ADAPTER THICKETS: SPLITTING AN RLVR BUDGET BEATS CONCENTRATING IT

Anonymous authors
Paper under double-blind review

ABSTRACT

Majority voting over sampled completions is the workhorse of test-time scaling, and reinforcement learning with verifiable rewards (RLVR) is the workhorse for making each completion better. The standard pipeline composes the two: train one policy with RLVR, then sample it many times and vote. We show that this composition is lossy. A vote can only overturn mistakes that its voters do not share, and RLVR sharpens a policy so that its samples increasingly make the same mistakes. With every method drawing exactly 160 completions per problem, training a single LoRA adapter on the full RLVR budget raises single-sample accuracy on every model we test (1.5B–8B). Yet on three of four models it leaves the majority vote below that of the untrained base model, by up to 4.8 points. The damage builds during training: voter errors grow steadily more correlated, and the majority vote accuracy peaks early before falling by up to 7.0 points. The cause is concentration, not RLVR itself. We split the same data and training budget across K LoRA adapters, each trained on its own random disjoint shard, and call the result an *adapter thicket*. Thickets out-vote the fully trained adapter in all 16 (model, K) settings, and for $K \geq 4$ they stay within 0.8 points of the base model or above it. A single adapter stopped early, at a thicket member's step count, is a strong control that matches thickets for small K. For $K \geq 8$, thickets keep more of RLVR's single-sample gain and out-vote this control in six of eight settings. The cost of concentration also grows with the number of votes: from 16 to 160 votes, the thicket's lead over the fully trained adapter widens from 1.3 to 3.3 points. When the plan is to sample and vote, an RLVR budget is better spent broad than deep.

1 INTRODUCTION

Test-time scaling has many sophisticated instantiations, but the workhorse remains the simplest one. We sample N completions and return the answer that appears most often (Wang et al., 2023b). This self-consistency majority vote needs no verifier and no reward model, and it is a surprisingly hard baseline to beat (Anonymous, 2026; Feng et al., 2025). Training has a workhorse too. Reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Guo et al., 2025) reliably raises the accuracy of a single sample. The standard pipeline composes the two: train one policy with RLVR, then sample it many times and vote. This paper asks whether that composition works. Holding the inference budget fixed, does spending an RLVR budget on one policy improve its vote?

Ensemble theory suggests caution. A good voter pool needs members that are individually competent and collectively diverse. When one voter is wrong, the others should not be wrong in the same way (Lakshminarayanan et al., 2017). A vote can only overturn a mistake that the majority does not share, so what majority voting really pays for is *decorrelated errors*. RLVR works by sharpening the policy around behaviors that already earn reward. Pass@1 rises, but the sampling distribution narrows and diversity collapses (Yue et al., 2025; Dang et al., 2025; Wen et al., 2025; Li et al., 2025b). Resampling one sharpened policy therefore produces voters that make the same mistakes. Prior work measures this collapse through pass@k and coverage. We measure what it does to the majority vote.

We find that concentrated RLVR hurts the vote, and that the fix is simple. The damage is not inherent to RLVR. It comes from concentrating the whole budget in one policy. We therefore split the training pool into K disjoint shards and train K independent LoRA adapters (Hu et al., 2022), each with $1/K$

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

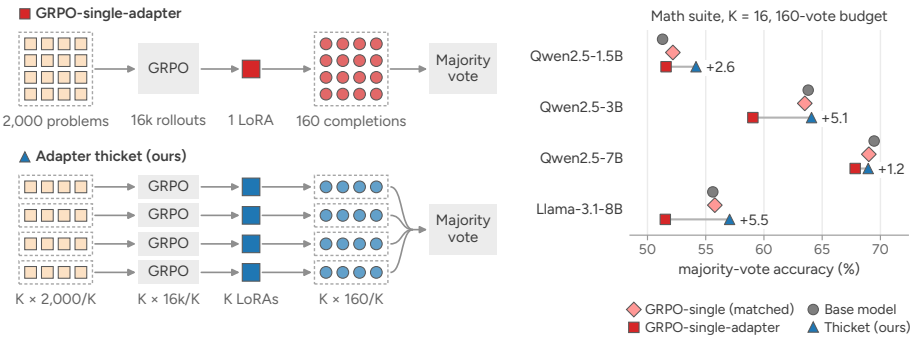


Figure 1: **One RLVR budget, one inference budget, two allocations.** *Left:* GRPO-single-adapter trains one LoRA on the full 2,000-problem pool with all 16k rollouts. An adapter thicket trains K LoRAs on disjoint $2000/K$-example shards with $16\text{k}/K$ rollouts each. At inference every configuration pools exactly 160 completions per problem (K voters $\times 160/K$ samples). *Right:* full-pool majority vote at $K{=}16$ on the math suite (macro mean over five benchmarks, three seeds), including the step-matched single-adapter control. The fully trained GRPO-single-adapter (red squares) falls below the base on three of four models. Deltas: thicket − GRPO-single-adapter.

of the rollout budget. This uses the same total data and the same total training compute as the single adapter. We call the result an *adapter thicket*, echoing the finding that diverse task specialists sit densely around pretrained weights (Gan and Isola, 2026). Four design choices keep the comparison honest. Every configuration pools exactly 160 completions per problem (K voters $\times$ $160/K$ samples). A *step-matched* control stops the same single-adapter run at the step count a thicket member receives. We trace the single adapter's vote along its whole training run. And we recount every vote over each method's first B completions, for B from 16 to 160. We report six findings.

1. **Concentrated RLVR hurts the vote.** A single adapter trained on the full RLVR budget raises single-sample accuracy on every model, yet votes below the untrained base on three of four models (by up to 4.8 p.p.) and covers less of the answer space than the base in seven of eight model–suite settings (Section 4.1).

2. **The damage accrues with training.** Along the single adapter's run, voter error correlation $\bar{\rho}$ rises monotonically on every model while voter accuracy rises overall. The vote peaks by step 125 and then falls by 1.4 to 7.0 p.p. On Qwen2.5-3B and 7B, no checkpoint out-votes the base (Section 4.2).

3. **Splitting the budget avoids the damage.** Thickets beat the fully trained adapter in all 16 (model, K) majority-vote settings and all 32 (model, suite, K) coverage settings. Once $K{\geq}4$, they stay within 0.8 p.p. of the base or above it on every model. Against the step-matched control, they win 10 of 16 settings and tie one (Section 4.3).

4. **. . . while keeping most of the single-sample gain.** A single adapter stopped at a member's step count is a strong control at $K \leq 4$, winning four of eight vote comparisons. At $K \geq 8$ thickets win six of eight, tie one, and have more accurate voters on three of four models. Where RLVR adds real pass@1 (Qwen2.5-1.5B and Llama, +7.3 and +9.7 p.p.), thicket members keep 85–95% of it at $K \leq 4$ and 63–78% at $K \geq 8$, while the step-matched run only 8–66% (Section 4.4).

5. **The cost of concentration grows with the vote budget.** Averaged over the 16 settings, the fully trained adapter votes level with the base at 16 votes (+0.15 p.p.) and 2.58 p.p. below it at 160. The thicket's lead over it grows from +1.33 to +3.28 p.p. The thicket's lead over the base shrinks, from +1.48 to +0.71 p.p. (Section 4.5).

6. **Random shards are enough.** Engineering the shards by subject does not help. At $K{=}7$, random disjoint shards vote at least as well as subject-specialist shards on all four models (Section 4.7).

2 RELATED WORK

Self-consistency majority voting (Wang et al., 2023b) remains the canonical verifier-free test-time scaling method. We treat it as the aggregator to feed rather than an opponent to replace. Our results connect most directly to work showing that RLVR narrows the sampling distribution and reduces test-time coverage (Yue et al., 2025; Li et al., 2025b). That work measures the collapse through pass@k. We measure its consequence for the majority vote at a fixed inference budget, and show that it can leave the vote below that of the untrained model. Prior approaches try to recover diversity within or around a single trained policy (Dang et al., 2025). We instead preserve it by splitting the training budget across independent policies. Neural Thickets (Gan and Isola, 2026) showed that full-weight perturbations around a pretrained model can produce diverse specialists that form an effective post-training ensemble. Existing LoRA ensemble work has primarily targeted uncertainty estimation and calibration (Wang et al., 2023a; Li et al., 2025a), while related methods compose, route, or merge already-trained adapters (Huang et al., 2023; Wu et al., 2024; Chronopoulou et al., 2023). We instead replace the perturb-and-select procedure of Neural Thickets with reward-directed RLVR training in LoRA space and study how a fixed RLVR budget should be allocated between one deeply trained adapter and many shallowly trained adapters. We do not consider techniques that produce extra voters from a single run (e.g., snapshot ensembles (Huang et al., 2017)) because they apply equally to both sides of our comparison.

3 ADAPTER THICKETS AND EXPERIMENTAL SETUP

Setup. Let θ_0 be a frozen instruction-tuned model and $\mathcal{P} = \{(x_j, a_j)\}_{j=1}^{D}$ a pool of problems with verifiable answers. A binary verifier $V(x, y, a) \in \{0, 1\}$ checks the final answer of completion y against a. All trained arms share one recipe: LoRA + GRPO with reward $r(x, y) = V(x, y, a)$, identical hyperparameters, and a total budget of B_{train} on-policy training rollouts. We compare four ways of fielding K voters.

- **Base**: the untrained θ_0, K i.i.d. streams.
- **GRPO-single-adapter**: the standard recipe. One adapter trained on all of $\mathcal{P}$ with the full budget B_{train}, sampled as K i.i.d. streams from the final checkpoint.
- **GRPO-single (matched)**: the *same* single-adapter run, evaluated at the checkpoint matched to a thicket member's step count (Appendix A), again sampled as K i.i.d. streams. This control tests whether the thicket still wins when no voter is trained longer than a thicket member.
- **adapter thicket**: K independently trained adapters, each trained for B_{train}/K rollouts on one of K disjoint random shards of $\mathcal{P}$. Each adapter is one voter.

Every training problem is seen exactly once, by one adapter, in both training arms.

A fixed inference budget. Each voter contributes $160/K$ completions per problem, so *every* method at *every* K pools exactly 160 votes. Any difference between methods is therefore a difference in *who votes*, never in how much is sampled. For a single policy, the K voters are nominal. We draw one pool of 160 i.i.d. completions per problem, each with its own sampling seed, and group it into K streams. The base and GRPO-single-adapter votes are therefore identical at every K, and only the thicket and GRPO-single (matched) (whose checkpoint depends on K) change with K. Section 4.5 varies the budget from 16 to 160 votes. Before voting, we drop unparseable completions (nulls), merge mathematically equivalent answers (e.g., 0.5 and $1/2$) into one candidate, and marginalize exact ties (Appendix B). We report four quantities: **majority vote** on the math suite, **coverage** (Brown et al., 2024) (pass@pool, whether any of the 160 completions is correct), **pass@1** (mean single-completion accuracy, which we also call voter accuracy), and voter diversity. Diversity is the mean pairwise Pearson correlation of the voters' binary error indicators, $\bar{\rho} = \frac{1}{K(K-1)} \sum_{i \neq j} \operatorname{corr}(e_i, e_j)$.

Seeds. We repeat every main experiment under three seeds. The seed controls the thicket's data partition, the single adapter's data ordering, and the sampling seeds of every pool. Every number is a mean over the three seeds, with the sample standard deviation after $\pm$. The one exception is the subject-versus-random shard ablation (Section 4.7), which uses seed 42. Deltas quoted in prose are differences of the displayed means. A mean margin is the mean of these differences over settings.

Table 1: **Majority-vote accuracy on the math suite under the fixed 160-vote budget** (canonical-class votes; macro means (%) over five benchmarks; mean ± std over three seeds). Base and GRPO-single-adapter each vote over one 160-completion pool, so their rows are constant across K by construction. Bold: best method per model and K by displayed mean; underlined: second best.

Model	Method	K=2	K=4	K=8	K=16
Qwen2.5-1.5B	Base	51.3 ±0.4	51.3 ±0.4	51.3 ±0.4	51.3 ±0.4
	GRPO-single (matched)	54.6 ±0.7	**55.3 ±2.0**	53.7 ±0.2	52.2 ±0.9
	GRPO-single-adapter	51.6 ±1.6	51.6 ±1.6	51.6 ±1.6	51.6 ±1.6
	Thicket	**55.2 ±2.0**	53.6 ±0.7	**54.6 ±0.5**	**54.2 ±0.2**
Qwen2.5-3B	Base	**63.8 ±0.3**	63.8 ±0.3	**63.8 ±0.3**	63.8 ±0.3
	GRPO-single (matched)	61.9 ±1.2	63.5 ±1.2	63.3 ±0.1	63.5 ±0.8
	GRPO-single-adapter	59.0 ±5.1	59.0 ±5.1	59.0 ±5.1	59.0 ±5.1
	Thicket	63.0 ±1.2	**64.0 ±0.4**	63.6 ±0.2	**64.1 ±0.4**
Qwen2.5-7B	Base	**69.5 ±0.2**	**69.5 ±0.2**	**69.5 ±0.2**	**69.5 ±0.2**
	GRPO-single (matched)	68.7 ±1.1	67.9 ±1.2	69.2 ±0.1	69.0 ±0.2
	GRPO-single-adapter	67.8 ±1.2	67.8 ±1.2	67.8 ±1.2	67.8 ±1.2
	Thicket	68.6 ±1.1	68.9 ±0.4	68.7 ±1.0	69.0 ±0.4
Llama-3.1-8B	Base	**55.6 ±0.3**	55.6 ±0.3	55.6 ±0.3	55.6 ±0.3
	GRPO-single (matched)	53.9 ±1.0	**58.5 ±1.5**	56.8 ±0.5	55.8 ±0.6
	GRPO-single-adapter	51.5 ±3.2	51.5 ±3.2	51.5 ±3.2	51.5 ±3.2
	Thicket	53.0 ±4.2	56.8 ±2.1	**57.8 ±1.5**	**57.0 ±0.7**

Training and evaluation. We use Qwen2.5-{1.5B, 3B, 7B}-Instruct (Yang et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). The Qwen2.5 family spans the size axis. We add Llama-3.1-8B to confirm that the results generalize across model families rather than being a Qwen-specific artifact. We study two domains: *math* ($D = 2{,}000$ MATH problems (Hendrycks et al., 2021), with a verified-answer reward) and *code* ($D = 2{,}000$ LeetCodeDataset problems (Xia et al., 2025), with a unit-test reward). All runs use rank-64 LoRA, GRPO with 8 rollouts per prompt ($B_{\text{train}} = 16$k rollouts per arm), and $K \in \{2, 4, 8, 16\}$ (Appendix A). Evaluation is zero-shot at temperature 1.0 and top-p 0.95. The *math* suite is MATH500, GSM8K (Cobbe et al., 2021), OlympiadBench (He et al., 2024), Countdown (Katz et al., 2025), and Knights & Knaves (Xie et al., 2024). The *code* suite is MBPP (Austin et al., 2021), HumanEval (Chen et al., 2021), and the LeetCodeDataset test split. Suite results are *macro means over benchmarks*. Programs have no canonicalizable final answer to vote over, so code enters only the coverage, pass@1, and error-correlation analyses.

4 Results

4.1 Concentrated RLVR hurts the vote

Table 1 gives the central comparison. Read the table in blocks of four rows. For a given model and voter count K, the four rows share one 160-completion budget and differ only in who the voters are: fresh samples of the base model, samples of the single adapter stopped early (GRPO-single (matched)), samples of the fully trained single adapter (GRPO-single-adapter), or one sample stream from each of K thicket members.

Start with the standard recipe. Full single-adapter RLVR does what it is supposed to do for a single sample: it raises pass@1 on every model, by +7.3, +1.0, +0.7, and +9.7 p.p. on Qwen2.5-1.5B, Qwen2.5-3B, Qwen2.5-7B, and Llama-3.1-8B. Yet at the same 160-completion budget it votes *below the untrained base* on Qwen2.5-3B (−4.8 p.p.), Qwen2.5-7B (−1.7), and Llama (−4.1), and only matches it on Qwen2.5-1.5B (+0.3). As work on diversity collapse would predict (Yue et al., 2025), the lost votes track lost diversity. Full training raises the voters' error correlation $\bar{\rho}$ above the base in all eight model–suite settings at K=16, by +0.05 to +0.20 (Table 18 in Appendix E). The fully trained adapter also covers *less* of the answer space than the base in seven of eight model–suite settings, by up to 6.4 p.p. on code, and ties it in the eighth (Table 2). This deficit grows with the number of votes (Section 4.5).

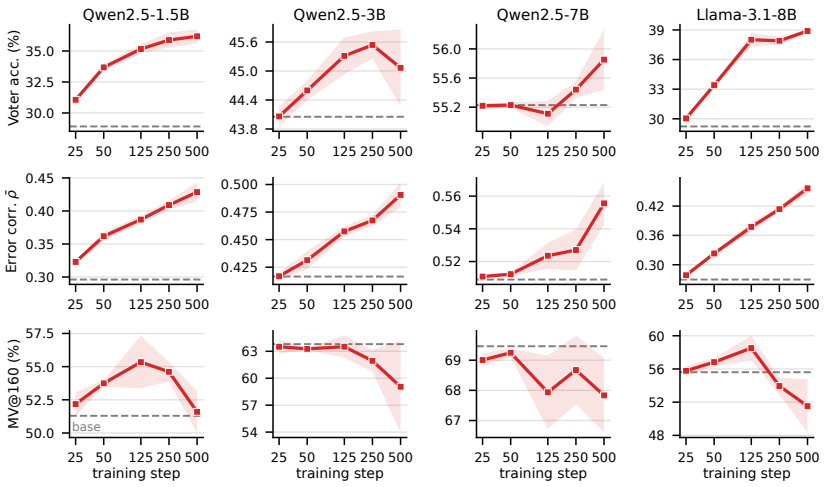


Figure 2: **Single-adapter training raises voter accuracy and voter correlation together – and the vote pays for it.** Checkpoints of the GRPO-single-adapter run at steps $25/50/125/250$ and the final checkpoint (step 499 for Qwen, 500 for Llama; plotted at 500), evaluated on the math suite with the same fixed 160-completion budget per problem (grouped as 16 voters $\times$ 10 samples; mean over three seeds, ± 1 std band). Top: mean voter accuracy. Middle: mean pairwise error correlation $\bar{\rho}$. Bottom: majority vote over the 160-vote pool. Dashed lines: the untrained base model. On every model $\bar{\rho}$ rises monotonically with training, and the vote peaks by step 125 before declining.

4.2 The damage accrues with training

Figure 2 traces the GRPO-single-adapter run along its training trajectory. We evaluate checkpoints at steps $25/50/125/250$ and the final checkpoint (≈ 500) under the same fixed 160-vote budget. Mean voter accuracy rises overall, for example $30.0 \rightarrow 38.9$ on Llama and $31.1 \rightarrow 36.2$ on Qwen2.5-1.5B. But the voters' error correlation rises with it, monotonically on all four models: $0.28 \rightarrow 0.46$ on Llama and $0.32 \rightarrow 0.43$ on Qwen2.5-1.5B. The majority vote is what a test-time-scaling user actually consumes, and it peaks by step 125 and then falls. Llama reaches 58.5 at step 125 and ends at 51.5, a drop of 7.0 p.p. Qwen2.5-1.5B peaks at 55.3 (step 125) and ends at 51.6. On Qwen2.5-3B and 7B the vote never rises above the untrained base at all (peaks of 63.5 and 69.2, against base votes of 63.8 and 69.5), and training ends 4.8 and 1.7 p.p. below it.

This reframes the usual early-stopping question. If you plan to sample many times and vote, pass@1 is the wrong dial to watch, because the vote can be deteriorating while pass@1 still climbs. Stopping early can rescue the vote, but only at the right step, and the right step differs across models: step 125 on Qwen2.5-1.5B and Llama, while on Qwen2.5-3B and 7B no checkpoint beats the base. Stopping early also leaves much of the training data and budget unused (three quarters of it at step 125). The next section asks whether the budget can be spent in full without paying for it in the vote.

4.3 Splitting the budget avoids the damage

A thicket spends the same budget as GRPO-single-adapter on the same data, but spreads it across K members. At the fixed 160-vote budget, thickets beat the fully trained GRPO-single-adapter in *all sixteen* (model, K) settings (Table 1). The margins range from $+0.8$ to $+6.3$ p.p. At $K{=}16$ they are $+2.6$, $+5.1$, $+1.2$, and $+5.5$ p.p. on Qwen2.5-1.5B, Qwen2.5-3B, Qwen2.5-7B, and Llama. Coverage follows the vote. Thickets cover more than GRPO-single-adapter in all 32 (model, suite, K) settings, by $+0.8$ to $+3.6$ p.p. on math and $+1.5$ to $+5.2$ p.p. on code (Table 2).

Against the untrained base, the thicket's job is to avoid the damage, and it mostly does. Thickets beat the base in 9 of 16 settings, including every K on Qwen2.5-1.5B (by $+2.3$ to $+3.9$) and every $K{\geq}4$ on Llama ($+1.2$ to $+2.2$). On Qwen2.5-3B they stay within 0.8 p.p. of it, and on Qwen2.5-7B they trail it at every K, by 0.5 to 0.9 p.p. The one setting where a thicket trails the base by more than a point is the two-member Llama thicket (53.0 vs. 55.6). A two-member committee is the least diverse

Table 2: **Coverage** (pass@pool: at least one correct completion among the 160) on the math and code suites; macro means (%); mean ± std over three seeds. Bold: best method per model, suite, and K; underlined: second best.

		Coverage, math				Coverage, code			
Model	Method (K=)	2	4	8	16	2	4	8	16
Qwen2.5-1.5B	Base	**89.6 ±0.3**	**89.6 ±0.3**	89.6 ±0.3	89.6 ±0.3	**72.2 ±0.4**	**72.2 ±0.4**	**72.2 ±0.4**	**72.2 ±0.4**
	GRPO-single (matched)	**89.6 ±0.4**	89.3 ±0.5	**90.0 ±0.3**	89.5 ±0.5	69.0 ±1.8	70.2 ±0.9	72.0 ±0.5	71.9 ±0.3
	GRPO-single-adapter	87.7 ±0.7	87.7 ±0.7	87.7 ±0.7	87.7 ±0.7	65.8 ±1.4	65.8 ±1.4	65.8 ±1.4	65.8 ±1.4
	Thicket	88.8 ±1.0	89.5 ±0.4	89.6 ±0.5	**89.8 ±0.6**	69.2 ±1.4	70.6 ±0.6	70.6 ±0.5	70.3 ±0.3
Qwen2.5-3B	Base	**91.7 ±0.1**	**91.7 ±0.1**	**91.7 ±0.1**	**91.7 ±0.1**	**75.6 ±1.1**	**75.6 ±1.1**	**75.6 ±1.1**	75.6 ±1.1
	GRPO-single (matched)	90.8 ±0.3	91.0 ±0.5	91.6 ±0.6	91.6 ±0.5	73.7 ±2.0	73.1 ±1.7	75.4 ±0.8	**76.4 ±0.9**
	GRPO-single-adapter	88.9 ±1.0	88.9 ±1.0	88.9 ±1.0	88.9 ±1.0	70.7 ±0.5	70.7 ±0.5	70.7 ±0.5	70.7 ±0.5
	Thicket	90.8 ±0.6	90.8 ±0.8	91.3 ±0.4	91.6 ±0.1	73.5 ±1.4	74.0 ±1.4	75.2 ±0.4	75.9 ±0.3
Qwen2.5-7B	Base	**92.8 ±0.4**	**92.8 ±0.4**	**92.8 ±0.4**	**92.8 ±0.4**	**76.3 ±0.2**	**76.3 ±0.2**	**76.3 ±0.2**	76.3 ±0.2
	GRPO-single (matched)	92.3 ±0.2	91.7 ±0.6	92.6 ±0.2	92.3 ±0.8	74.3 ±0.5	75.7 ±0.7	**76.3 ±0.4**	**76.4 ±0.2**
	GRPO-single-adapter	91.2 ±0.5	91.2 ±0.5	91.2 ±0.5	91.2 ±0.5	73.1 ±0.4	73.1 ±0.4	73.1 ±0.4	73.1 ±0.4
	Thicket	92.0 ±0.2	92.4 ±0.4	92.3 ±0.2	92.6 ±0.2	75.7 ±0.2	75.9 ±0.2	**76.3 ±0.4**	**76.4 ±0.4**
Llama-3.1-8B	Base	86.3 ±0.3	86.3 ±0.3	86.3 ±0.3	86.3 ±0.3	**75.2 ±0.2**	**75.2 ±0.2**	**75.2 ±0.2**	75.2 ±0.2
	GRPO-single (matched)	87.5 ±0.3	**89.4 ±0.5**	87.9 ±0.1	86.8 ±0.6	72.6 ±1.7	73.8 ±1.1	75.0 ±0.4	**75.4 ±0.1**
	GRPO-single-adapter	86.3 ±2.0	86.3 ±2.0	86.3 ±2.0	86.3 ±2.0	71.8 ±1.9	71.8 ±1.9	71.8 ±1.9	71.8 ±1.9
	Thicket	**88.2 ±0.2**	89.1 ±0.5	**89.9 ±0.7**	**88.5 ±0.2**	73.3 ±1.4	74.4 ±0.5	74.3 ±0.1	74.8 ±0.5

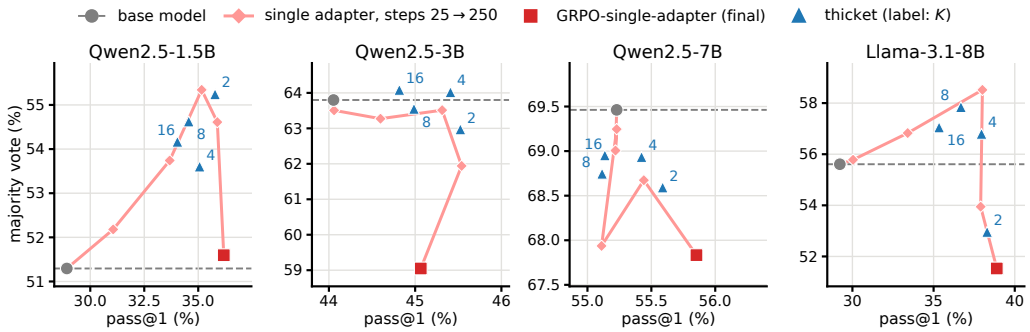


Figure 3: **Single-sample accuracy vs. majority vote** on the math suite (pass@1 = mean voter accuracy; majority vote over the fixed 160-completion pool; macro means, mean over three seeds). The pink path follows the single adapter from the untrained base (gray circle) through steps 25/50/125/250 to its final checkpoint, GRPO-single-adapter (red square). These checkpoints are the step-matched pools for $K = 16/8/4/2$. Blue triangles are thickets, labeled by K. Dashed line: the base model's vote. Full training moves right but, on three of four models, down below the base. Thickets with $K \geq 4$ keep most of the rightward move where there is one, while staying within 0.8 p.p. of the base's vote or above it. At $K \leq 4$ the matched checkpoints sit at about the thickets' pass@1, and the vote goes either way. At $K \geq 8$ the thickets lie to the right of and above the matched checkpoints on three of four models.

thicket, since each member casts half the votes, and this cell has the largest seed spread of any thicket entry in Table 1 (±4.2). Coverage tells the same story. Thickets stay close to base-level coverage, though usually just below it (23 of 32 settings, by at most 3.0 p.p.), and on Llama math they exceed it at every K.

The step-matched control is the stronger opponent. Thickets beat GRPO-single (matched) in 10 of 16 settings and tie one. Its two largest wins (1.7 p.p. each) come at K=4 on Qwen2.5-1.5B and Llama, and its other wins are at most 0.9 p.p. (Section 4.4). Per-benchmark exceptions to the thicket's win over GRPO-single-adapter are rare: 6 of 80 cells, all on Countdown or Knights & Knaves.

4.4 ... WHILE KEEPING MOST OF THE SINGLE-SAMPLE GAIN

Avoiding the damage is easy if we give up training: the untrained base does it, and so, at the right step, does stopping the single run early. The fair question is whether a thicket keeps more of RLVR's single-sample gain than a single adapter stopped at a member's step count. The pink path in Figure 3 is that control.

At small K, early stopping is as good as splitting. On Qwen2.5-1.5B and Llama, where full training lifts pass@1 by $+7.3$ and $+9.7$ p.p. (only $+1.0$ and $+0.7$ on Qwen2.5-3B and 7B, too little to divide), thicket members keep 94–95% of that gain at $K{=}2$ and 85–91% at $K{=}4$. The step-matched adapter keeps about as much (90–96% and 86–91%). At $K{=}4$ it also votes higher, $+4.0$ and $+2.9$ p.p. above the base against $+2.3$ and $+1.2$ for the thicket. Across all four models at $K \leq 4$, the two win four of eight vote comparisons each.

At large K, splitting wins clearly. The step-matched run keeps only 43–66% of the pass@1 gain at $K{=}8$ and 8–30% at $K{=}16$ on the same two models. Thicket members, each trained on $1/8$ or $1/16$ of the budget and its own shard, keep 77–78% and 63–70%. Their voters are 0.9 to 5.3 p.p. more accurate (on three of four models overall), and the thicket out-votes the step-matched pool by $+0.9$ to $+2.0$ p.p. Across all four models at $K \geq 8$, the thicket wins six of eight comparisons, ties one, and loses one (Qwen2.5-7B at $K{=}8$, by 0.5).

Splitting is also the more robust choice. The thicket's vote varies by at most 1.0 p.p. across $K \in \{4, 8, 16\}$ on every model, whereas the step-matched vote varies by more than 1.0 p.p. on three of four models (up to 3.1). The single run's best step also differs across models (Section 4.2).

Why does the thicket keep the accuracy but not the damage? Its members are competent without being redundant. At $K{=}2$ they are on average as accurate as the fully trained adapter (within 0.6 p.p. on every model's math suite), yet thicket voters are less error-correlated than GRPO-single-adapter voters in all 32 (model, suite, K) settings, and at $K{=}2$ the thicket out-votes GRPO-single-adapter by $+0.8$ to $+4.0$ p.p. At $K{=}16$ they add at most $+0.07$ correlation over the base, against $+0.05$ to $+0.20$ for GRPO-single-adapter. Appendix E splits the thicket's margin over GRPO-single-adapter into the RLVR gain it keeps and the vote damage it avoids (Table 17).

4.5 THE COST OF CONCENTRATION GROWS WITH THE VOTE BUDGET

We now ask how the comparison changes with the number of votes. For $B = 16, 32, \ldots, 160$, every method votes over its first B completions in generation order (Appendix B), so smaller budgets are prefixes of larger ones. A thicket member contributes its first B/K completions, one each at $B{=}16$ and $K{=}16$. A single policy contributes the completions drawn with its first B sampling seeds, so the base and GRPO-single-adapter votes are again identical at every K. Figure 4 reports mean margins over the 16 (model, K) settings.

The fully trained adapter's deficit grows with the budget. At 16 votes it is level with the base on average ($+0.15$ p.p.). It votes above the base on Qwen2.5-1.5B and Llama ($+1.4$ and $+1.8$ p.p.) and below it on Qwen2.5-3B and 7B (-1.5 and -1.1). From 32 votes on it trails the base on average, and the gap widens to 2.58 p.p. at 160 votes. On Llama it leads the base by 1.8 p.p. at 16 votes and trails it by 4.1 at 160. Extra samples help its vote much less. From 16 to 160 votes its majority vote rises by 3.78 p.p. on average, against 6.50 for the base, 6.02 for GRPO-single (matched), and 5.73 for the thicket (means of the per-setting gains). Its voters have the most correlated errors of the four pools in all 32 (model, suite, K) settings (Appendix E). This is consistent with extra samples repeating answers the pool has already given.

The thicket's mean lead over GRPO-single-adapter never shrinks as the budget grows. It rises from $+1.33$ p.p. at 16 votes (ahead in 15 of 16 settings) to $+3.28$ at 160 (ahead in all 16). The growth comes from three models. On Qwen2.5-7B, where RLVR adds least, the lead stays between $+0.78$ and $+1.18$ p.p. The thicket's leads over the other two controls shrink. Over the base, the mean lead falls from $+1.48$ p.p. at 16 votes (ahead in 13 settings) to between $+0.83$ and $+0.98$ from 48 to 144 votes, and to $+0.71$ at 160 (ahead in 9). Over GRPO-single (matched), it falls from $+0.56$ (ahead in 11) to $+0.27$ (ahead in 10). Averaged over K, each trained pool gains less from extra votes than the base, so each loses ground to it as the budget grows.

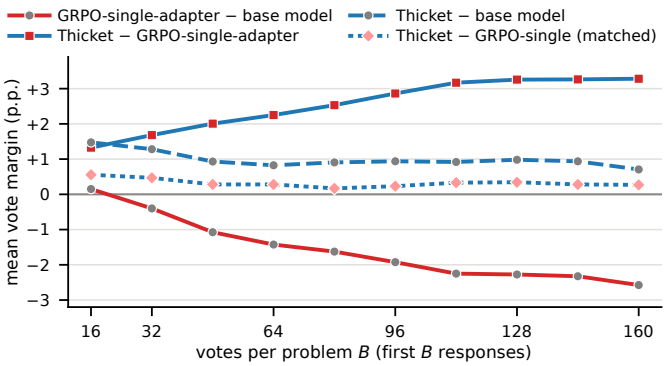


Figure 4: **The cost of concentration grows with the vote budget.** Mean majority-vote margin on the math suite (p.p.) over the 16 (model, K) settings, as every method votes over its first B completions in generation order. Each setting's margin is the difference of displayed three-seed means. Line colour marks the method in front, and markers mark the method subtracted. Red line with gray circles: GRPO-single-adapter − base. Blue line with red squares: thicket − GRPO-single-adapter. Dashed blue with gray circles: thicket − base. Dotted blue with pink diamonds: thicket − GRPO-single (matched). B=160 is Table 1. Table 19 in Appendix F gives the values.

Within the thicket, the loss again follows concentration. A larger K spreads the budget more thinly, and the thicket's vote gains more from extra samples: $+4.68$ p.p. from 16 to 160 votes at K=2, $+5.43$ at K=4, $+6.35$ at K=8, and $+6.48$ at K=16, level with the base's $+6.50$. From 16 to 160 votes the thicket's mean lead over the base falls from $+1.73$ to -0.10 p.p. at K=2 and from $+1.85$ to $+0.78$ at K=4, but only from $+1.28$ to $+1.13$ at K=8 and from $+1.05$ to $+1.03$ at K=16. Appendix F plots every curve and repeats Tables 1 and 2 at 16 votes.

4.6 Confidence-filtered voting

A pool that has lost its dissent should also gain little from weighing its votes. We test this with *confidence-filtered voting*. Each voter keeps only its top-q fraction of completions by mean log-likelihood $\bar{\ell}$, and plain majority voting runs over the survivors. We select $q \in \{1, 1/2, 1/4, \text{top-}1\}$ by per-benchmark five-fold cross-validation, so every problem is scored under a q chosen without seeing it, and q=1 (plain voting) is always available (Appendix B). We score all four methods at 160 votes, each against its own plain majority vote. Correct completions have the higher median $\bar{\ell}$ in every method–model pool at K=16 (Figure 9 in Appendix C), so the filter has a signal to use.

Filtering helps three of the four pools, and it gives nothing to the pool whose errors are most correlated. Its mean gain over plain voting is $+0.53 \pm 0.10$ p.p. for the thicket, $+0.51 \pm 0.06$ for the untrained base, and $+0.43 \pm 0.08$ for GRPO-single (matched) (means over the 16 settings of differences of displayed three-seed means; $\pm$ std over seeds). The thicket gains in 15 of 16 settings (by up to 1.1 p.p.), and the base and GRPO-single (matched) in all 16. For the fully trained GRPO-single-adapter it is $+0.01 \pm 0.14$, that is, no gain. Filtering raises its vote in 6 settings, leaves it unchanged in 4, and lowers it in 6. We read this as a sharpened policy leaving little dissent to filter. Filtering widens the thicket's lead over GRPO-single-adapter from $+3.28$ to $+3.80$ p.p. and leaves its lead over the base almost unchanged ($+0.71$ to $+0.73$).

4.7 Random shards are enough

Random disjoint sharding is only the simplest way to split a budget. A natural question is whether *engineered* shards do better. We partition the MATH training pool by its subject metadata into seven specialist shards (Algebra, Counting & Probability, Geometry, Intermediate Algebra, Number Theory, Prealgebra, Precalculus), equalized at 184 examples per adapter (the size of the smallest subject). The control is a thicket of seven *random* disjoint shards of the same size. Both arms use the same GRPO

Table 3: **Subject-specialist vs. random disjoint shards at** $K{=}7$ (184 training problems per adapter in both arms; $7 \times 23 = 161$ completions per problem; single seed, seed 42). Macro means (%) over the five math benchmarks, except $\bar{\rho}$. Bold: better arm per model and column (ties bold both; $\bar{\rho}$ is not bolded).

Model	Shards	Majority vote	Filtered vote	Coverage	Voter acc.	$\bar{\rho}$
Qwen2.5-1.5B	Subject	54.2	55.2	**89.6**	**34.4**	0.37
	Random	**54.8**	**56.0**	**89.6**	**34.4**	0.37
Qwen2.5-3B	Subject	62.9	**63.4**	**91.5**	44.8	0.44
	Random	**63.1**	62.7	91.0	**44.9**	0.44
Qwen2.5-7B	Subject	69.1	69.3	**92.7**	**55.2**	0.51
	Random	**69.3**	**69.5**	92.1	55.0	0.51
Llama-3.1-8B	Subject	57.1	57.9	89.1	35.9	0.34
	Random	**57.9**	**59.0**	**89.6**	**36.1**	0.35

recipe and draw 23 samples per member, for 161 completions per problem. We run this ablation for one seed (Table 3; Appendix G).

Engineering the shards buys nothing. Random shards vote at least as well as subject shards on all four models ($+0.2$ to $+0.8$ p.p.), and win under confidence filtering on three of four. Coverage, voter accuracy, and error correlation are essentially identical across the two arms (within 0.6 p.p., 0.2 p.p., and 0.01). Which disjoint split a thicket uses matters little, and random shards are enough.

5 Discussion, limitations, and conclusion

Practical picture. The standard pipeline trains one policy with RLVR and then samples it many times. Our results say this composition is lossy. Training raises pass@1, but it spends the diversity the vote needs, and at a fixed inference budget the vote can end below where it started. The more votes a user draws, the larger the loss. Splitting the budget is a drop-in fix that changes how training is spent, not how inference works. At deployment a thicket is K LoRA adapters over one frozen base model, and modern multi-adapter servers batch concurrent LoRAs over shared base weights at near-single-model cost (Sheng et al., 2023). Training is embarrassingly parallel: on K workers a thicket finishes in $1/K$ of the single adapter's wall-clock time. The recipe adds no hyperparameters beyond K, and for $K \geq 4$ the choice of K is forgiving (Section 4.4). The two-member thicket is the least reliable configuration: it is the weakest thicket on three of four models and has the largest seed spread of any thicket entry in Table 1.

Limitations. Our study covers one RL algorithm (GRPO), voter counts $K \leq 16$, models up to 8B, and budgets of 16 to 160 votes. Thickets protect the vote rather than guarantee a gain over the base. Their lead over the base and over GRPO-single (matched) shrinks as the budget grows, and at 160 votes they beat the base in 9 of 16 settings. On Llama the base's vote is still rising at 160 votes, so the lead may shrink further at larger budgets. At a single sample the order reverses: GRPO-single-adapter has the highest pass@1 of the four pools on Qwen2.5-1.5B, Qwen2.5-7B, and Llama at every K. When RLVR adds little pass@1, as on Qwen2.5-7B, a thicket roughly matches the untrained model. All arms train without a KL penalty (Table 4), and we do not test whether KL regularization changes the single adapter's vote. Majority voting needs comparable answers, so our voting results are math-only, and code enters only through coverage, pass@1, and correlation. The subject-shard ablation uses a single seed. Finally, we test only random and subject-based sharding.

Conclusion. Majority voting and RLVR are each hard to beat, but composing them naively is costly. Concentrating an RLVR budget in one policy spends the diversity the vote needs: on three of four models its vote ends below the untrained model's, by more as votes grow. Splitting the same budget across disjoint shards keeps most of the accuracy and avoids the damage. When the plan is to sample and vote, train a thicket, not a champion.

REFERENCES

Anonymous. Counting is hard to beat: The limits of confidence-weighted test-time aggregation. Anonymous manuscript under review, 2026.

J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. Program synthesis with large language models. *arXiv:2108.07732*, 2021.

B. Brown et al. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv:2407.21787*, 2024.

M. Chen et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

A. Chronopoulou, M. E. Peters, A. Fraser, and J. Dodge. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In *Findings of EACL*, 2023.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

X. Dang, C. Baek, K. Wen, Z. Kolter, and A. Raghunathan. Weight ensembling improves reasoning in language models. In *Conference on Language Modeling (COLM)*, 2025.

Z. Feng, Q. Chen, N. Lu, Y. Li, S. Cheng, S. Peng, D. Tang, S. Liu, and Z. Zhang. Is PRM necessary? Problem-solving RL implicitly induces PRM capability in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, pages 36515–36533, 2025.

Y. Gan and P. Isola. Neural thickets: Diverse task experts are dense around pretrained weights. *arXiv:2603.12228*, 2026.

A. Grattafiori et al. The Llama 3 herd of models. *arXiv:2407.21783*, 2024.

D. Guo et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, J. Liu, L. Qi, Z. Liu, and M. Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *ACL*, 2024.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*, 2021.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, and M. Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. *arXiv:2307.13269*, 2023.

G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *ICLR*, 2017.

M. Katz, H. Kokel, and S. Sreedharan. Seemingly simple planning problems are computationally hard: The countdown game. *arXiv:2508.02900*, 2025.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Y. Li et al. Ensembles of low-rank expert adapters. *arXiv:2502.00089*, 2025a.

L. Li, Z. Zhou, J. Hao, J. K. Liu, Y. Miao, W. Pang, X. Tan, W. Chu, Z. Wang, S. Pan, C. Qu, and Y. Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv:2509.07430*, 2025b.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

Y. Sheng, S. Cao, D. Li, C. Hooper, N. Lee, S. Yang, C. Chou, B. Zhu, L. Zheng, K. Keutzer, J. Gonzalez, and I. Stoica. S-LoRA: Serving thousands of concurrent LoRA adapters. *arXiv:2311.03285*, 2023.

G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. HybridFlow: A flexible and efficient RLHF framework. In *EuroSys*, 2025.

X. Wang, L. Aitchison, and M. Rudolph. LoRA ensembles for large language model fine-tuning. *arXiv:2310.00035*, 2023a.

X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023b.

X. Wen et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. *arXiv:2506.14245*, 2025.

X. Wu, S. Huang, and F. Wei. Mixture of LoRA experts. In *ICLR*, 2024.

Y. Xia, W. Shen, Y. Wang, J. K. Liu, H. Sun, S. Wu, J. Hu, and X. Xu. LeetCodeDataset: A temporal dataset for robust evaluation and efficient training of code LLMs. *arXiv:2504.14655*, 2025.

C. Xie, Y. Huang, C. Zhang, D. Yu, X. Chen, B. Y. Lin, B. Li, B. Ghazi, and R. Kumar. On memorization of large language models in logical reasoning. *arXiv:2410.23123*, 2024.

A. Yang et al. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, Y. Yue, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv:2504.13837*, 2025.

A TRAINING DETAILS

Models. Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), all frozen; only LoRA parameters train.

Data and partitioning. All math experiments use a fixed pool of $D = 2{,}000$ problems sampled from the MATH training split; all code experiments use a fixed pool of $D = 2{,}000$ problems from the LeetCodeDataset training split. For a thicket of size K, the pool is partitioned uniformly at random (partition seed $\in \{42, 8, 23\}$) into K disjoint shards of exactly $2000/K$ problems. Shards are sampled without replacement, so the union of the K shards is exactly the single adapter's training set.

Seeds. Each seed fixes the thicket's data partition, the data ordering (shuffle) for the single-adapter runs, and the sampling seeds of every evaluation pool; adapter initialization seeds are held fixed, so reported stds isolate data-allocation and sampling randomness.

GRPO training and checkpoints. All runs use the same verl-based GRPO implementation (Sheng et al., 2025; Shao et al., 2024) with group-normalized advantages, a binary verified reward, and identical hyperparameters (Table 4); the *only* difference between arms is the training file. The single-adapter runs train for ≈ 500 optimizer steps and checkpoint every 25; the GRPO-single (matched) control evaluates the last saved checkpoint at or before a thicket member's step count (steps $250/125/50/25$ for $K{=}2/4/8/16$, where members train 248–250, 124–125, 62–63, and 31–32 steps), and Figure 2 evaluates steps $\{25, 50, 125, 250, \text{final}\}$. Thicket member i trains for one epoch on its shard ($\approx 500/K$ steps).

Table 4: Shared GRPO/LoRA hyperparameters. The two training arms differ only in the training file.

LoRA rank / α / dropout	64 / 128 / 0.0	prompts per step	4
LoRA targets	q,k,v,o,gate,up,down proj.	rollouts per prompt	8
learning rate	2×10^{-5} (constant, 5% warmup)	rollout temperature	1.0
weight decay / grad. clip	0.01 / 0.1	max prompt / completion	1,024 / 2,048 tok.
KL coefficient	0.0	reward	binary verified
advantage	group-normalized (GRPO)	epochs	1

B EVALUATION PROTOCOL

Zero-shot, temperature 1.0, top-p 0.95, max 2,048 new tokens. *Math:* MATH500 (500 problems), GSM8K (1,319), OlympiadBench (674), Countdown (150), Knights & Knaves (300). *Code:* MBPP (427), HumanEval (164), LeetCodeDataset (228). Suite-level numbers are macro means of per-benchmark values, computed per seed and then averaged over the three seeds (except the $K{=}7$ ablation of Appendix G, seed 42).

The fixed 160-vote budget. Every (method, K) pool contains exactly 160 completions per problem, so all comparisons – across methods *and* across K – hold the inference budget fixed. A thicket fields K members, and each member contributes $160/K$ completions per problem ($80/40/20/10$ for $K{=}2/4/8/16$). A single policy – the base model, the final GRPO-single-adapter checkpoint, or the GRPO-single (matched) checkpoint (step $250/125/50/25$ for $K{=}2/4/8/16$) – is sampled once as a pool of 160 completions per problem, drawn as 10 requests of 16 parallel samples, and the K-voter view assigns these i.i.d. completions to K nominal voters of $160/K$ each. The majority vote and coverage of the base model and GRPO-single-adapter are therefore identical at every K (we verified this to machine precision), and only the pairwise error-correlation estimate depends, negligibly, on the grouping. Every completion in every pool is drawn with its own sampling seed.

The first-B convention. Section 4.5 and Appendix F vote over a prefix of each pool in generation order. At a budget of $B \in \{16, 32, \ldots, 160\}$ votes, each thicket member contributes its first B/K completions ($8/4/2/1$ at $B{=}16$ for $K{=}2/4/8/16$). A single policy contributes the completions drawn with its first B sampling seeds, and the K-voter view assigns them to K nominal voters. The majority vote and coverage of the base model and GRPO-single-adapter are therefore identical across K at every B (we verified this to machine precision). Every B is a multiple of 16, so B/K is a

whole number at every K, and $B{=}160$ is the full pool of the main tables. Each budget is one real deployment of that many samples, and smaller budgets are prefixes of larger ones.

Voting details. Completions from which no candidate answer can be parsed (*nulls*) are dropped before voting. Ties among plurality answers are scored exactly as the expected accuracy under uniform tie-breaking, removing tie-break randomness. Coverage counts a problem as covered if *any* pooled completion is graded correct.

Answer canonicalization. Votes are cast over canonical equivalence classes of answers rather than exact strings. On MATH500, GSM8K, and OlympiadBench we parse each extracted answer with `math_verify`: numeric answers are keyed by value (to 12 significant digits, so $0.5 \equiv 1/2 \equiv$ comma variants), and non-numeric answers (tuples, intervals, symbolic expressions) by their parse tree; unparseable strings fall back to a whitespace/LaTeX-normalized key. On Countdown we normalize notation only ($\times \equiv *$, $\div \equiv /$, spacing) – structurally different expressions are different solutions – and on Knights & Knaves the assignment is canonicalized order-insensitively. A canonical candidate is graded correct if any member completion was graded correct by the verifier.

Confidence-filtered voting details. $\bar{\ell}(r)$ is the mean per-token log-likelihood of response r under the model or adapter that generated it (available for math; code pools are not likelihood-scored). All four arms are scored on all three seeds at 160 votes, each against its own plain majority vote over the same completions. Within each problem, each voter ranks its own $160/K$ completions by $\bar{\ell}$ and keeps the top q fraction (at least one), with $q \in \{1, 1/2, 1/4, \text{top-1}\}$. Plain majority voting then runs over the surviving completions under the usual canonical-class, null-dropped, tie-marginalized protocol. q is selected per benchmark on four folds and applied to the held-out fold (fold = hash of the problem key). Exact training-fold ties are resolved toward the weaker filter, so $q{=}1$ (no filtering) is always available as a fallback. Because each nominal voter filters its own completions, the filtered vote of the base model and GRPO-single-adapter, unlike their plain vote, depends on K. Across the 1,200 selections per arm (3 seeds $\times$ 16 settings $\times$ 5 benchmarks $\times$ 5 folds), the CV picks a real filter ($q \neq 1$) in 67% of cases for the base model, 64% for the thicket, 59% for GRPO-single (matched), and 53% for GRPO-single-adapter. It picks the aggressive top-1-per-voter setting in at most 2.5% of any arm's selections (0.8% for the thicket). Confidence is informative, but the vote still wants breadth. Figure 9 visualizes the correct-vs-incorrect likelihood separation the filter exploits.

C Per-benchmark majority vote and confidence-filtered voting

Tables 5 and 6 break the majority vote of Table 1 down by benchmark, and Tables 7 and 8 do the same for confidence-filtered voting. Figure 6 plots the majority-vote comparison against voter count K (at a fixed 160 votes) with ± 1 std bands. Table 9 gives the thicket's confidence-filtered grid, and Tables 10–12 replicate its format for the base model, GRPO-single-adapter, and GRPO-single (matched). Figure 8 compares plain and confidence-filtered voting for all four methods, Figure 7 shows the most frequent CV-selected filter strength per setting, and Figure 9 shows the correct-vs-incorrect likelihood separation the filter exploits. All likelihood results are three-seed means, with the sample standard deviation after $\pm$.

Figure 5 reproduces the headline comparison of Figure 1 (right) — per-model majority-vote accuracy for all four methods on the math suite — at every voter count K, not only $K{=}16$.

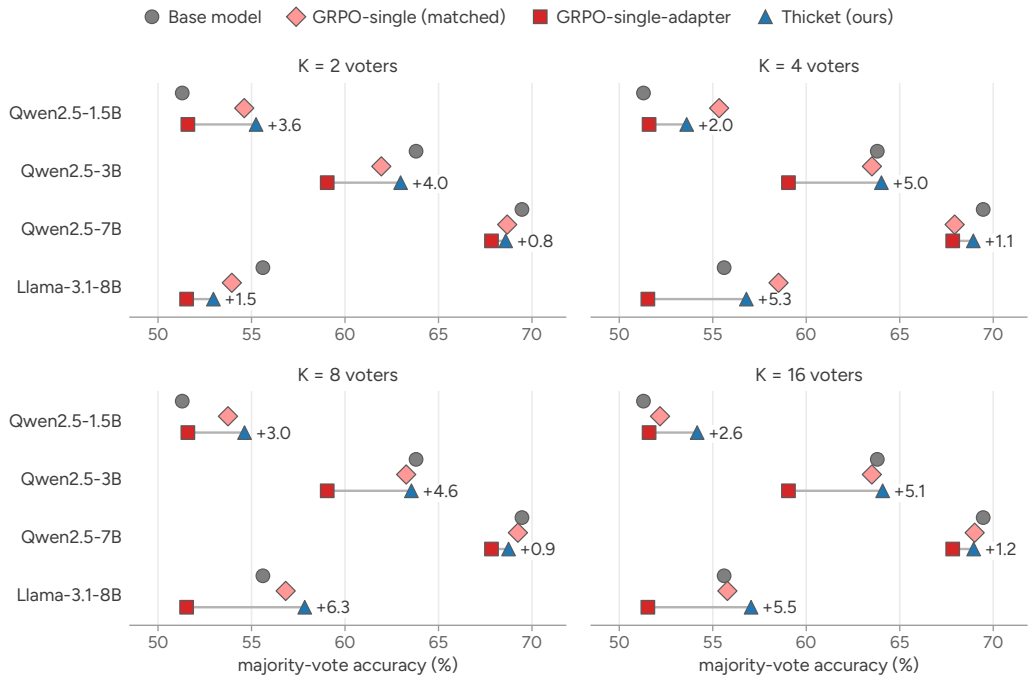


Figure 5: **The headline comparison at every voter count.** Per-model majority-vote accuracy on the math suite (macro mean over five benchmarks, mean over three seeds) for all four methods, at $K \in \{2, 4, 8, 16\}$. The K=16 panel reproduces the right panel of Figure 1. Number at the end of each red–blue connector: thicket − GRPO-single-adapter accuracy.

Table 5: Per-benchmark majority-vote accuracy (%) at $K \in \{2, 4\}$, fixed 160-vote pools; mean ± std over three seeds. Mean rows are macro means. Bold: best method per row and K; underlined: second best.

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark (K=)	2	4	2	4	2	4	2	4
Qwen2.5-1.5B	MATH500	67.7 ±0.3	67.7 ±0.3	68.5 ±1.3	**69.9 ±1.0**	67.5 ±1.7	67.5 ±1.7	**68.6 ±1.0**	69.1 ±0.6
	GSM8K	**86.4 ±0.4**	**86.4 ±0.4**	84.1 ±0.3	85.1 ±0.3	82.4 ±1.1	82.4 ±1.1	84.6 ±0.1	85.1 ±0.4
	OlympiadBench	32.5 ±0.4	32.5 ±0.4	33.6 ±0.5	33.7 ±1.3	31.9 ±0.3	31.9 ±0.3	**34.1 ±0.4**	**34.3 ±0.8**
	Countdown	39.8 ±1.9	39.8 ±1.9	51.8 ±2.2	**54.7 ±8.0**	42.4 ±10.5	42.4 ±10.5	**53.8 ±9.8**	47.8 ±3.2
	Knights & Knaves	30.1 ±0.5	30.1 ±0.5	35.0 ±4.1	33.3 ±1.0	33.7 ±1.7	**33.7 ±1.7**	**35.1 ±1.6**	31.7 ±0.5
	Mean	51.3 ±0.4	51.3 ±0.4	54.6 ±0.7	**55.3 ±2.0**	51.6 ±1.6	51.6 ±1.6	**55.2 ±2.0**	53.6 ±0.7
Qwen2.5-3B	MATH500	76.3 ±0.6	76.3 ±0.6	75.2 ±0.9	76.3 ±0.5	74.6 ±2.2	74.6 ±2.2	**77.3 ±0.7**	**77.7 ±0.9**
	GSM8K	**91.7 ±0.1**	**91.7 ±0.1**	90.7 ±0.2	91.3 ±0.3	89.9 ±0.4	89.9 ±0.4	91.0 ±0.6	91.4 ±0.2
	OlympiadBench	**44.1 ±0.2**	**44.1 ±0.2**	42.7 ±0.7	42.6 ±0.9	42.2 ±2.1	42.2 ±2.1	44.0 ±0.5	42.9 ±0.8
	Countdown	**51.9 ±1.3**	51.9 ±1.3	46.2 ±4.5	50.8 ±3.5	35.7 ±18.1	35.7 ±18.1	47.6 ±6.1	**52.6 ±2.6**
	Knights & Knaves	**55.1 ±1.9**	55.1 ±1.9	54.9 ±1.0	**56.6 ±1.3**	52.9 ±3.2	52.9 ±3.2	55.0 ±1.0	55.5 ±2.1
	Mean	**63.8 ±0.3**	63.8 ±0.3	61.9 ±1.2	63.5 ±1.2	59.0 ±5.1	59.0 ±5.1	63.0 ±1.2	**64.0 ±0.4**
Qwen2.5-7B	MATH500	**83.7 ±0.7**	**83.7 ±0.7**	83.5 ±0.6	83.3 ±1.1	83.2 ±1.1	83.2 ±1.1	83.6 ±0.8	83.5 ±1.0
	GSM8K	**94.8 ±0.0**	**94.8 ±0.0**	94.4 ±0.2	94.3 ±0.1	94.1 ±0.3	94.1 ±0.3	94.3 ±0.1	94.6 ±0.2
	OlympiadBench	53.0 ±0.3	**53.0 ±0.3**	52.8 ±1.6	52.8 ±0.4	50.7 ±1.5	50.7 ±1.5	**53.1 ±0.7**	52.7 ±0.2
	Countdown	**56.3 ±1.5**	**56.3 ±1.5**	51.9 ±1.6	49.7 ±3.5	51.4 ±2.2	51.4 ±2.2	50.1 ±5.1	52.2 ±1.0
	Knights & Knaves	59.5 ±1.1	59.5 ±1.1	60.8 ±2.8	59.7 ±1.0	59.8 ±2.2	59.8 ±2.2	**61.8 ±2.8**	**61.7 ±0.3**
	Mean	**69.5 ±0.2**	**69.5 ±0.2**	68.7 ±1.1	67.9 ±1.2	67.8 ±1.2	67.8 ±1.2	68.6 ±1.1	68.9 ±0.4
Llama-3.1-8B	MATH500	**66.2 ±1.0**	66.2 ±1.0	62.1 ±3.3	**66.6 ±0.8**	60.3 ±1.0	60.3 ±1.0	64.8 ±3.0	66.2 ±0.3
	GSM8K	**92.2 ±0.4**	**92.2 ±0.4**	90.8 ±0.8	91.0 ±1.0	89.3 ±0.8	89.3 ±0.8	90.4 ±0.7	91.2 ±0.7
	OlympiadBench	32.9 ±1.1	32.9 ±1.1	31.6 ±2.5	35.4 ±0.9	30.0 ±0.4	30.0 ±0.4	**33.6 ±1.6**	**35.6 ±0.9**
	Countdown	**46.8 ±2.1**	46.8 ±2.1	43.7 ±3.9	**56.9 ±6.6**	38.4 ±15.8	38.4 ±15.8	31.6 ±16.6	48.9 ±10.3
	Knights & Knaves	39.9 ±0.3	39.9 ±0.3	41.6 ±2.4	**42.7 ±0.6**	39.6 ±5.1	39.6 ±5.1	**44.4 ±1.5**	42.1 ±1.4
	Mean	**55.6 ±0.3**	55.6 ±0.3	53.9 ±1.0	**58.5 ±1.5**	51.5 ±3.2	51.5 ±3.2	53.0 ±4.2	56.8 ±2.1

Table 6: Per-benchmark majority-vote accuracy (%) at $K \in \{8, 16\}$, fixed 160-vote pools (format of Table 5).

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark (K=)	8	16	8	16	8	16	8	16
Qwen2.5-1.5B	MATH500	67.7 ±0.3	67.7 ±0.3	68.7 ±0.2	67.8 ±0.2	67.5 ±1.7	67.5 ±1.7	**69.0 ±0.9**	**68.8 ±1.0**
	GSM8K	**86.4 ±0.4**	**86.4 ±0.4**	86.0 ±0.2	86.2 ±0.3	82.4 ±1.1	82.4 ±1.1	85.6 ±0.2	85.9 ±0.1
	OlympiadBench	32.5 ±0.4	32.5 ±0.4	32.9 ±0.9	32.2 ±0.3	31.9 ±0.3	31.9 ±0.3	**33.2 ±0.4**	**33.9 ±0.7**
	Countdown	39.8 ±1.9	39.8 ±1.9	48.5 ±1.0	45.0 ±3.5	42.4 ±10.5	42.4 ±10.5	**53.6 ±0.3**	**49.6 ±0.6**
	Knights & Knaves	30.1 ±0.5	30.1 ±0.5	32.7 ±0.3	29.7 ±1.3	**33.7 ±1.7**	**33.7 ±1.7**	31.7 ±2.1	32.6 ±1.7
	Mean	51.3 ±0.4	51.3 ±0.4	53.7 ±0.2	52.2 ±0.9	51.6 ±1.6	51.6 ±1.6	**54.6 ±0.5**	**54.2 ±0.2**
Qwen2.5-3B	MATH500	76.3 ±0.6	76.3 ±0.6	**76.6 ±0.7**	76.8 ±0.5	74.6 ±2.2	74.6 ±2.2	**76.6 ±0.2**	**77.1 ±0.4**
	GSM8K	**91.7 ±0.1**	91.7 ±0.1	91.6 ±0.4	**91.8 ±0.2**	89.9 ±0.4	89.9 ±0.4	91.5 ±0.1	91.7 ±0.3
	OlympiadBench	**44.1 ±0.2**	**44.1 ±0.2**	43.3 ±0.9	43.5 ±0.6	42.2 ±2.1	42.2 ±2.1	43.5 ±0.1	43.1 ±0.6
	Countdown	**51.9 ±1.3**	51.9 ±1.3	51.3 ±1.3	51.1 ±2.0	35.7 ±18.1	35.7 ±18.1	51.2 ±1.7	**52.3 ±2.3**
	Knights & Knaves	**55.1 ±1.9**	55.1 ±1.9	53.7 ±0.7	54.4 ±1.1	52.9 ±3.2	52.9 ±3.2	54.9 ±1.6	**56.3 ±0.7**
	Mean	**63.8 ±0.3**	63.8 ±0.3	63.3 ±0.1	63.5 ±0.8	59.0 ±5.1	59.0 ±5.1	63.6 ±0.2	**64.1 ±0.4**
Qwen2.5-7B	MATH500	**83.7 ±0.7**	**83.7 ±0.7**	83.2 ±0.3	83.3 ±0.1	83.2 ±1.1	83.2 ±1.1	83.4 ±0.3	83.5 ±0.6
	GSM8K	**94.8 ±0.0**	**94.8 ±0.0**	94.5 ±0.2	94.7 ±0.1	94.1 ±0.3	94.1 ±0.3	94.4 ±0.1	94.5 ±0.1
	OlympiadBench	53.0 ±0.3	**53.0 ±0.3**	**53.3 ±0.6**	52.7 ±0.6	50.7 ±1.5	50.7 ±1.5	52.4 ±0.4	52.3 ±0.9
	Countdown	**56.3 ±1.5**	**56.3 ±1.5**	55.2 ±1.0	55.0 ±0.9	51.4 ±2.2	51.4 ±2.2	54.0 ±4.1	53.7 ±2.2
	Knights & Knaves	59.5 ±1.1	59.5 ±1.1	**60.1 ±1.0**	59.2 ±0.3	59.8 ±2.2	59.8 ±2.2	59.6 ±1.5	**60.7 ±1.4**
	Mean	**69.5 ±0.2**	**69.5 ±0.2**	69.2 ±0.1	69.0 ±0.2	67.8 ±1.2	67.8 ±1.2	68.7 ±1.0	69.0 ±0.4
Llama-3.1-8B	MATH500	66.2 ±1.0	66.2 ±1.0	**66.8 ±0.3**	66.8 ±0.6	60.3 ±1.0	60.3 ±1.0	**66.8 ±0.9**	**67.4 ±0.7**
	GSM8K	**92.2 ±0.4**	**92.2 ±0.4**	91.7 ±0.1	92.1 ±0.3	89.3 ±0.8	89.3 ±0.8	91.8 ±0.2	91.8 ±0.1
	OlympiadBench	32.9 ±1.1	32.9 ±1.1	33.8 ±0.5	33.7 ±0.5	30.0 ±0.4	30.0 ±0.4	**35.0 ±1.0**	**34.4 ±1.1**
	Countdown	46.8 ±2.1	46.8 ±2.1	50.0 ±1.6	47.2 ±3.9	38.4 ±15.8	38.4 ±15.8	**54.0 ±3.1**	**51.3 ±2.4**
	Knights & Knaves	39.9 ±0.3	39.9 ±0.3	**41.7 ±1.2**	39.1 ±1.0	39.6 ±5.1	39.6 ±5.1	41.5 ±2.5	**40.3 ±1.3**
	Mean	55.6 ±0.3	55.6 ±0.3	56.8 ±0.5	55.8 ±0.6	51.5 ±3.2	51.5 ±3.2	**57.8 ±1.5**	**57.0 ±0.7**

Table 7: Per-benchmark accuracy (%) under confidence-filtered voting, all four methods, $K \in \{2, 4\}$; fixed 160-vote pools; mean ± std over three seeds. Bold: best method per row and K; underlined: second best.

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark (K=)	2	4	2	4	2	4	2	4
Qwen2.5-1.5B	MATH500	69.7 ±0.5	69.5 ±0.6	69.6 ±1.0	71.4 ±1.0	68.1 ±1.3	67.1 ±0.7	**70.9 ±0.4**	**71.9 ±0.8**
	GSM8K	**86.4 ±0.3**	**86.4 ±0.2**	84.1 ±0.3	85.0 ±0.3	82.4 ±1.1	82.4 ±1.2	85.0 ±0.4	85.0 ±0.2
	OlympiadBench	**35.9 ±0.3**	35.4 ±0.5	35.4 ±1.3	36.2 ±1.7	31.9 ±1.3	32.5 ±0.9	35.4 ±1.1	**36.6 ±0.6**
	Countdown	38.0 ±2.8	38.2 ±2.7	51.3 ±3.2	**54.7 ±8.0**	41.7 ±10.9	42.4 ±10.5	**53.8 ±9.8**	47.8 ±3.2
	Knights & Knaves	28.5 ±0.6	29.4 ±0.6	34.9 ±2.3	33.0 ±1.3	34.0 ±3.0	**33.2 ±1.5**	**36.2 ±1.5**	32.2 ±0.4
	Mean	51.7 ±0.6	51.8 ±0.4	55.0 ±0.0	**56.1 ±2.0**	51.6 ±1.7	51.5 ±1.8	**56.3 ±1.7**	54.7 ±0.9
Qwen2.5-3B	MATH500	**78.9 ±0.4**	**78.8 ±0.4**	77.2 ±0.3	76.6 ±0.7	74.4 ±1.5	74.3 ±2.0	77.9 ±0.3	78.7 ±1.1
	GSM8K	**91.6 ±0.5**	**91.6 ±0.2**	90.9 ±0.3	91.1 ±0.6	90.0 ±0.3	89.8 ±0.5	91.1 ±0.6	91.5 ±0.2
	OlympiadBench	**45.7 ±1.3**	**45.6 ±0.9**	44.3 ±0.7	44.4 ±1.4	42.3 ±1.9	42.3 ±2.4	45.3 ±1.4	45.3 ±0.6
	Countdown	**51.7 ±1.5**	51.0 ±1.5	46.2 ±4.5	50.8 ±3.5	35.7 ±18.1	35.7 ±18.1	47.6 ±6.1	**52.6 ±2.6**
	Knights & Knaves	54.2 ±3.4	54.4 ±3.0	**54.6 ±2.8**	**56.0 ±2.2**	53.2 ±5.0	52.5 ±5.2	53.7 ±1.3	55.3 ±3.7
	Mean	**64.4 ±0.4**	64.3 ±0.1	62.6 ±1.3	63.8 ±1.6	59.1 ±5.3	58.9 ±5.5	63.1 ±1.7	**64.7 ±0.4**
Qwen2.5-7B	MATH500	**85.0 ±0.3**	**84.8 ±0.2**	84.3 ±0.8	83.8 ±0.7	83.2 ±1.2	83.4 ±1.2	84.4 ±0.7	84.3 ±1.1
	GSM8K	**94.8 ±0.0**	**94.8 ±0.0**	94.4 ±0.1	94.3 ±0.3	94.0 ±0.3	94.1 ±0.4	94.3 ±0.1	94.5 ±0.1
	OlympiadBench	**53.3 ±0.9**	**53.9 ±0.4**	52.3 ±1.5	53.4 ±1.2	50.4 ±0.9	50.9 ±1.3	52.3 ±0.5	52.3 ±0.1
	Countdown	**56.3 ±1.5**	**56.3 ±1.5**	51.1 ±3.0	49.7 ±3.5	51.4 ±2.2	51.4 ±2.2	50.1 ±5.1	52.2 ±1.0
	Knights & Knaves	60.1 ±1.0	60.6 ±1.9	**61.9 ±3.5**	59.2 ±3.0	58.8 ±1.8	58.9 ±1.9	61.1 ±3.4	**61.6 ±1.5**
	Mean	**69.9 ±0.2**	**70.1 ±0.3**	68.8 ±1.6	68.1 ±1.7	67.6 ±1.1	67.8 ±1.2	68.5 ±1.0	69.0 ±0.1
Llama-3.1-8B	MATH500	**67.3 ±0.8**	67.3 ±0.2	64.2 ±2.7	**67.5 ±0.5**	61.4 ±0.4	61.2 ±1.0	66.6 ±3.2	67.0 ±1.5
	GSM8K	**92.6 ±0.3**	**92.6 ±0.3**	90.7 ±1.0	91.1 ±0.8	89.7 ±0.8	89.8 ±0.9	90.9 ±1.0	91.3 ±0.5
	OlympiadBench	33.2 ±0.5	33.5 ±0.4	32.4 ±3.5	36.0 ±0.8	30.5 ±0.4	30.2 ±0.7	**34.2 ±2.0**	**36.3 ±0.8**
	Countdown	**48.0 ±0.6**	48.1 ±0.8	43.7 ±3.9	**58.4 ±7.5**	37.9 ±15.4	38.1 ±15.6	30.8 ±17.2	48.8 ±10.2
	Knights & Knaves	40.0 ±2.3	40.5 ±0.8	40.7 ±2.9	**42.7 ±0.6**	38.8 ±5.4	38.1 ±5.7	**44.9 ±0.5**	42.1 ±2.2
	Mean	**56.2 ±0.7**	56.4 ±0.0	54.3 ±1.4	**59.1 ±1.3**	51.6 ±3.2	51.5 ±3.3	53.5 ±4.4	57.1 ±1.2

Table 8: Per-benchmark accuracy (%) under confidence-filtered voting, $K \in \{8, 16\}$ (format of Table 7).

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark ($K=$)	8	16	8	16	8	16	8	16
Qwen2.5-1.5B	MATH500	69.9 ±0.6	69.5 ±0.4	70.9 ±0.2	**70.8 ±0.7**	67.3 ±1.0	67.2 ±0.7	**71.0 ±0.4**	70.0 ±0.7
	GSM8K	**86.1 ±0.2**	**86.3 ±0.4**	85.8 ±0.2	**86.3 ±0.2**	82.4 ±1.1	82.4 ±1.1	85.3 ±0.5	**86.3 ±0.6**
	OlympiadBench	35.8 ±0.4	34.7 ±0.8	35.2 ±0.6	34.5 ±0.7	32.6 ±1.0	32.8 ±1.4	**36.6 ±1.0**	**35.3 ±0.2**
	Countdown	38.8 ±2.1	38.1 ±3.0	47.6 ±2.4	45.0 ±3.5	42.4 ±10.5	42.4 ±10.5	**53.6 ±0.3**	**49.6 ±0.6**
	Knights & Knaves	29.4 ±1.1	29.4 ±0.8	31.6 ±0.6	28.6 ±2.6	**32.8 ±2.6**	**34.4 ±2.1**	31.1 ±1.6	32.6 ±1.7
	Mean	52.0 ±0.4	51.6 ±0.3	54.2 ±0.4	53.1 ±0.7	51.5 ±1.7	51.8 ±1.6	**55.5 ±0.5**	**54.8 ±0.2**
Qwen2.5-3B	MATH500	**78.4 ±0.8**	**78.5 ±1.6**	77.5 ±0.7	78.0 ±0.6	74.5 ±1.7	74.6 ±2.0	77.9 ±1.4	77.8 ±1.0
	GSM8K	**91.6 ±0.1**	91.4 ±0.2	**91.6 ±0.2**	**91.6 ±0.3**	89.7 ±0.5	89.8 ±0.6	91.5 ±0.2	**91.6 ±0.2**
	OlympiadBench	**46.4 ±0.6**	**45.5 ±0.8**	45.7 ±0.8	45.4 ±0.6	42.6 ±2.1	42.4 ±2.0	46.0 ±0.4	44.6 ±0.4
	Countdown	**52.1 ±0.8**	51.5 ±1.9	51.3 ±1.3	50.6 ±2.6	35.7 ±18.1	35.7 ±18.1	51.2 ±1.7	**52.3 ±2.3**
	Knights & Knaves	**54.2 ±3.5**	54.0 ±3.7	53.9 ±0.7	53.7 ±1.3	53.2 ±5.2	53.3 ±4.8	**54.2 ±1.6**	**56.3 ±0.7**
	Mean	**64.6 ±0.4**	64.2 ±0.4	64.0 ±0.4	63.9 ±0.8	59.2 ±5.4	59.2 ±5.5	64.2 ±0.2	**64.5 ±0.4**
Qwen2.5-7B	MATH500	**84.9 ±0.7**	**84.5 ±0.8**	84.7 ±0.7	83.7 ±0.8	83.5 ±0.7	83.3 ±1.1	84.5 ±0.7	84.4 ±0.7
	GSM8K	**94.8 ±0.0**	**94.8 ±0.0**	94.5 ±0.2	94.7 ±0.2	94.1 ±0.5	94.1 ±0.5	94.4 ±0.1	94.4 ±0.2
	OlympiadBench	**53.3 ±0.5**	**53.7 ±0.7**	**53.3 ±0.3**	53.3 ±0.1	51.1 ±1.2	50.3 ±1.2	52.4 ±0.8	52.7 ±1.1
	Countdown	**56.3 ±1.5**	**56.3 ±1.5**	55.2 ±1.0	55.0 ±0.9	51.4 ±2.2	51.4 ±2.2	54.0 ±4.1	53.7 ±2.2
	Knights & Knaves	60.5 ±2.0	59.9 ±0.9	60.1 ±1.6	59.1 ±1.6	59.1 ±1.6	59.3 ±1.7	**61.4 ±1.5**	**61.6 ±2.9**
	Mean	**70.0 ±0.6**	**69.8 ±0.2**	69.6 ±0.6	69.2 ±0.3	67.8 ±1.0	67.7 ±1.0	69.3 ±1.0	69.4 ±0.4
Llama-3.1-8B	MATH500	67.6 ±0.2	67.3 ±0.7	67.3 ±1.8	67.5 ±0.2	61.3 ±0.5	61.1 ±0.3	**67.8 ±0.9**	**68.6 ±0.6**
	GSM8K	**92.6 ±0.4**	**92.4 ±0.2**	91.7 ±0.4	92.3 ±0.1	89.6 ±0.8	89.7 ±0.6	91.6 ±0.4	91.8 ±0.3
	OlympiadBench	33.4 ±0.3	33.3 ±0.8	34.5 ±0.8	33.7 ±0.6	30.6 ±0.7	30.5 ±0.3	**36.3 ±1.1**	**35.6 ±0.3**
	Countdown	47.9 ±1.0	47.0 ±2.0	49.7 ±0.8	46.6 ±4.9	37.4 ±15.0	37.7 ±15.2	**55.0 ±1.7**	**51.5 ±3.7**
	Knights & Knaves	38.8 ±2.6	39.8 ±0.3	41.5 ±1.3	39.6 ±1.1	39.4 ±5.4	38.0 ±5.4	**41.6 ±2.3**	**40.5 ±0.7**
	Mean	56.0 ±0.5	55.9 ±0.5	56.9 ±0.9	55.9 ±1.1	51.6 ±3.1	51.4 ±3.1	**58.4 ±1.1**	**57.6 ±0.7**

Table 9: **Thicket accuracy under confidence-filtered voting** (each voter keeps its top-q completions by mean log-likelihood, q chosen by per-benchmark five-fold CV; fixed 160-vote pools; mean ± std over three seeds). Mean column: macro over the five benchmarks. Bold: the displayed value improves on the same setting's plain majority vote.

Model	K	MATH500	GSM8K	OlympiadBench	Countdown	Knights & Knaves	*Mean*
Qwen2.5-1.5B	2	**70.9 ±0.4**	**85.0 ±0.4**	**35.4 ±1.1**	53.8 ±9.8	**36.2 ±1.5**	**56.3 ±1.7**
	4	**71.9 ±0.8**	85.0 ±0.2	**36.6 ±0.6**	47.8 ±3.2	**32.2 ±0.4**	**54.7 ±0.9**
	8	**71.0 ±0.4**	85.3 ±0.5	**36.6 ±1.0**	53.6 ±0.3	31.1 ±1.6	**55.5 ±0.5**
	16	**70.0 ±0.7**	**86.3 ±0.6**	**35.3 ±0.2**	49.6 ±0.6	32.6 ±1.7	**54.8 ±0.2**
Qwen2.5-3B	2	**77.9 ±0.3**	**91.1 ±0.6**	**45.3 ±1.4**	47.6 ±6.1	53.7 ±1.3	**63.1 ±1.7**
	4	**78.7 ±1.1**	**91.5 ±0.2**	**45.3 ±0.6**	52.6 ±2.6	55.3 ±3.7	**64.7 ±0.4**
	8	**77.9 ±1.4**	91.5 ±0.2	**46.0 ±0.4**	51.2 ±1.7	54.2 ±1.6	**64.2 ±0.2**
	16	**77.8 ±1.0**	91.6 ±0.2	**44.6 ±0.4**	52.3 ±2.3	56.3 ±0.7	**64.5 ±0.4**
Qwen2.5-7B	2	**84.4 ±0.7**	94.3 ±0.1	52.3 ±0.5	50.1 ±5.1	61.1 ±3.4	68.5 ±1.0
	4	**84.3 ±1.1**	94.5 ±0.1	52.3 ±0.1	52.2 ±1.0	61.6 ±1.5	**69.0 ±0.1**
	8	**84.5 ±0.7**	94.4 ±0.1	52.4 ±0.8	54.0 ±4.1	**61.4 ±1.5**	**69.3 ±1.0**
	16	**84.4 ±0.7**	94.4 ±0.2	**52.7 ±1.1**	53.7 ±2.2	**61.6 ±2.9**	**69.4 ±0.4**
Llama-3.1-8B	2	**66.6 ±3.2**	**90.9 ±1.0**	**34.2 ±2.0**	30.8 ±17.2	**44.9 ±0.5**	**53.5 ±4.4**
	4	**67.0 ±1.5**	**91.3 ±0.5**	**36.3 ±0.8**	48.8 ±10.2	42.1 ±2.2	**57.1 ±1.2**
	8	**67.8 ±0.9**	91.6 ±0.4	**36.3 ±1.1**	**55.0 ±1.7**	**41.6 ±2.3**	**58.4 ±1.1**
	16	**68.6 ±0.6**	91.8 ±0.3	**35.6 ±0.3**	**51.5 ±3.7**	**40.5 ±0.7**	**57.6 ±0.7**

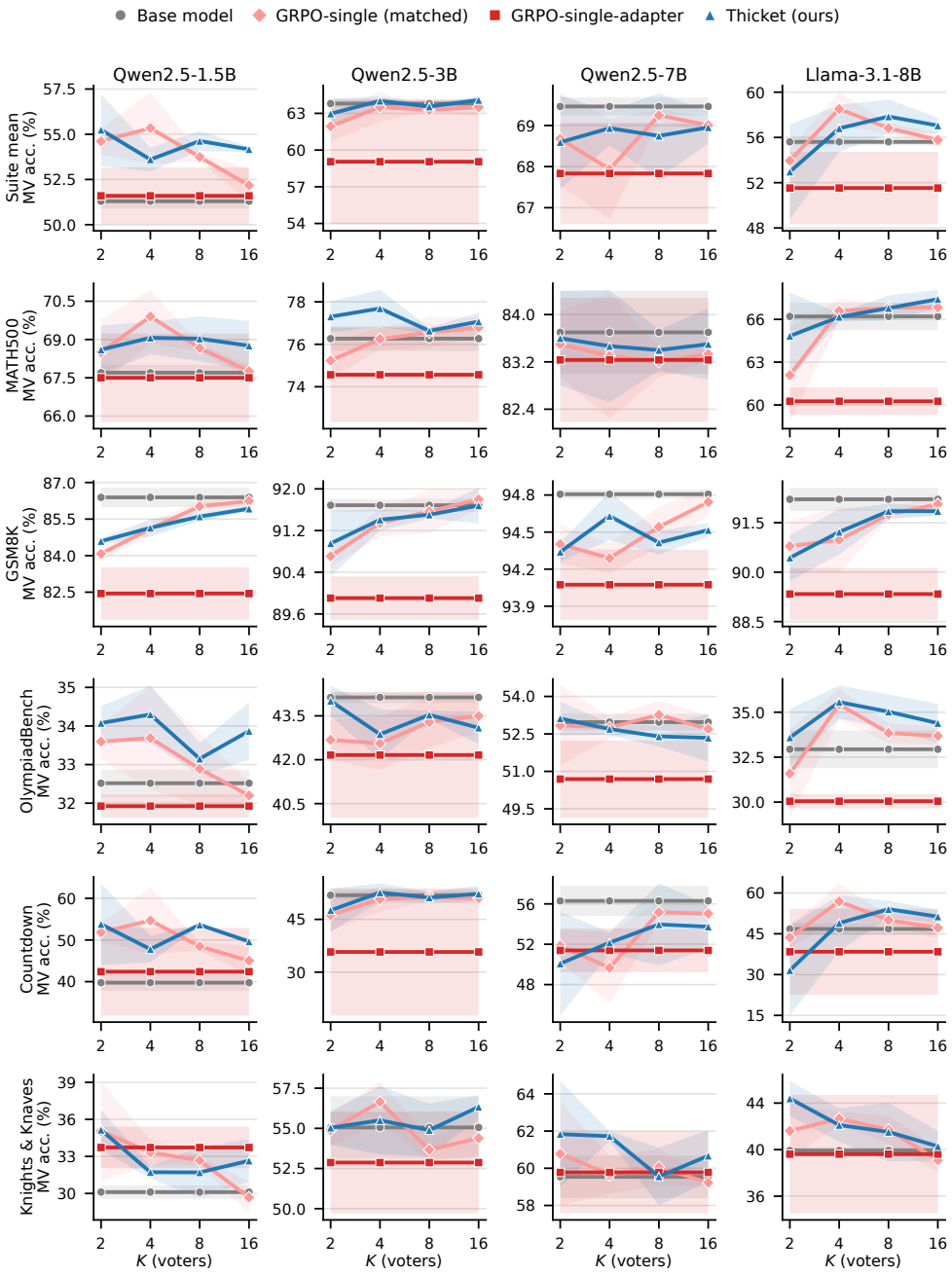


Figure 6: Majority-vote accuracy vs. voter count K (fixed 160-vote budget at every K): suite macro (top row) and each benchmark, all models (columns). Lines: seed means; bands: ± 1 std. Base and GRPO-single-adapter are flat by construction (one 160-completion pool regrouped at every K).

Table 10: Base-model accuracy under confidence-filtered voting (format of Table 9; three seeds).

Model	K	MATH500	GSM8K	OlympiadBench	Countdown	Knights & Knaves	*Mean*
Qwen2.5-1.5B	2	**69.7** ±0.5	86.4 ±0.3	**35.9** ±0.3	38.0 ±2.8	28.5 ±0.6	**51.7** ±0.6
	4	**69.5** ±0.6	86.4 ±0.2	**35.4** ±0.5	38.2 ±2.7	29.4 ±0.6	**51.8** ±0.4
	8	**69.9** ±0.6	86.1 ±0.2	**35.8** ±0.4	38.8 ±2.1	29.4 ±1.1	**52.0** ±0.4
	16	**69.5** ±0.4	86.3 ±0.4	**34.7** ±0.8	38.1 ±3.0	29.4 ±0.8	**51.6** ±0.3
Qwen2.5-3B	2	**78.9** ±0.4	91.6 ±0.5	**45.7** ±1.3	51.7 ±1.5	54.2 ±3.4	**64.4** ±0.4
	4	**78.8** ±0.4	91.6 ±0.2	**45.6** ±0.9	51.0 ±1.5	54.4 ±3.0	**64.3** ±0.1
	8	**78.4** ±0.8	91.6 ±0.1	**46.4** ±0.6	**52.1** ±0.8	54.2 ±3.5	**64.6** ±0.4
	16	**78.5** ±1.6	91.4 ±0.2	**45.5** ±0.8	51.5 ±1.9	54.0 ±3.7	**64.2** ±0.4
Qwen2.5-7B	2	**85.0** ±0.3	94.8 ±0.0	**53.3** ±0.9	56.3 ±1.5	**60.1** ±1.0	**69.9** ±0.2
	4	**84.8** ±0.2	94.8 ±0.0	**53.9** ±0.4	56.3 ±1.5	**60.6** ±1.9	**70.1** ±0.3
	8	**84.9** ±0.7	94.8 ±0.0	**53.3** ±0.5	56.3 ±1.5	**60.5** ±2.0	**70.0** ±0.6
	16	**84.5** ±0.8	94.8 ±0.0	**53.7** ±0.7	56.3 ±1.5	**59.9** ±0.9	**69.8** ±0.2
Llama-3.1-8B	2	**67.3** ±0.8	**92.6** ±0.3	**33.2** ±0.5	**48.0** ±0.6	**40.0** ±2.3	**56.2** ±0.7
	4	**67.3** ±0.2	**92.6** ±0.3	**33.5** ±0.4	**48.1** ±0.8	**40.5** ±0.8	**56.4** ±0.0
	8	**67.6** ±0.2	**92.6** ±0.4	**33.4** ±0.3	**47.9** ±1.0	38.8 ±2.6	**56.0** ±0.5
	16	**67.3** ±0.7	**92.4** ±0.2	**33.3** ±0.8	**47.0** ±2.0	39.8 ±0.3	**55.9** ±0.5

Table 11: GRPO-single-adapter accuracy under confidence-filtered voting (format of Table 9; three seeds).

Model	K	MATH500	GSM8K	OlympiadBench	Countdown	Knights & Knaves	*Mean*
Qwen2.5-1.5B	2	**68.1** ±1.3	82.4 ±1.1	31.9 ±1.3	41.7 ±10.9	**34.0** ±3.0	51.6 ±1.7
	4	67.1 ±0.7	82.4 ±1.2	**32.5** ±0.9	42.4 ±10.5	33.2 ±1.5	51.5 ±1.8
	8	67.3 ±1.0	82.4 ±1.1	**32.6** ±1.0	42.4 ±10.5	32.8 ±2.6	51.5 ±1.7
	16	67.2 ±0.7	82.4 ±1.1	**32.8** ±1.4	42.4 ±10.5	**34.4** ±2.1	**51.8** ±1.6
Qwen2.5-3B	2	74.4 ±1.5	**90.0** ±0.3	**42.3** ±1.9	35.7 ±18.1	**53.2** ±5.0	**59.1** ±5.3
	4	74.3 ±2.0	89.8 ±0.5	**42.3** ±2.4	35.7 ±18.1	52.5 ±5.2	58.9 ±5.5
	8	74.5 ±1.7	89.7 ±0.5	**42.6** ±2.1	35.7 ±18.1	**53.2** ±5.2	**59.2** ±5.4
	16	74.6 ±2.0	89.8 ±0.6	**42.4** ±2.0	35.7 ±18.1	**53.3** ±4.8	**59.2** ±5.5
Qwen2.5-7B	2	83.2 ±1.2	94.0 ±0.3	50.4 ±0.9	51.4 ±2.2	58.8 ±1.8	67.6 ±1.1
	4	**83.4** ±1.2	94.1 ±0.4	**50.9** ±1.3	51.4 ±2.2	58.9 ±1.9	67.8 ±1.2
	8	**83.5** ±0.7	94.1 ±0.5	**51.1** ±1.2	51.4 ±2.2	59.1 ±1.6	67.8 ±1.0
	16	**83.3** ±1.1	94.1 ±0.5	50.3 ±1.2	51.4 ±2.2	59.3 ±1.7	67.7 ±1.0
Llama-3.1-8B	2	**61.4** ±0.4	**89.7** ±0.8	**30.5** ±0.4	37.9 ±15.4	38.8 ±5.4	**51.6** ±3.2
	4	**61.2** ±1.0	**89.8** ±0.9	**30.2** ±0.7	38.1 ±15.6	38.1 ±5.7	51.5 ±3.3
	8	**61.3** ±0.5	**89.6** ±0.8	**30.6** ±0.7	37.4 ±15.0	39.4 ±5.4	**51.6** ±3.1
	16	**61.1** ±0.3	**89.7** ±0.6	**30.5** ±0.3	37.7 ±15.2	38.0 ±5.4	51.4 ±3.1

Table 12: GRPO-single (matched) accuracy under confidence-filtered voting (format of Table 9; three seeds).

Model	K	MATH500	GSM8K	OlympiadBench	Countdown	Knights & Knaves	*Mean*
Qwen2.5-1.5B	2	**69.6** ±1.0	84.1 ±0.3	**35.4** ±1.3	51.3 ±3.2	34.9 ±2.3	**55.0** ±0.0
	4	**71.4** ±1.0	85.0 ±0.3	**36.2** ±1.7	54.7 ±8.0	33.0 ±1.3	**56.1** ±2.0
	8	**70.9** ±0.2	85.8 ±0.2	**35.2** ±0.6	47.6 ±2.4	31.6 ±0.6	**54.2** ±0.4
	16	**70.8** ±0.7	**86.3** ±0.2	**34.5** ±0.7	45.0 ±3.5	28.6 ±2.6	**53.1** ±0.7
Qwen2.5-3B	2	**77.2** ±0.3	**90.9** ±0.3	**44.3** ±0.7	46.2 ±4.5	54.6 ±2.8	**62.6** ±1.3
	4	**76.6** ±0.7	91.1 ±0.6	**44.4** ±1.4	50.8 ±3.5	56.0 ±2.2	**63.8** ±1.6
	8	**77.5** ±0.7	91.6 ±0.2	**45.7** ±0.8	51.3 ±1.3	**53.9** ±0.7	**64.0** ±0.4
	16	**78.0** ±0.6	91.6 ±0.3	**45.4** ±0.6	50.6 ±2.6	53.7 ±1.3	**63.9** ±0.8
Qwen2.5-7B	2	**84.3** ±0.8	94.4 ±0.1	52.3 ±1.5	51.1 ±3.0	**61.9** ±3.5	**68.8** ±1.6
	4	**83.8** ±0.7	94.3 ±0.3	**53.4** ±1.2	49.7 ±3.5	59.2 ±3.0	**68.1** ±1.7
	8	**84.7** ±0.7	94.5 ±0.2	53.3 ±0.3	55.2 ±1.0	60.1 ±1.6	**69.6** ±0.6
	16	**83.7** ±0.8	94.7 ±0.2	**53.3** ±0.1	55.0 ±0.9	59.1 ±1.6	**69.2** ±0.3
Llama-3.1-8B	2	**64.2** ±2.7	90.7 ±1.0	**32.4** ±3.5	43.7 ±3.9	40.7 ±2.9	**54.3** ±1.4
	4	**67.5** ±0.5	**91.1** ±0.8	**36.0** ±0.8	**58.4** ±7.5	42.7 ±0.6	**59.1** ±1.3
	8	**67.3** ±1.8	91.7 ±0.4	**34.5** ±0.8	49.7 ±0.8	41.5 ±1.3	**56.9** ±0.9
	16	**67.5** ±0.2	**92.3** ±0.1	33.7 ±0.6	46.6 ±4.9	**39.6** ±1.1	**55.9** ±1.1

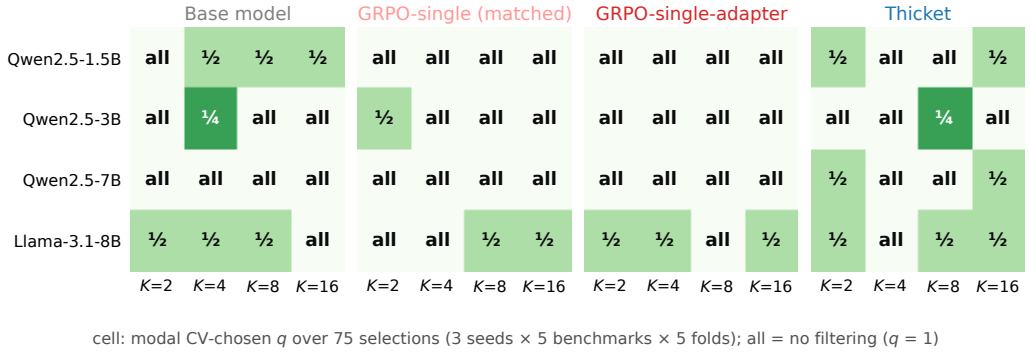


Figure 7: Most frequently selected CV filter strength per (model, K) setting and method (mode over 75 selections: three seeds $\times$ five math benchmarks $\times$ five folds; ties toward the weaker filter, consistent with the selection rule; darker = more aggressive filtering). The most frequent choice is no filtering in 13 of 16 settings for GRPO-single-adapter and for GRPO-single (matched), in 9 for the base model, and in 8 for the thicket. Because real-filter selections split between $1/2$ and $1/4$, the mode understates how often each pool filters. Across all 1,200 selections per method, the CV picks a real filter in 67% (base), 64% (thicket), 59% (GRPO-single (matched)), and 53% (GRPO-single-adapter) of cases.

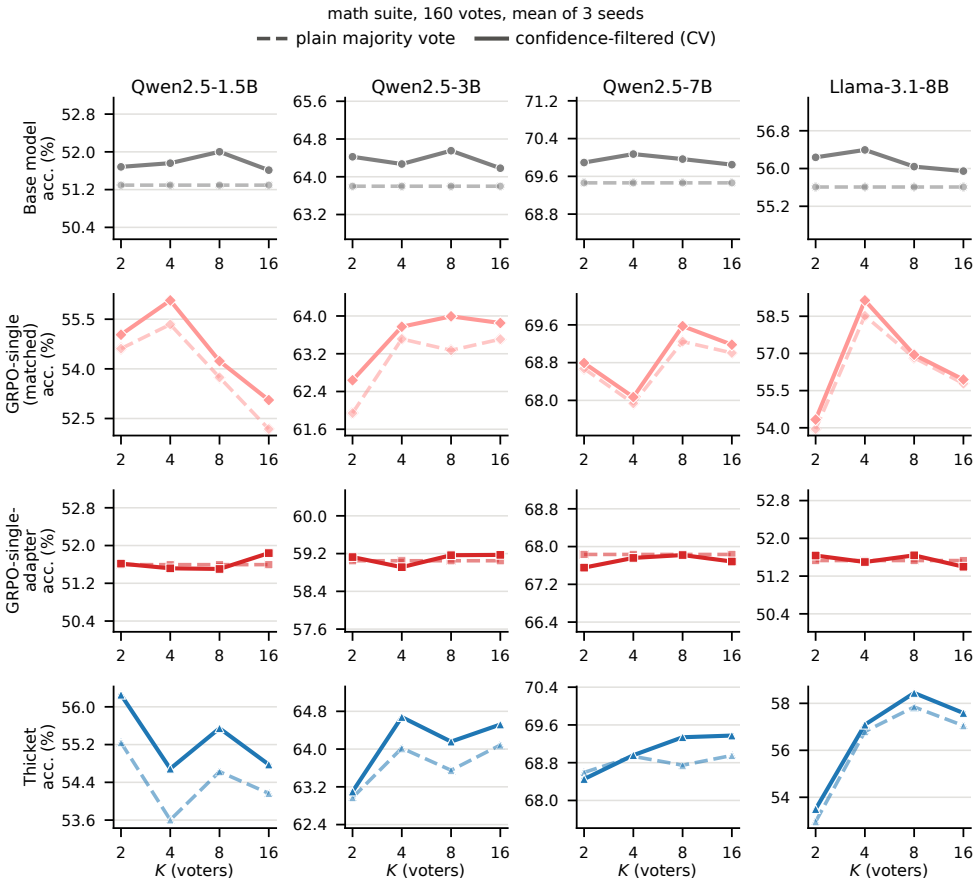


Figure 8: Plain majority vote (dashed) vs. confidence-filtered voting (solid) on the math-suite macro at 160 votes, per method (rows) and model (columns); mean of three seeds. By displayed means, the solid curve sits above the dashed one in all 16 settings for the base model and GRPO-single (matched) and in 15 of 16 for the thicket. For GRPO-single-adapter it is above in 6 settings, level in 4, and below in 6. No bands are drawn, because bands of the two curves would hide a gain of about half a point. Tables 9–12 give the stds.

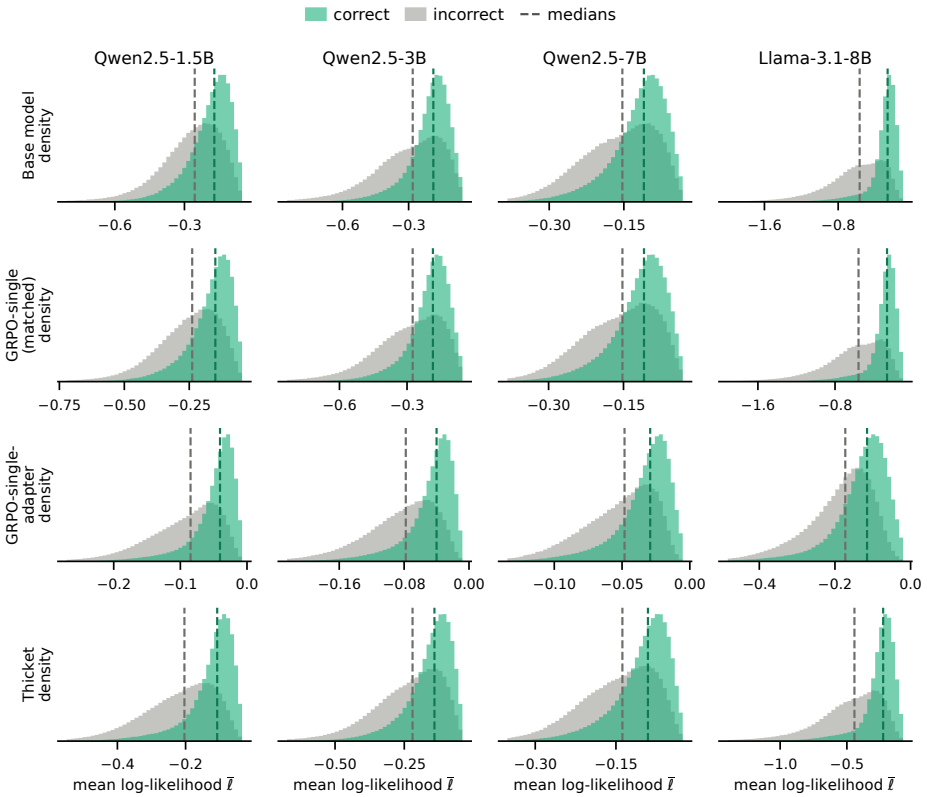


Figure 9: **Why confidence is worth acting on.** Distributions of the per-response mean log-likelihood $\bar{\ell}$ for correct (green) vs. incorrect (gray) completions at $K{=}16$, over all 160 completions per problem, pooled over the three seeds and the five math benchmarks (non-null responses; histogram ranges clipped to the 0.5–99.5th percentiles; dashed lines: medians, dark green for correct and gray for incorrect). x-axes differ across panels. Correct completions have the higher median in every method–model pool – the signal confidence-filtered voting exploits. The fully trained GRPO-single-adapter (third row) concentrates its completions much closer to $\bar{\ell}{=}0$ than the other pools.

D PER-BENCHMARK COVERAGE

Tables 13–16 break the suite-mean coverage of Table 2 down by benchmark. Figures 10 and 11 plot the same numbers as scaling curves with ± 1 std bands.

Table 13: Per-benchmark coverage (pass@pool %) on the math suite, $K \in \{2, 4\}$; mean $\pm$ std over three seeds. Bold: best method per row and K; underlined: second best.

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark (K=)	2	4	2	4	2	4	2	4
Qwen2.5-1.5B	MATH500	93.5 ±0.8	93.5 ±0.8	**94.0 ±0.7**	**93.9 ±0.3**	92.7 ±1.1	92.7 ±1.1	92.8 ±0.3	93.7 ±0.2
	GSM8K	**99.1 ±0.1**	**99.1 ±0.1**	98.6 ±0.2	99.0 ±0.2	98.1 ±0.3	98.1 ±0.3	98.9 ±0.3	98.9 ±0.1
	OlympiadBench	67.9 ±0.4	67.9 ±0.4	**69.1 ±0.3**	68.4 ±0.1	66.3 ±0.9	66.3 ±0.9	68.2 ±1.0	**69.3 ±0.9**
	Countdown	**87.8 ±1.0**	**87.8 ±1.0**	86.2 ±2.7	85.1 ±2.1	81.6 ±4.5	81.6 ±4.5	84.2 ±4.3	85.3 ±1.8
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	**100.0 ±0.0**
	Mean	**89.6 ±0.3**	**89.6 ±0.3**	**89.6 ±0.4**	89.3 ±0.5	87.7 ±0.7	87.7 ±0.7	88.8 ±1.0	89.5 ±0.4
Qwen2.5-3B	MATH500	**95.3 ±0.7**	**95.3 ±0.7**	94.3 ±0.2	94.5 ±0.4	93.5 ±0.8	93.5 ±0.8	94.5 ±0.6	94.9 ±0.5
	GSM8K	**99.3 ±0.0**	**99.3 ±0.0**	98.9 ±0.2	99.0 ±0.0	98.5 ±0.2	98.5 ±0.2	98.8 ±0.3	99.1 ±0.1
	OlympiadBench	**75.1 ±0.6**	**75.1 ±0.6**	74.4 ±0.2	74.4 ±1.1	72.2 ±1.7	72.2 ±1.7	74.3 ±0.8	74.6 ±1.1
	Countdown	**89.1 ±1.0**	**89.1 ±1.0**	86.7 ±1.8	87.1 ±2.5	80.9 ±2.7	80.9 ±2.7	86.4 ±1.7	85.1 ±3.0
	Knights & Knaves	99.8 ±0.2	99.8 ±0.2	99.8 ±0.2	**100.0 ±0.0**	99.7 ±0.6	99.7 ±0.6	**100.0 ±0.0**	**100.0 ±0.0**
	Mean	**91.7 ±0.1**	**91.7 ±0.1**	90.8 ±0.3	91.0 ±0.5	88.9 ±1.0	88.9 ±1.0	90.8 ±0.6	90.8 ±0.8
Qwen2.5-7B	MATH500	**96.3 ±0.1**	**96.3 ±0.1**	95.9 ±0.3	95.7 ±0.1	95.5 ±0.5	95.5 ±0.5	95.9 ±0.3	96.0 ±0.2
	GSM8K	**98.7 ±0.0**	**98.7 ±0.0**	98.6 ±0.0	98.6 ±0.1	98.2 ±0.1	98.2 ±0.1	98.4 ±0.0	**98.7 ±0.1**
	OlympiadBench	**78.8 ±0.2**	78.8 ±0.2	78.7 ±0.1	78.0 ±0.6	76.7 ±0.7	76.7 ±0.7	78.4 ±0.7	**78.9 ±0.8**
	Countdown	**90.0 ±1.8**	**90.0 ±1.8**	88.4 ±1.0	86.2 ±2.5	86.0 ±2.0	86.0 ±2.0	87.3 ±1.3	88.2 ±1.0
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	99.9 ±0.2	**100.0 ±0.0**	**100.0 ±0.0**
	Mean	**92.8 ±0.4**	**92.8 ±0.4**	92.3 ±0.2	91.7 ±0.6	91.2 ±0.5	91.2 ±0.5	92.0 ±0.2	92.4 ±0.4
Llama-3.1-8B	MATH500	91.3 ±0.5	91.3 ±0.5	90.9 ±0.1	**92.1 ±0.6**	88.9 ±0.3	88.9 ±0.3	**92.0 ±0.0**	91.5 ±0.6
	GSM8K	**99.2 ±0.1**	**99.2 ±0.1**	98.4 ±0.3	99.0 ±0.2	98.0 ±0.5	98.0 ±0.5	98.9 ±0.3	99.1 ±0.1
	OlympiadBench	63.6 ±0.5	63.6 ±0.5	65.5 ±0.7	**68.2 ±0.4**	63.5 ±2.0	63.5 ±2.0	**67.4 ±1.3**	67.8 ±1.0
	Countdown	77.6 ±1.5	77.6 ±1.5	**83.3 ±1.3**	**87.8 ±2.0**	81.6 ±6.8	81.6 ±6.8	82.9 ±1.5	87.6 ±2.0
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	99.6 ±0.2	99.8 ±0.2	99.4 ±1.0	99.4 ±1.0	99.7 ±0.0	99.7 ±0.0
	Mean	86.3 ±0.3	86.3 ±0.3	87.5 ±0.3	**89.4 ±0.5**	86.3 ±2.0	86.3 ±2.0	**88.2 ±0.2**	89.1 ±0.5

Table 14: Per-benchmark coverage (pass@pool %) on the math suite, $K \in \{8, 16\}$ (format of Table 13).

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark (K=)	8	16	8	16	8	16	8	16
Qwen2.5-1.5B	MATH500	93.5 ±0.8	**93.5 ±0.8**	**94.1 ±0.4**	92.5 ±0.4	92.7 ±1.1	92.7 ±1.1	93.9 ±0.5	93.4 ±0.2
	GSM8K	**99.1 ±0.1**	**99.1 ±0.1**	**99.1 ±0.2**	**99.1 ±0.1**	98.1 ±0.3	98.1 ±0.3	98.8 ±0.2	98.8 ±0.2
	OlympiadBench	67.9 ±0.4	67.9 ±0.4	**68.9 ±0.1**	68.6 ±0.6	66.3 ±0.9	66.3 ±0.9	68.7 ±0.1	**69.7 ±0.8**
	Countdown	87.8 ±1.0	**87.8 ±1.0**	**88.0 ±1.3**	87.6 ±2.7	81.6 ±4.5	81.6 ±4.5	86.4 ±2.0	87.1 ±1.9
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	**100.0 ±0.0**
	Mean	89.6 ±0.3	89.6 ±0.3	**90.0 ±0.3**	89.5 ±0.5	87.7 ±0.7	87.7 ±0.7	89.6 ±0.5	**89.8 ±0.6**
Qwen2.5-3B	MATH500	95.3 ±0.7	**95.3 ±0.7**	95.0 ±0.4	94.9 ±0.1	93.5 ±0.8	93.5 ±0.8	**95.5 ±0.5**	94.9 ±0.1
	GSM8K	**99.3 ±0.0**	**99.3 ±0.0**	**99.3 ±0.1**	**99.3 ±0.2**	98.5 ±0.2	98.5 ±0.2	99.2 ±0.2	**99.3 ±0.2**
	OlympiadBench	75.1 ±0.6	75.1 ±0.6	**75.2 ±1.2**	74.5 ±0.7	72.2 ±1.7	72.2 ±1.7	74.5 ±1.0	**75.8 ±1.2**
	Countdown	**89.1 ±1.0**	89.1 ±1.0	88.4 ±2.0	**89.3 ±1.8**	80.9 ±2.7	80.9 ±2.7	87.3 ±1.2	88.2 ±1.0
	Knights & Knaves	99.8 ±0.2	99.8 ±0.2	**100.0 ±0.0**	**100.0 ±0.0**	99.7 ±0.6	99.7 ±0.6	**100.0 ±0.0**	**100.0 ±0.0**
	Mean	**91.7 ±0.1**	**91.7 ±0.1**	91.6 ±0.6	91.6 ±0.5	88.9 ±1.0	88.9 ±1.0	91.3 ±0.4	91.6 ±0.1
Qwen2.5-7B	MATH500	**96.3 ±0.1**	**96.3 ±0.1**	95.7 ±0.2	95.8 ±0.2	95.5 ±0.5	95.5 ±0.5	**96.3 ±0.6**	**96.3 ±0.3**
	GSM8K	98.7 ±0.0	**98.7 ±0.0**	**98.8 ±0.2**	**98.7 ±0.1**	98.2 ±0.1	98.2 ±0.1	98.6 ±0.1	98.6 ±0.1
	OlympiadBench	78.8 ±0.2	78.8 ±0.2	**79.1 ±0.3**	**78.9 ±0.6**	76.7 ±0.7	76.7 ±0.7	78.5 ±0.5	**78.9 ±0.5**
	Countdown	**90.0 ±1.8**	**90.0 ±1.8**	89.3 ±0.7	88.2 ±3.2	86.0 ±2.0	86.0 ±2.0	88.2 ±0.8	89.3 ±0.7
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	99.9 ±0.2	**100.0 ±0.0**	**100.0 ±0.0**
	Mean	**92.8 ±0.4**	**92.8 ±0.4**	92.6 ±0.2	92.3 ±0.8	91.2 ±0.5	91.2 ±0.5	92.3 ±0.2	92.6 ±0.2
Llama-3.1-8B	MATH500	91.3 ±0.5	91.3 ±0.5	92.1 ±0.8	91.7 ±0.7	88.9 ±0.3	88.9 ±0.3	**92.4 ±0.3**	**92.2 ±0.6**
	GSM8K	99.2 ±0.1	**99.2 ±0.1**	99.2 ±0.1	**99.2 ±0.1**	98.0 ±0.5	98.0 ±0.5	**99.3 ±0.2**	99.1 ±0.2
	OlympiadBench	63.6 ±0.5	63.6 ±0.5	66.7 ±1.3	64.8 ±1.7	63.5 ±2.0	63.5 ±2.0	**68.6 ±1.1**	**68.0 ±1.0**
	Countdown	77.6 ±1.5	77.6 ±1.5	81.3 ±2.4	78.4 ±1.7	81.6 ±6.8	81.6 ±6.8	**89.3 ±1.8**	**83.6 ±1.4**
	Knights & Knaves	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**	99.9 ±0.2	99.4 ±1.0	99.4 ±1.0	99.8 ±0.4	99.8 ±0.4
	Mean	86.3 ±0.3	86.3 ±0.3	87.9 ±0.1	86.8 ±0.6	86.3 ±2.0	86.3 ±2.0	**89.9 ±0.7**	**88.5 ±0.2**

Table 15: Per-benchmark coverage (pass@pool %) on the code suite, $K \in \{2, 4\}$; mean $\pm$ std over three seeds. Bold: best method per row and K; underlined: second best.

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark ($K=$)	2	4	2	4	2	4	2	4
Qwen2.5-1.5B	MBPP	**93.4 ±0.0**	**93.4 ±0.0**	90.6 ±2.4	92.4 ±1.0	88.1 ±1.6	88.1 ±1.6	91.4 ±0.9	92.7 ±0.3
	HumanEval	**96.7 ±0.9**	**96.7 ±0.9**	92.5 ±3.0	93.9 ±0.6	88.8 ±1.4	88.8 ±1.4	92.9 ±1.4	93.9 ±0.6
	LeetCode	**26.5 ±0.3**	**26.5 ±0.3**	24.0 ±3.7	24.4 ±2.1	20.5 ±2.6	20.5 ±2.6	23.4 ±2.8	25.1 ±1.1
	Mean	**72.2 ±0.4**	**72.2 ±0.4**	69.0 ±1.8	70.2 ±0.9	65.8 ±1.4	65.8 ±1.4	69.2 ±1.4	70.6 ±0.6
Qwen2.5-3B	MBPP	**95.2 ±0.5**	**95.2 ±0.5**	93.0 ±2.8	92.9 ±1.8	90.8 ±1.1	90.8 ±1.1	92.6 ±1.3	93.1 ±1.8
	HumanEval	**97.6 ±0.6**	**97.6 ±0.6**	96.5 ±1.5	96.1 ±1.5	95.3 ±0.9	95.3 ±0.9	97.4 ±0.9	96.7 ±0.7
	LeetCode	**34.1 ±2.4**	**34.1 ±2.4**	31.7 ±2.1	30.4 ±2.1	26.0 ±1.5	26.0 ±1.5	30.4 ±2.1	32.2 ±1.8
	Mean	**75.6 ±1.1**	**75.6 ±1.1**	73.7 ±2.0	73.1 ±1.7	70.7 ±0.5	70.7 ±0.5	73.5 ±1.4	74.0 ±1.4
Qwen2.5-7B	MBPP	**93.1 ±0.1**	**93.1 ±0.1**	92.0 ±0.9	92.1 ±1.2	90.1 ±0.9	90.1 ±0.9	93.0 ±0.5	92.6 ±0.5
	HumanEval	**96.7 ±0.4**	**96.7 ±0.4**	94.7 ±0.9	96.5 ±0.4	94.9 ±0.4	94.9 ±0.4	96.3 ±0.6	96.1 ±0.9
	LeetCode	**39.0 ±0.9**	**39.0 ±0.9**	36.3 ±0.7	38.3 ±1.1	34.2 ±0.8	34.2 ±0.8	37.9 ±0.3	38.9 ±0.9
	Mean	**76.3 ±0.2**	**76.3 ±0.2**	74.3 ±0.5	75.7 ±0.7	73.1 ±0.4	73.1 ±0.4	75.7 ±0.2	75.9 ±0.2
Llama-3.1-8B	MBPP	**93.4 ±0.5**	**93.4 ±0.5**	89.2 ±0.5	90.9 ±1.0	88.4 ±1.6	88.4 ±1.6	90.7 ±0.8	90.6 ±0.8
	HumanEval	**93.1 ±0.7**	**93.1 ±0.7**	90.7 ±2.7	91.7 ±1.5	90.4 ±0.9	90.4 ±0.9	90.0 ±2.0	91.9 ±0.4
	LeetCode	**39.0 ±0.4**	39.0 ±0.4	38.0 ±2.1	38.7 ±1.5	36.7 ±3.2	36.7 ±3.2	**39.0 ±1.9**	**40.6 ±0.5**
	Mean	**75.2 ±0.2**	**75.2 ±0.2**	72.6 ±1.7	73.8 ±1.1	71.8 ±1.9	71.8 ±1.9	73.3 ±1.4	74.4 ±0.5

Table 16: Per-benchmark coverage (pass@pool %) on the code suite, $K \in \{8, 16\}$ (format of Table 15).

		Base		GRPO-single (matched)		GRPO-single-adapter		Thicket	
Model	Benchmark ($K=$)	8	16	8	16	8	16	8	16
Qwen2.5-1.5B	MBPP	93.4 ±0.0	93.4 ±0.0	**93.8 ±0.1**	**93.7 ±0.8**	88.1 ±1.6	88.1 ±1.6	93.3 ±0.1	93.0 ±0.5
	HumanEval	**96.7 ±0.9**	**96.7 ±0.9**	**96.7 ±0.7**	96.1 ±1.5	88.8 ±1.4	88.8 ±1.4	94.1 ±1.4	93.5 ±0.9
	LeetCode	**26.5 ±0.3**	**26.5 ±0.3**	25.6 ±0.9	25.9 ±0.4	20.5 ±2.6	20.5 ±2.6	24.3 ±0.3	24.6 ±1.5
	Mean	**72.2 ±0.4**	**72.2 ±0.4**	72.0 ±0.5	71.9 ±0.3	65.8 ±1.4	65.8 ±1.4	70.6 ±0.5	70.3 ±0.3
Qwen2.5-3B	MBPP	**95.2 ±0.5**	95.2 ±0.5	94.8 ±0.5	**95.6 ±0.2**	90.8 ±1.1	90.8 ±1.1	94.2 ±0.9	95.4 ±0.7
	HumanEval	97.6 ±0.6	97.6 ±0.6	97.8 ±0.7	**98.2 ±0.6**	95.3 ±0.9	95.3 ±0.9	**98.0 ±0.4**	97.8 ±0.4
	LeetCode	**34.1 ±2.4**	34.1 ±2.4	33.8 ±1.8	**35.4 ±2.0**	26.0 ±1.5	26.0 ±1.5	33.5 ±0.7	34.6 ±0.4
	Mean	**75.6 ±1.1**	75.6 ±1.1	75.4 ±0.8	**76.4 ±0.9**	70.7 ±0.5	70.7 ±0.5	75.2 ±0.4	75.9 ±0.3
Qwen2.5-7B	MBPP	93.1 ±0.1	93.1 ±0.1	92.9 ±0.4	**93.2 ±0.5**	90.1 ±0.9	90.1 ±0.9	**93.4 ±0.4**	92.7 ±0.1
	HumanEval	**96.7 ±0.4**	96.7 ±0.4	96.1 ±0.4	96.7 ±0.4	94.9 ±0.4	94.9 ±0.4	96.5 ±0.9	**97.0 ±0.0**
	LeetCode	39.0 ±0.9	39.0 ±0.9	**39.8 ±1.8**	39.3 ±1.0	34.2 ±0.8	34.2 ±0.8	38.9 ±0.5	**39.5 ±1.2**
	Mean	**76.3 ±0.2**	76.3 ±0.2	**76.3 ±0.4**	**76.4 ±0.2**	73.1 ±0.4	73.1 ±0.4	**76.3 ±0.4**	**76.4 ±0.4**
Llama-3.1-8B	MBPP	**93.4 ±0.5**	93.4 ±0.5	92.8 ±0.5	**93.6 ±0.3**	88.4 ±1.6	88.4 ±1.6	92.3 ±0.2	92.1 ±0.7
	HumanEval	**93.1 ±0.7**	93.1 ±0.7	92.9 ±0.4	**93.3 ±0.0**	90.4 ±0.9	90.4 ±0.9	92.3 ±0.9	92.3 ±0.4
	LeetCode	39.0 ±0.4	39.0 ±0.4	**39.2 ±1.0**	39.3 ±0.7	36.7 ±3.2	36.7 ±3.2	38.5 ±1.5	**40.1 ±0.7**
	Mean	**75.2 ±0.2**	75.2 ±0.2	75.0 ±0.4	**75.4 ±0.1**	71.8 ±1.9	71.8 ±1.9	74.3 ±0.1	74.8 ±0.5

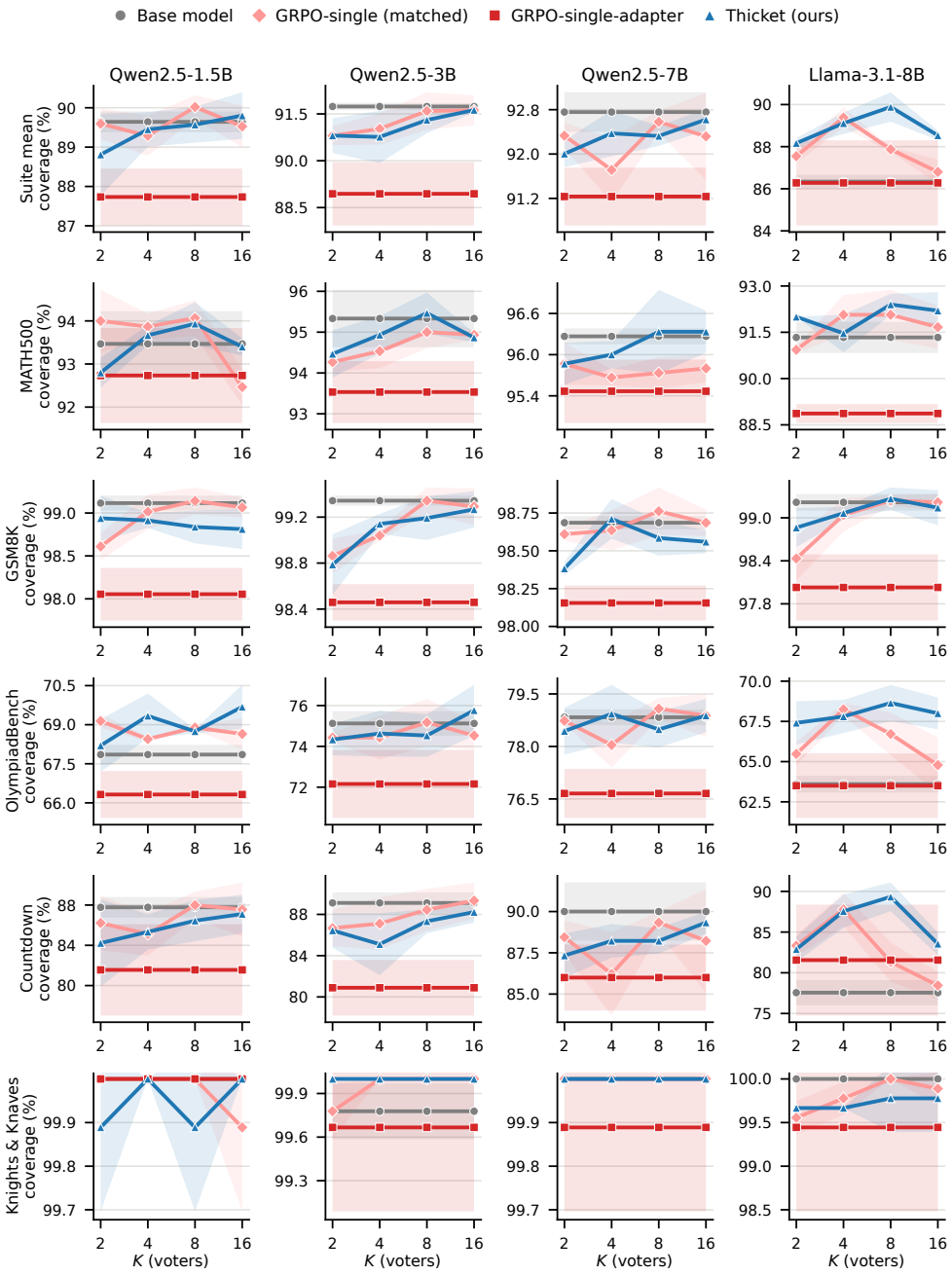


Figure 10: Coverage vs. voter count K on the math suite (fixed 160-completion pools). Lines: seed means; bands: ± 1 std. Base and GRPO-single-adapter are flat by construction. In the Knights & Knaves row, coverage saturates at 100% and coinciding lines hide one another.

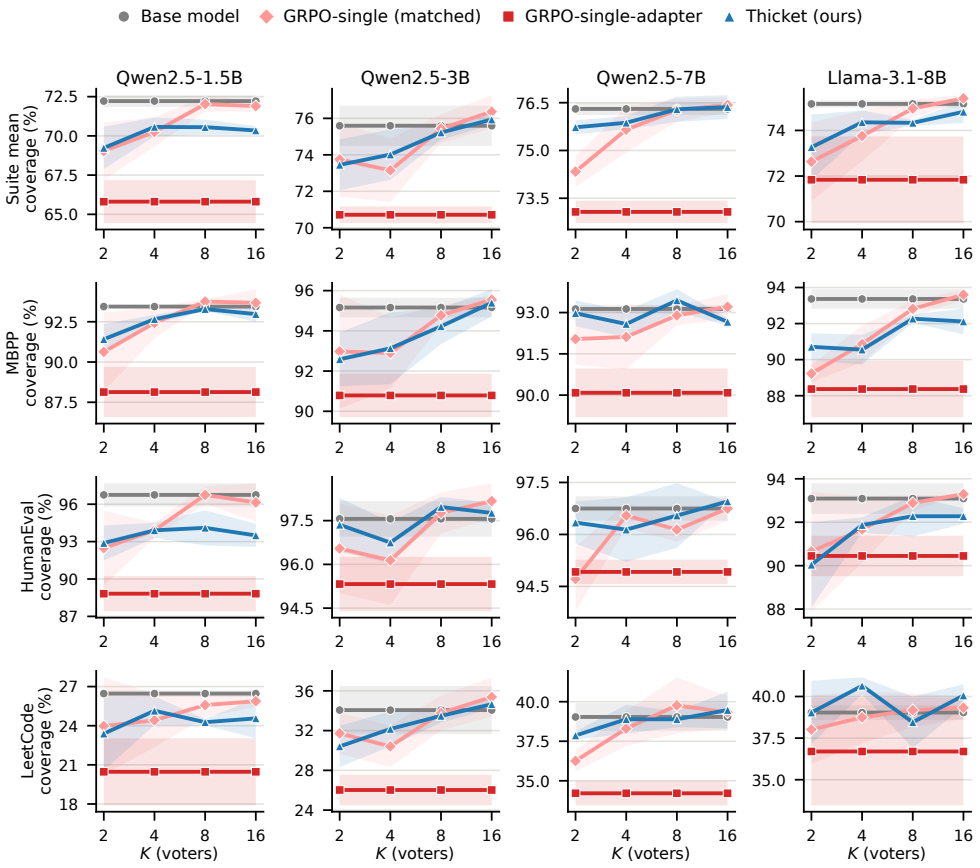


Figure 11: Coverage vs. voter count K on the code suite (fixed 160-completion pools). Lines: seed means; bands: ± 1 std. Base and GRPO-single-adapter are flat by construction.

E VOTER ACCURACY, ERROR CORRELATION, AND A MARGIN DECOMPOSITION

Figure 12 places every voter at $K{=}16$ in the accuracy–correlation plane, and Table 18 summarizes it; Figures 13 and 14 give the complete plot set for $K \in \{2, 4, 8, 16\}$ on both suites, with the voters of all three seeds pooled in each panel. Per-benchmark $\bar{\rho}$ is computed over each voter's $160/K$ stored completions; suite-level values are per-voter macro means over the suite's benchmarks.

Where does the thicket's margin come from? Table 17 splits the thicket's $K{=}16$ margin over GRPO-single-adapter at 160 votes into the vote gain it keeps over the base and the damage it avoids. On Qwen2.5-3B and Llama, most of the margin is avoided damage ($+4.8$ and $+4.1$ p.p.). On Qwen2.5-7B the avoided damage ($+1.7$) exceeds the whole margin ($+1.2$), because the thicket itself trails the base by 0.5. On Qwen2.5-1.5B, where the fully trained adapter's vote merely matches the base, the margin ($+2.6$) comes entirely from retained gain ($+2.9$, offset by -0.3). The retained gain in turn tracks how much RLVR can add at all. Where full training lifts pass@1 by $+7.3$ and $+9.7$ p.p. (Qwen2.5-1.5B and Llama), the thicket beats the base ($+2.9$ and $+1.4$). Where it lifts pass@1 by only $+1.0$ and $+0.7$ (Qwen2.5-3B and 7B), the thicket roughly matches the base ($+0.3$ and -0.5). On these four models, a thicket converts RLVR's accuracy gain into a vote gain when there is one, and otherwise shields the base model's vote from most of the damage that concentrated training inflicts.

Table 17: **Decomposing the thicket's vote margin over GRPO-single-adapter** at $K{=}16$ and 160 votes (math macro, mean over three seeds). $G = (\text{thicket} - \text{base}) + (\text{base} - \text{GRPO-single-adapter})$: the RLVR gain the thicket retains, plus the damage that concentrated training inflicts and the thicket avoids. H and $\Delta\bar{\rho}$ are the pass@1 and error-correlation changes that full single-adapter training causes relative to the base.

	What full RLVR does		Thicket margin (MV, p.p.)		
Model	H (pass@1)	$\Delta\bar{\rho}$	thicket−base	base−GRPO-single-adapter	G
Qwen2.5-1.5B	+7.3	+0.13	+2.9	−0.3	+2.6
Qwen2.5-3B	+1.0	+0.07	+0.3	+4.8	+5.1
Qwen2.5-7B	+0.7	+0.05	−0.5	+1.7	+1.2
Llama-3.1-8B	+9.7	+0.19	+1.4	+4.1	+5.5

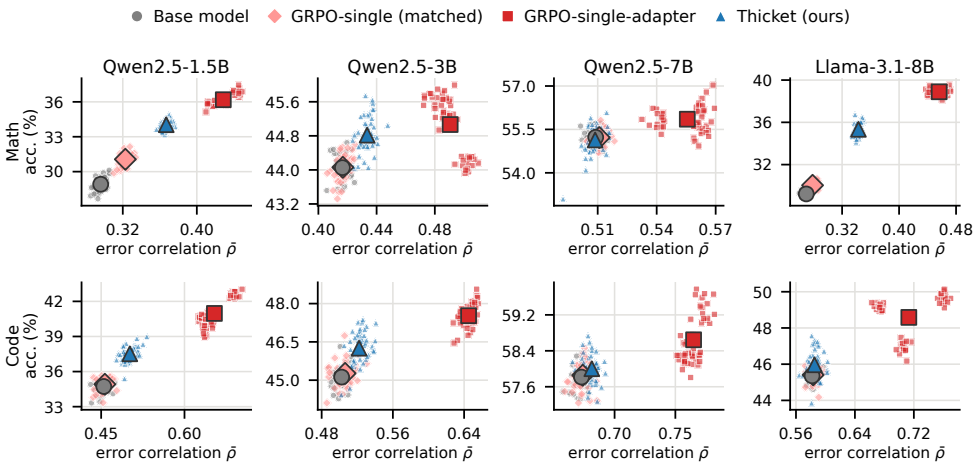


Figure 12: **Voter accuracy vs. error correlation at** $K{=}16$ (per-voter macro means over each suite's benchmarks, voters of all three seeds pooled; large markers are method means). The fully trained GRPO-single-adapter moves up and far to the right of the base model. The step-matched control stays close to the base model in both correlation and accuracy. Thicket members add less correlation than GRPO-single-adapter in every panel, and gain accuracy over the base model in all but Qwen2.5-7B math (-0.1).

Table 18: Mean voter accuracy (single completion, %) and mean pairwise error correlation $\bar{\rho}$ at $K=16$; macro means over each suite's benchmarks, mean $\pm$ std over three seeds. Matched: GRPO-single (matched); Single: GRPO-single-adapter.

		Voter accuracy (%)				Error correlation $\bar{\rho}$			
Suite	Model	Base	Matched	Single	Thicket	Base	Matched	Single	Thicket
Math	Qwen2.5-1.5B	28.9 $\pm$0.1	31.1 $\pm$0.1	36.2 $\pm$0.5	34.0 $\pm$0.1	0.30 $\pm$0.00	0.32 $\pm$0.00	0.43 $\pm$0.01	0.37 $\pm$0.00
	Qwen2.5-3B	44.1 $\pm$0.1	44.1 $\pm$0.2	45.1 $\pm$0.8	44.8 $\pm$0.1	0.42 $\pm$0.00	0.42 $\pm$0.00	0.49 $\pm$0.01	0.43 $\pm$0.00
	Qwen2.5-7B	55.2 $\pm$0.1	55.2 $\pm$0.0	55.9 $\pm$0.4	55.1 $\pm$0.0	0.51 $\pm$0.00	0.51 $\pm$0.00	0.56 $\pm$0.01	0.51 $\pm$0.00
	Llama-3.1-8B	29.2 $\pm$0.0	30.0 $\pm$0.2	38.9 $\pm$0.2	35.3 $\pm$0.0	0.27 $\pm$0.00	0.28 $\pm$0.00	0.46 $\pm$0.01	0.34 $\pm$0.00
Code	Qwen2.5-1.5B	34.7 $\pm$0.1	34.9 $\pm$0.2	41.0 $\pm$1.5	37.5 $\pm$0.2	0.45 $\pm$0.00	0.46 $\pm$0.00	0.65 $\pm$0.03	0.50 $\pm$0.01
	Qwen2.5-3B	45.1 $\pm$0.1	45.3 $\pm$0.1	47.5 $\pm$0.4	46.3 $\pm$0.1	0.50 $\pm$0.00	0.51 $\pm$0.01	0.65 $\pm$0.01	0.52 $\pm$0.00
	Qwen2.5-7B	57.8 $\pm$0.1	57.9 $\pm$0.1	58.6 $\pm$0.6	58.0 $\pm$0.1	0.67 $\pm$0.00	0.67 $\pm$0.00	0.77 $\pm$0.01	0.68 $\pm$0.00
	Llama-3.1-8B	45.4 $\pm$0.1	45.4 $\pm$0.2	48.6 $\pm$1.4	46.0 $\pm$0.2	0.58 $\pm$0.00	0.58 $\pm$0.00	0.71 $\pm$0.04	0.59 $\pm$0.01

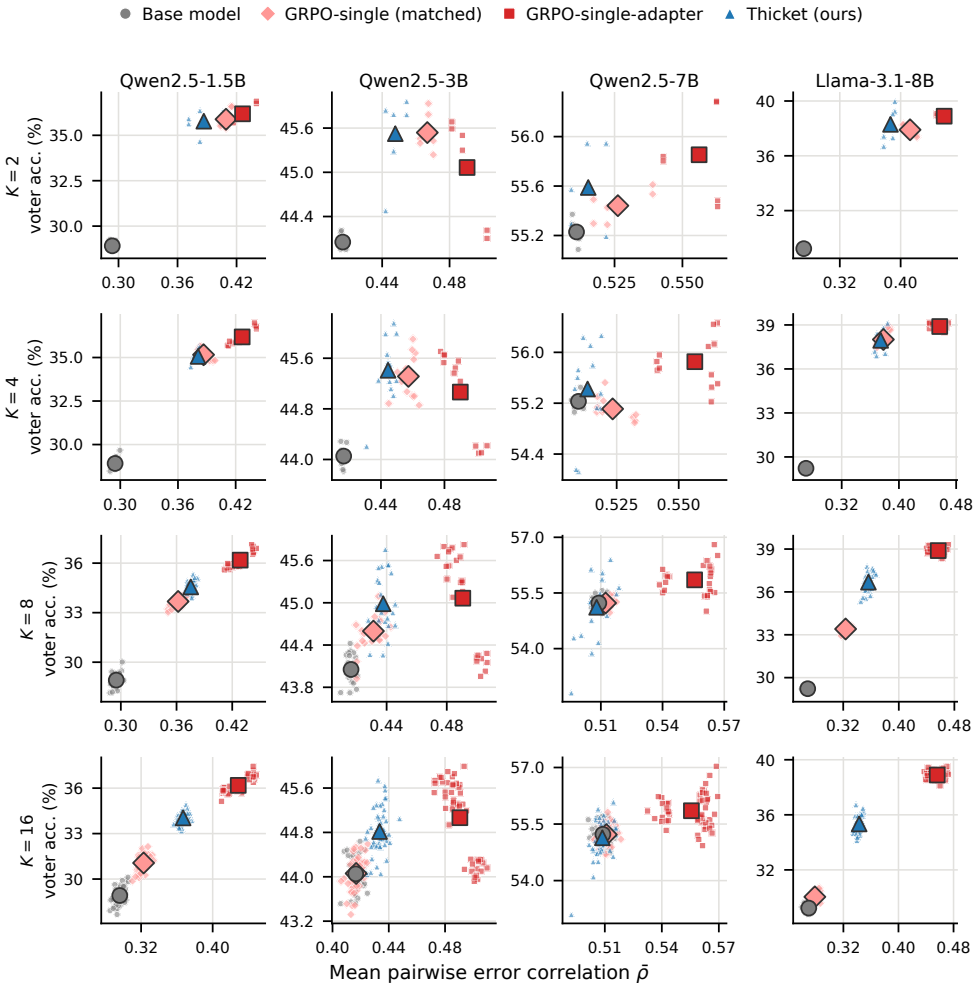


Figure 13: Complete plot set, math suite: per-voter accuracy vs. $\bar{\rho}$ for $K \in \{2, 4, 8, 16\}$ (rows) and all models (columns); voters of all three seeds pooled; large markers are method means.

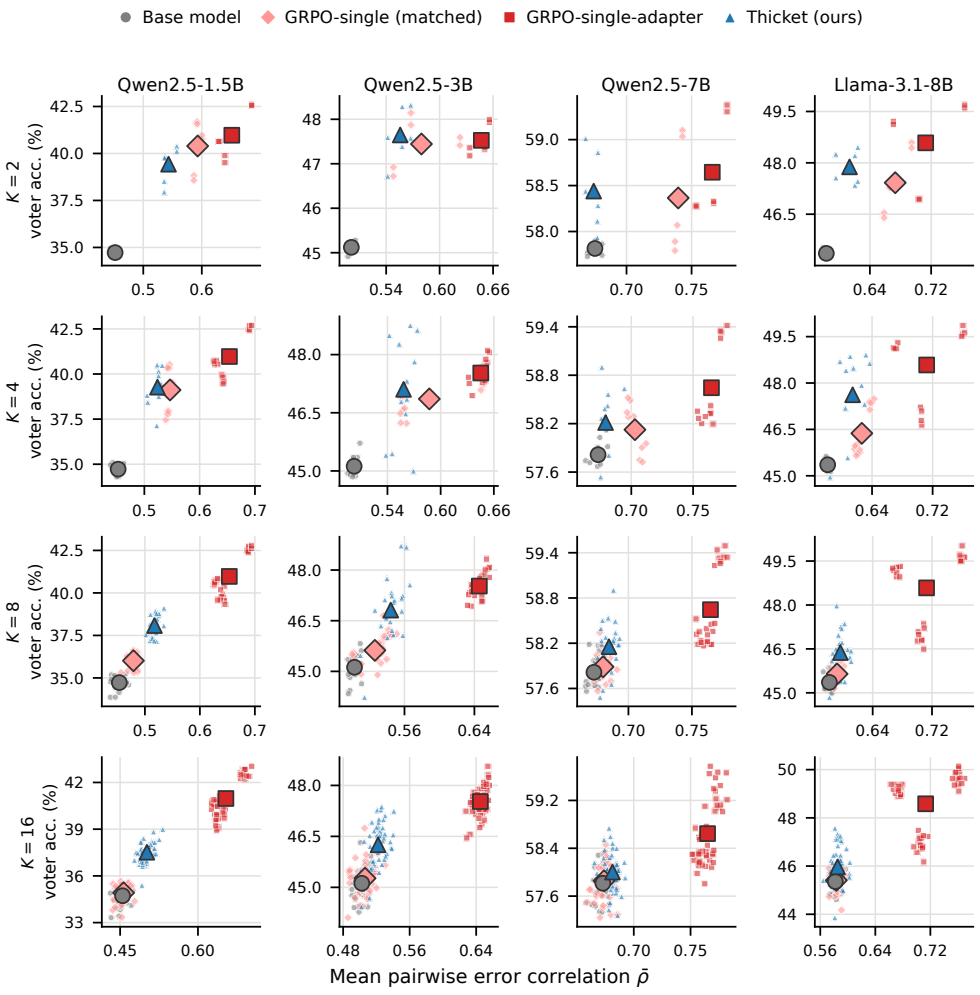


Figure 14: Complete plot set, code suite (protocol as in Figure 13).

F THE VOTE BUDGET: FROM 16 TO 160 VOTES

Section 4.5 summarizes these results by mean margins. Here every method votes over its first B completions, $B \in \{16, 32, \ldots, 160\}$, under the first-$B$ convention of Appendix B. Table 19 gives the mean margins of Figure 4 at seven budgets, together with each method's mean vote. Figure 15 shows each method's curves by voter count K, and Figure 16 overlays the four methods in each (model, K) panel.

Three patterns stand out. First, the base-model and GRPO-single-adapter curves are identical at every K, because their first B completions do not depend on K. Second, the fully trained GRPO-single-adapter saturates fastest. Its vote rises by 3.78 p.p. from 16 to 160 votes, against 6.50 for the base model (means of the per-setting gains). Along the single adapter's run the gain shrinks as training continues past step 50: it is $+6.55$ and $+6.58$ p.p. at steps 25 and 50 (GRPO-single (matched) at K=16 and 8), $+6.18$ at step 125, $+4.78$ at step 250, and $+3.78$ at the final checkpoint. Third, larger thickets gain more from the same extra votes: $+4.68$ p.p. at K=2, $+5.43$ at K=4, $+6.35$ at K=8, and $+6.48$ at K=16 (means over models of the per-model gains).

Table 19: **The vote budget** (math majority vote over each method's first B completions; three-seed means). Upper rows: mean margin (p.p.) over the 16 (model, K) settings, each the difference of displayed means, with the number of settings where it is positive in parentheses. Lower rows: each method's majority vote, averaged over the 16 settings, $\pm$ the std over seeds of that average. B=160 is Table 1, and B=16 is Table 20. Figure 4 plots all ten budgets. Base and GRPO-single-adapter do not depend on K, so the count for GRPO-single-adapter − Base moves in steps of four (one model at a time).

	Votes per problem B (first B responses of each method)						
	16	32	48	64	96	128	160
Mean margin (p.p.) over the 16 (model, K) settings (settings with a positive margin)							
GRPO-single-adapter − Base	+0.15 (8)	−0.40 (8)	−1.08 (4)	−1.43 (4)	−1.93 (0)	−2.28 (4)	−2.58 (4)
Thicket − GRPO-single-adapter	+1.33 (15)	+1.68 (16)	+2.01 (16)	+2.25 (16)	+2.86 (16)	+3.26 (16)	+3.28 (16)
Thicket − Base	+1.48 (13)	+1.28 (13)	+0.93 (10)	+0.83 (8)	+0.94 (11)	+0.98 (10)	+0.71 (9)
Thicket − GRPO-single (matched)	+0.56 (11)	+0.47 (11)	+0.28 (10)	+0.28 (9)	+0.23 (9)	+0.34 (11)	+0.27 (10)
Majority vote (%), mean over the 16 settings							
Base	53.6 ±0.4	56.1 ±0.3	57.6 ±0.4	58.3 ±0.3	59.1 ±0.2	59.6 ±0.0	60.1 ±0.2
GRPO-single (matched)	54.5 ±0.5	56.9 ±0.5	58.2 ±0.6	58.8 ±0.4	59.8 ±0.4	60.2 ±0.4	60.5 ±0.4
GRPO-single-adapter	53.7 ±1.3	55.7 ±1.5	56.5 ±1.7	56.9 ±1.6	57.2 ±1.8	57.3 ±2.0	57.5 ±2.2
Thicket	55.0 ±0.4	57.4 ±0.2	58.5 ±0.1	59.1 ±0.2	60.0 ±0.2	60.5 ±0.2	60.8 ±0.3

A 16-vote replication. Tables 20 and 21 repeat Tables 1 and 2 at B=16, the smallest budget that splits evenly across every $K \in \{2, 4, 8, 16\}$. A thicket member contributes its first $16/K$ completions ($8/4/2/1$ for K=$2/4/8/16$), and every single policy the first 16 completions of its pool, those drawn with its first 16 sampling seeds. At 16 votes the thicket beats GRPO-single-adapter in 15 of 16 settings (mean margin $+1.33$ p.p.), the base in 13 (tying 2; $+1.48$), and GRPO-single (matched) in 11 (tying 1; $+0.56$). The fully trained GRPO-single-adapter votes above the base on Qwen2.5-1.5B ($+1.4$) and Llama ($+1.8$), and below it on Qwen2.5-3B (-1.5) and Qwen2.5-7B (-1.1). For GRPO-single-adapter, coverage tells the same story as at 160 votes. Thickets cover more than GRPO-single-adapter in all 32 (model, suite, K) settings (mean margins $+2.02$ p.p. on math and $+3.03$ on code). GRPO-single-adapter covers less than the base in seven of eight model–suite settings, and more only on Llama math ($+3.2$), where it ties the base at 160 votes. Thickets cover more than the base in 9 of 16 settings on each suite. That is more often than at 160 votes, where they are below the base in 23 of 32, in line with their shrinking lead over the base.

Table 20: **Majority-vote accuracy on the math suite at** 16 **votes** (canonical-class votes; macro means (%) over five benchmarks; mean ± std over three seeds). A thicket member contributes its first $16/K$ completions. Every other method contributes the first 16 completions of its pool, so the base and GRPO-single-adapter rows are constant across K by construction. Bold: best method per model and K by displayed mean (ties bold both); underlined: second best.

Model	Method	K=2	K=4	K=8	K=16
Qwen2.5-1.5B	Base	44.6 ±0.3	44.6 ±0.3	44.6 ±0.3	44.6 ±0.3
	GRPO-single (matched)	47.9 ±1.4	46.6 ±1.7	46.1 ±0.4	45.0 ±0.1
	GRPO-single-adapter	46.0 ±1.7	46.0 ±1.7	46.0 ±1.7	46.0 ±1.7
	Thicket	**48.1 ±1.0**	**47.5 ±0.9**	**46.9 ±0.6**	**46.4 ±1.0**
Qwen2.5-3B	Base	56.9 ±0.9	56.9 ±0.9	56.9 ±0.9	56.9 ±0.9
	GRPO-single (matched)	56.9 ±1.0	**57.7 ±0.8**	**57.8 ±0.7**	56.6 ±0.9
	GRPO-single-adapter	55.4 ±3.2	55.4 ±3.2	55.4 ±3.2	55.4 ±3.2
	Thicket	**57.6 ±1.0**	**57.7 ±0.2**	57.6 ±0.6	**57.8 ±0.6**
Qwen2.5-7B	Base	**66.1 ±0.9**	66.1 ±0.9	**66.1 ±0.9**	66.1 ±0.9
	GRPO-single (matched)	65.5 ±1.1	65.0 ±0.9	65.1 ±0.7	**66.3 ±0.4**
	GRPO-single-adapter	65.0 ±1.2	65.0 ±1.2	65.0 ±1.2	65.0 ±1.2
	Thicket	**66.1 ±0.9**	**66.5 ±0.2**	**66.1 ±0.5**	66.0 ±0.6
Llama-3.1-8B	Base	46.6 ±0.7	46.6 ±0.7	46.6 ±0.7	46.6 ±0.7
	GRPO-single (matched)	**49.7 ±0.7**	**51.2 ±1.4**	47.7 ±0.8	46.4 ±0.6
	GRPO-single-adapter	48.4 ±1.6	48.4 ±1.6	48.4 ±1.6	**48.4 ±1.6**
	Thicket	49.3 ±2.6	49.9 ±0.8	**48.7 ±0.2**	48.2 ±0.7

Table 21: **Coverage at** 16 **votes** (at least one correct completion among the 16 pooled, with the first-16 convention of Table 20) on the math and code suites; macro means (%); mean ± std over three seeds. Bold: best method per model, suite, and K; underlined: second best.

		Coverage, math				Coverage, code			
Model	Method (K=)	2	4	8	16	2	4	8	16
Qwen2.5-1.5B	Base	73.1 ±0.4	73.1 ±0.4	73.1 ±0.4	73.1 ±0.4	61.3 ±1.1	61.3 ±1.1	61.3 ±1.1	61.3 ±1.1
	GRPO-single (matched)	**74.7 ±0.8**	**74.4 ±1.5**	73.7 ±0.7	73.3 ±0.5	**61.4 ±2.2**	61.4 ±1.1	**62.5 ±0.8**	61.9 ±0.9
	GRPO-single-adapter	72.8 ±1.5	72.8 ±1.5	72.8 ±1.5	72.8 ±1.5	58.2 ±0.8	58.2 ±0.8	58.2 ±0.8	58.2 ±0.8
	Thicket	74.5 ±0.6	74.2 ±0.8	**74.3 ±0.6**	**73.9 ±0.7**	**61.4 ±1.9**	**62.1 ±1.1**	62.4 ±0.7	**63.5 ±0.3**
Qwen2.5-3B	Base	**80.6 ±0.0**	**80.6 ±0.0**	**80.6 ±0.0**	80.6 ±0.0	**68.1 ±0.5**	**68.1 ±0.5**	68.1 ±0.5	68.1 ±0.5
	GRPO-single (matched)	78.8 ±0.8	79.8 ±0.9	80.5 ±0.3	**81.0 ±0.2**	67.3 ±2.1	65.5 ±1.5	67.9 ±0.4	68.0 ±1.2
	GRPO-single-adapter	76.5 ±1.9	76.5 ±1.9	76.5 ±1.9	76.5 ±1.9	64.9 ±0.9	64.9 ±0.9	64.9 ±0.9	64.9 ±0.9
	Thicket	78.9 ±0.9	80.1 ±0.8	80.1 ±0.6	80.4 ±0.2	66.8 ±0.6	67.8 ±0.3	**68.8 ±1.0**	**68.6 ±1.0**
Qwen2.5-7B	Base	**84.6 ±0.3**	**84.6 ±0.3**	84.6 ±0.3	84.6 ±0.3	**71.6 ±0.3**	**71.6 ±0.3**	**71.6 ±0.3**	71.6 ±0.3
	GRPO-single (matched)	83.8 ±0.7	83.6 ±0.3	84.3 ±0.4	**84.8 ±0.2**	69.8 ±0.7	71.0 ±0.5	71.4 ±0.7	71.8 ±0.7
	GRPO-single-adapter	83.2 ±0.6	83.2 ±0.6	83.2 ±0.6	83.2 ±0.6	68.6 ±0.6	68.6 ±0.6	68.6 ±0.6	68.6 ±0.6
	Thicket	83.9 ±0.0	84.4 ±0.2	**84.9 ±0.8**	84.2 ±0.9	71.2 ±0.0	71.2 ±0.5	71.3 ±0.7	**71.9 ±0.4**
Llama-3.1-8B	Base	69.6 ±0.8	69.6 ±0.8	69.6 ±0.8	69.6 ±0.8	**66.2 ±0.4**	66.2 ±0.4	**66.2 ±0.4**	66.2 ±0.4
	GRPO-single (matched)	74.0 ±0.4	**76.6 ±0.5**	72.2 ±0.8	69.7 ±0.8	64.2 ±1.8	64.9 ±0.4	65.6 ±0.9	66.1 ±0.1
	GRPO-single-adapter	72.8 ±1.4	72.8 ±1.4	72.8 ±1.4	72.8 ±1.4	64.3 ±1.4	64.3 ±1.4	64.3 ±1.4	64.3 ±1.4
	Thicket	**74.6 ±1.7**	75.7 ±1.5	**75.3 ±0.9**	**74.1 ±0.6**	65.2 ±0.5	**67.2 ±0.2**	66.0 ±0.1	**67.1 ±0.3**

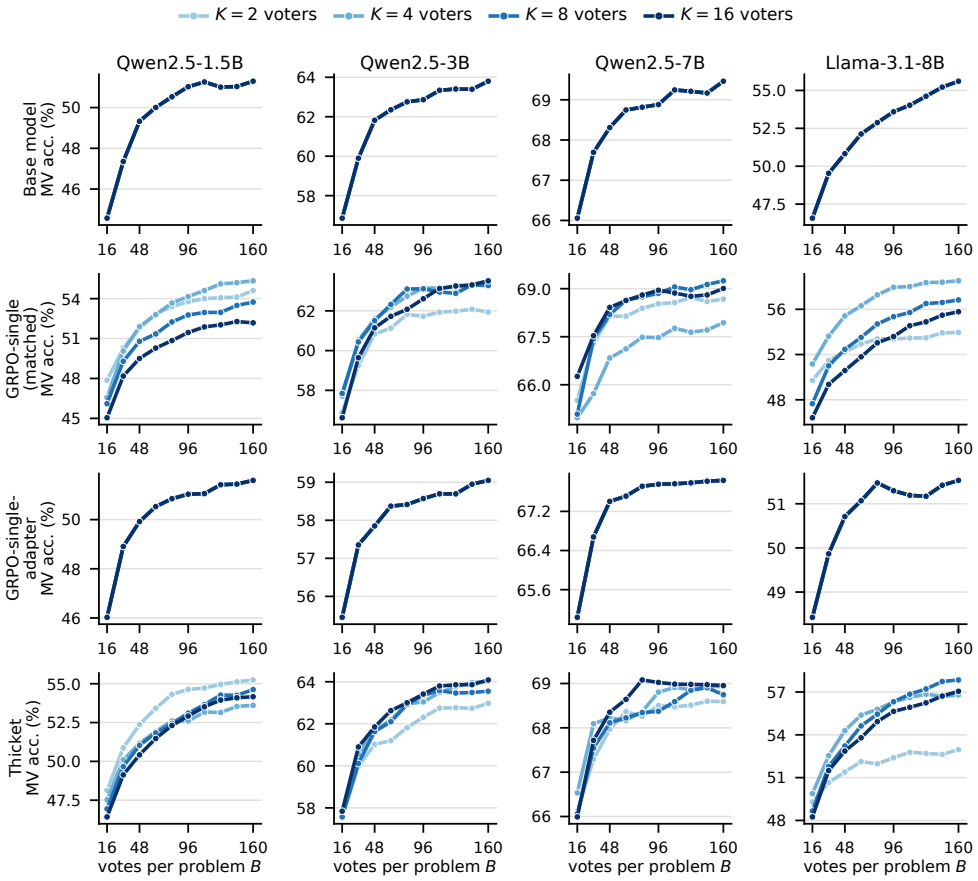


Figure 15: Majority-vote accuracy (math macro, mean over three seeds) as each method votes over its first B completions, $B = 16, 32, \ldots, 160$, per voter count K (line shade) and method (rows). The base-model and GRPO-single-adapter curves are identical at every K, so only the K=16 curve is visible in their rows. No bands are drawn here. Figure 16 draws ± 1 std bands.

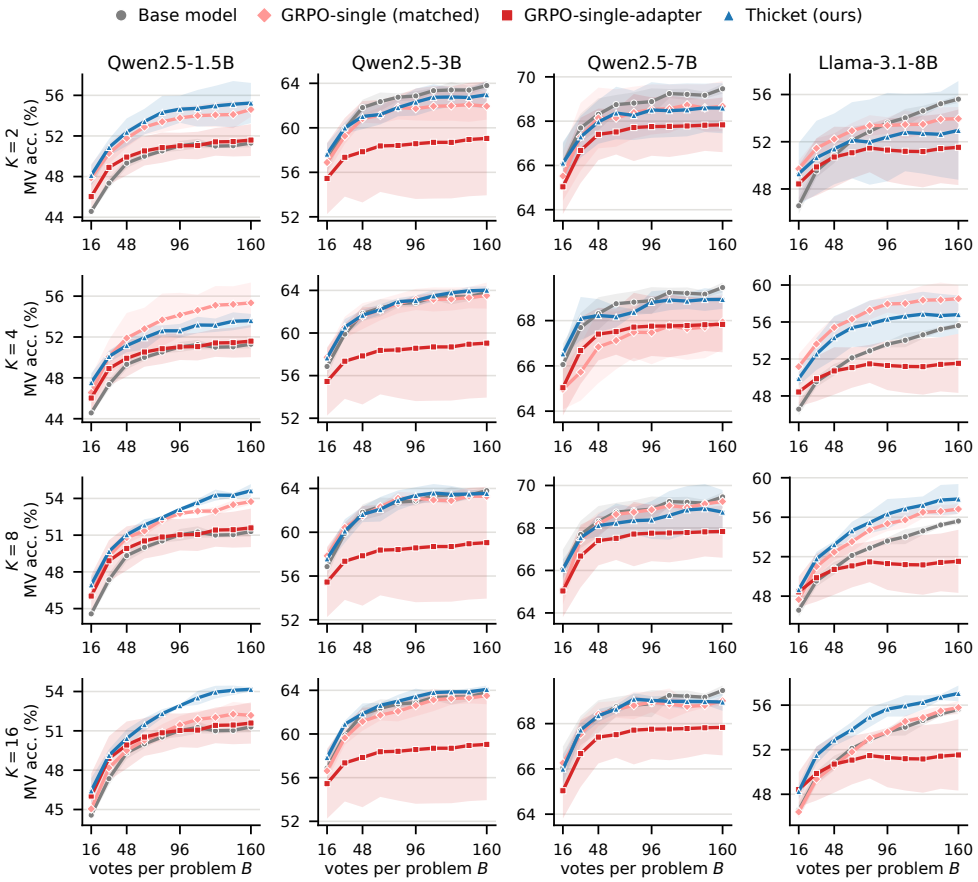


Figure 16: The same curves as Figure 15, with the four methods overlaid in each panel, per voter count K (rows) and model (columns). Bands: ± 1 std over seeds.

G DISJOINT RANDOM SHARDS VS. SUBJECT-SPECIALIST SHARDS

Setup. This ablation (Section 4.7) asks whether the thicket's diversity should be *engineered* rather than random. The MATH training pool is partitioned by its subject metadata into seven specialist shards (Algebra, Counting & Probability, Geometry, Intermediate Algebra, Number Theory, Prealgebra, Precalculus), equalized at 184 examples per adapter (the size of the smallest subject). The control thicket uses seven *random* disjoint shards of the same size, drawn from the same $2{,}000$-problem pool. Both arms use the GRPO recipe of Table 4, so each member trains for 46 steps. Each member contributes 23 samples per problem ($7 \times 23 = 161$ completions per problem), and pools are scored with the protocol of Appendix B, including confidence-filtered voting. The ablation uses a single seed (42) for four models.

Results. Table 22 breaks Table 3 down by benchmark. Across the 20 (model, benchmark) cells, random shards give the higher majority vote in 11, subject shards in 6, and 3 tie. Subject shards do best on Countdown for Qwen2.5-3B ($+2.2$ p.p.), while random shards do best on Knights & Knaves ($+1.4$ to $+3.8$ p.p. on three models). The specialists' only consistent edge is a small one on GSM8K (subject $\geq$ random on all four models, by at most 0.3 p.p.). On MATH500, despite its matching subject labels, random shards are never worse.

Table 22: Per-benchmark majority vote and confidence-filtered vote (%) for subject-specialist vs. random disjoint shards at $K{=}7$; single seed (42). Bold: better arm per row and vote type (ties bold both).

		Majority vote		Filtered vote	
Model	Benchmark	Subject	Random	Subject	Random
Qwen2.5-1.5B	MATH500	**69.4**	**69.4**	71.3	**71.5**
	GSM8K	**86.0**	85.8	85.5	**85.8**
	OlympiadBench	32.6	**33.6**	35.4	**37.8**
	Countdown	49.3	**49.8**	49.3	**49.8**
	Knights & Knaves	33.8	**35.2**	34.4	**35.2**
	Mean	54.2	**54.8**	55.2	**56.0**
Qwen2.5-3B	MATH500	77.4	**77.5**	**78.7**	78.2
	GSM8K	**91.4**	91.1	91.2	**91.4**
	OlympiadBench	43.3	**44.2**	**44.8**	44.2
	Countdown	**47.5**	45.3	**47.5**	45.3
	Knights & Knaves	54.9	**57.3**	**54.9**	54.5
	Mean	62.9	**63.1**	**63.4**	62.7
Qwen2.5-7B	MATH500	82.8	**83.3**	**83.2**	**83.2**
	GSM8K	**94.4**	**94.4**	94.2	**94.4**
	OlympiadBench	52.1	**52.9**	**53.1**	52.3
	Countdown	**54.6**	54.2	**54.6**	54.2
	Knights & Knaves	**61.7**	**61.7**	61.7	**63.7**
	Mean	69.1	**69.3**	69.3	**69.5**
Llama-3.1-8B	MATH500	66.5	**66.7**	67.0	**68.5**
	GSM8K	**91.7**	91.4	**91.5**	91.2
	OlympiadBench	**35.3**	35.2	**36.4**	36.0
	Countdown	53.2	**53.7**	52.0	**56.5**
	Knights & Knaves	38.9	**42.7**	**42.8**	42.7
	Mean	57.1	**57.9**	57.9	**59.0**