; the main paper therefore treats them as representative families, not class-universal baselines.

V4 outcome verifier family. V4-2B/4B/9B are full reward-model fine-tunes of Qwen3.5-2B/4B/9B backbones. Each model uses the same Bradley–Terry ranking objective over 13,477 within-problem outcome pairs where the chosen trace has the correct final answer and the rejected trace is incorrect for the same problem/prefix branch. The per-size configurations use cutoff length 2048, learning rate 10^{-5} , bf16, and seed 42 (1 epoch for 2B, 2 epochs for 4B/9B).

ORM outcome verifier family. ORM-2B/4B/9B use the same Qwen3.5 backbones, optimizer settings, and pair count, but replace the within-problem rejected trace with a rejected trace shuffled across problems. Thus the chosen side remains a correct trace for the prompt while the rejected side

is wrong for a different prompt, making the ablation more outcome-only and less setwise. We use the same per-size configuration format as for the V4 verifier family.

Overlap and reproducibility artifacts. The V4/ORM training-pair pool is drawn from public math sources and is not claimed to be decontaminated against the public math evaluation mix. This is why the paper phrases the result as seed- and verifier-family-conditioned. The paper-level allocation analyses depend only on per-candidate verifier scores and correctness labels.

Q Wave A: 12-target second-pass per-target summary

The second pass is not a cherry-picked replacement for the first-pass panel. It adds held-out generators after the diagnostic was fixed and keeps the same K , prompt, and verifier pool. The main pattern persists but weakens: most added Qwen2.5/Phi targets remain PRM-class, while the three largest reasoning-tuned additions are non-PRM. This is why the main paper reports both the first-pass diagnostic fit and the combined-panel degradation. After recovering the previously omitted Yi-1.5-34B $\times$ PRM-7B cell, the corrected Wave A winner split is 8/3/1 PRM/V4/ORM, and the second-pass per-target-oracle gain is approximately +1.04 pp.

R Wave B: cross-template winner-class summary

The same 12 Wave A second-pass targets are reranked with the same 7-verifier panel under T0 canonical boxed-answer prompting, T1 open chain-of-thought prompting, and T2 structured step-by-step prompting. Winner class is preserved on 10/12 T1 targets and 10/11 evaluable T2 targets. Mistral-Nemo-12B is the only flip shared by both templates; the frozen diagnostic had predicted PRM for this target ($p_{V4} \approx 10^{-4}$), and both alternative templates move the observed winner from V4 to PRM, toward that prediction. This movement should not be attributed to the candidate-length feature. Yi-1.5-34B-Chat is PRM-class under corrected T0 scoring and is preserved, not a template-flip miss. Every observed flip has a T0 winner margin of at most 1.67 pp.

S Full PVA-Calibrate / PVA-Adapt m-curve

The compact m -curve is now reported in the main text (Table 3); this appendix keeps the surrounding robustness material only.

T Stress-test ledger for the four-feature diagnostic

The compact audit summary is in Section 4.1; detailed family/architecture holdouts and style controls are in Tables 21 and 24.