DISCRIMINATION IS NOT SELECTION: RETHINKING VERIFIER EVALUATION FOR MULTIMODAL TEST-TIME SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

A verifier can rank correct responses above incorrect ones on average and still select the wrong answer. This distinction matters for multimodal test-time scaling, where the value of additional sampling depends on recovering the correct answers it produces. We investigate this verification–selection gap and identify a mismatch between how verifiers are evaluated and how their scores are used. Pooled discrimination compares candidates across queries that never compete for selection, while within-query ranking averages over pairs even though the final decision depends on the score maximum. We formalize these distinctions and derive an exact decomposition of selection loss into incorrect candidates that strictly dominate and mixed-quality ties that are resolved incorrectly. Across multimodal reasoning benchmarks, informative verifier scores coexist with substantial failures to recover available correct answers, with different failure patterns across scoring interfaces. Broader candidate sampling and targeted interventions over frozen representations do not consistently eliminate this loss. Our findings motivate treating the recovery of available correct answers as an explicit evaluation target: verifiers for test-time scaling should be assessed through the decisions their scores induce, alongside their ability to discriminate correctness.

1 INTRODUCTION

Additional inference-time computation gives a reasoning model more chances to produce a correct answer. Turning those chances into better performance requires selecting the right response. Verifier-guided test-time scaling implements this idea through Best-of-N: a generator samples multiple candidates, and a verifier selects the highest-scoring response (Cobbe et al., 2021; Snell et al., 2025; Brown et al., 2024). Recent multimodal verifiers extend this approach to visual reasoning (Sun et al., 2025; Wang et al., 2026; Zhang et al., 2026). The promise of additional sampling therefore rests on a crucial connection: correct answers made available by generation must be recovered by selection.

Our study reveals a substantial loss at this connection. Across MathVista, MathVerse, MMStar, and MMMU, verifiers repeatedly leave correct responses unselected even when those responses are already present in the candidate pool. Figure 1(a,b) illustrates this failure in a constructed visual reasoning example: a correct response is available, but selection returns an incorrect alternative. Figure 1(c) shows the corresponding empirical gap on MathVista, where all three verifiers miss available correct answers on the same underlying candidate pools. These are failures that improved candidate coverage alone cannot explain. They motivate our central question: *how well does a verifier's ability to discriminate correctness translate into its ability to select a correct answer?*

Discrimination and selection place different demands on verifier scores. Pairwise discrimination metrics such as AUROC summarize how often correct responses receive higher scores than incorrect ones. When computed on pooled candidates, AUROC also compares responses to different questions, although such responses never compete during selection. Restricting comparisons to a single question removes that mismatch, but still averages over many candidate pairs. The final choice depends only on the highest-scoring candidates: one incorrectly favored response can defeat every correct alternative. Our analysis constructs candidate pools with identical within-query AUROC but opposite

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.

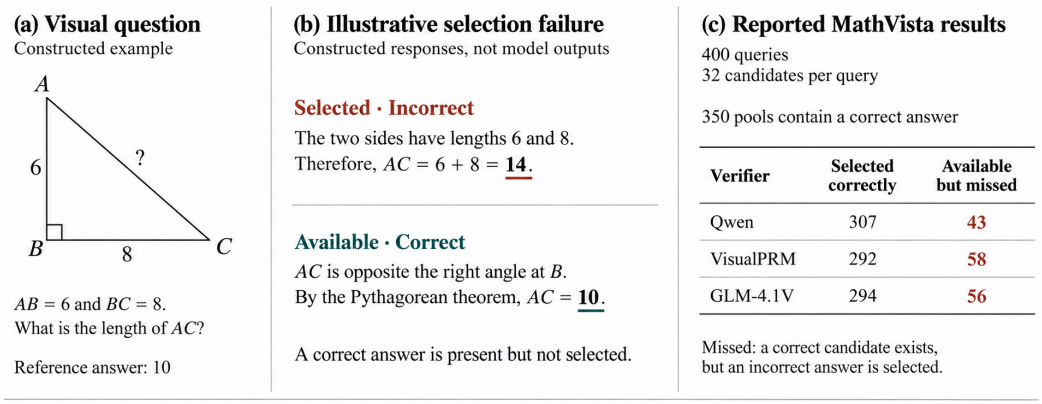

Figure 1: Available correct answers can remain unselected. (a,b) A constructed visual question and two illustrative responses show how selection can return an incorrect answer despite a correct alternative in the candidate pool. The example does not depict recorded model outputs. **(c)** Reported results on 400 MathVista queries with 32 candidates per query. Of the 350 pools containing a correct response, Qwen, VisualPRM, and GLM-4.1V select incorrectly on 43, 58, and 56 queries, respectively.

selection outcomes (Section 4). Informative ranking scores can therefore coexist with a failed final decision.

We investigate this verification–selection gap through a controlled study that holds candidate generation fixed when comparing verifiers. This lets us distinguish errors caused by the absence of a correct response from errors caused by failing to select one. Prior work evaluates judges within test-time scaling procedures (Zhou et al., 2025) and analyzes how verifier ROC properties affect sampling-based inference (Dorner et al., 2026). We complement these perspectives by connecting discrimination metrics to the specific decisions made on shared multimodal candidate pools. The focus is the accuracy left unrealized after generation has already supplied a correct answer.

Our analysis follows this connection from score summaries to the selected response. We first establish that pooled discrimination can change while every selection decision remains fixed, and that within-query ranking does not uniquely determine the winner. We then characterize selection loss exactly at the score maximum. An available correct answer is lost when an incorrect candidate strictly outranks every correct one, or when correct and incorrect candidates share the highest score and the tie is resolved incorrectly. This characterization accounts for the observed loss without assuming a particular cause of verifier error.

The experiments show that the gap recurs across benchmarks, while its form differs across scoring interfaces. Some interfaces frequently leave correct and incorrect responses tied at the top; others produce few ties but more strict incorrect maxima. Broader candidate sampling preserves limited recovery of available correct answers under the tested protocols. Targeted changes to training objectives, candidate-set context, and relative comparison also provide no clear accuracy gains with frozen representations.

2 Related Work

Test-time scaling and candidate selection. Test-time scaling uses additional computation to sample, search over, or refine candidate solutions (Wang et al., 2023; Yao et al., 2023; Madaan et al., 2023; Snell et al., 2025). Best-of-N couples sampling with a verifier that selects among the resulting responses (Cobbe et al., 2021). The benefits of generation and selection need not scale together: Brown et al. (2024) observe that sampling coverage continues to improve after common selection methods largely saturate on mathematical reasoning benchmarks. Complementary theory examines how coverage and imperfect rewards govern Best-of-N performance (Huang et al., 2025), and characterizes the sample complexity of different scaling procedures (Huang et al., 2026). Our study conditions on generated candidate pools to distinguish missing correct responses from failures to select those already available.

Process supervision and multimodal verification. Process supervision trains verifiers using feedback on intermediate reasoning steps (Lightman et al., 2024). Math-Shepherd and progress-based verification develop scalable ways to construct such supervision (Wang et al., 2024; Setlur et al., 2025). ProcessBench evaluates the localization of the first incorrect step in mathematical solutions (Zheng et al., 2025), while VerifyBench evaluates reference-conditioned correctness judgments across STEM domains (Li et al., 2026). For multimodal reasoning, VisualPRM learns step-level rewards and evaluates their utility through Best-of-N selection (Wang et al., 2026). MM-Verify trains an outcome verifier on synthetic chain-of-thought verification data and uses it to filter sampled solutions (Sun et al., 2025). These advances motivate examining how the scores produced by different verification interfaces translate into final selection decisions.

Evaluating judges for deployment. Judge evaluations measure agreement with human assessments (Liu et al., 2023; Kim et al., 2024), examine evaluation biases (Zheng et al., 2023), and test correctness judgments on challenging response pairs (Tan et al., 2025). More directly related to our setting, JETTS evaluates judges through reranking, beam search, and refinement; its reranking metric measures improvement over greedy decoding relative to oracle headroom (Zhou et al., 2025). ROC-n-reroll characterizes Best-of-N and rejection sampling through query-level ROC curves, relating verifier imperfection to expected selection performance (Dorner et al., 2026). We complement these studies with a diagnosis of frozen multimodal candidate pools. We use query centering to test changes in pooled discrimination that preserve every decision, compare within-query rankings with final choices, and account for selection loss through strict incorrect maxima and incorrectly resolved top-score ties. Our selection efficiency follows the headroom-normalization perspective used in JETTS, with an empirical single-draw baseline in place of greedy decoding.

3 Separating Generation from Selection

A wrong final answer can arise for two reasons: sampling never produced a correct response, or the verifier failed to select one that was already available. We distinguish these failures by comparing verifiers on the same generated candidates. This lets us measure the accuracy lost at selection without changing what generation makes available.

3.1 A Controlled Selection Problem

For each multimodal query $q \in \mathcal{Q}$, we sample N responses using a fixed generation policy and retain their order as $\mathcal{Y}_q^{(N)} = (y_{q,1}, \ldots, y_{q,N})$. Each response has a correctness label $z_{q,i} \in \{0, 1\}$, used only for evaluation. Queries, candidates, their order, and correctness labels are held fixed across matched verifier comparisons.

A verifier V assigns a scalar score $s_{q,i}$ to each response, either by scoring responses individually or by processing the candidate pool jointly. Selection returns a response with the highest score. To make this decision well defined when scores are tied, let T_q^V denote the set of maximum-score indices and τ a fixed tie-breaking rule:

$$T_q^V = \arg\max_{1 \le i \le N} s_{q,i}, \qquad \hat{i}_q^V = \tau(T_q^V). \tag{1}$$

Our primary protocol selects the smallest candidate index among tied maxima.

3.2 Available Answers and Selection Loss

For each question, we record whether the pool contains a correct response, $o_q = \max_i z_{q,i}$, and whether the selected response is correct, $a_q^V = z_{q,\hat{i}_q^V}$. Averaging these indicators gives

$$\mathrm{Pass@}N = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} o_q, \qquad \mathrm{Acc}_N^V = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} a_q^V. \tag{2}$$

Here $\mathrm{Pass@}N$ is the accuracy an oracle could achieve by choosing from the sampled pools, while Acc_N^V is the verifier's actual selection accuracy. Their difference is the fraction of all evaluation

questions for which a correct response is available but not selected:

$$\mathrm{Gap}_N^V = \mathrm{Pass@}N - \mathrm{Acc}_N^V = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} o_q(1 - a_q^V). \tag{3}$$

We use this *selection loss* as our primary diagnostic. For example, in Figure 1(c), VisualPRM selects correctly on 292 of 400 MathVista questions, although 350 pools contain a correct response. The 58 missed questions give a selection loss of 14.5 percentage points.

Every final error thus belongs to one of two groups: a generation failure, because the pool contains no correct response, or a selection failure, because an available correct response is missed. On fixed candidate pools, changing the verifier can affect only the second group.

3.3 Measuring the Gain from Selection

Selection loss measures missed accuracy. We also ask how much of the improvement made possible by sampling the verifier realizes. For this comparison, we use a frozen reference pool of M responses per query, generated by the same policy. If k_q of these responses are correct, the empirical single-candidate baseline is $A_1 = |\mathcal{Q}|^{-1}\sum_{q\in\mathcal{Q}} k_q/M$. This is the expected accuracy of drawing one response uniformly from each reference pool. The reference baseline remains fixed across matched verifier and budget comparisons.

Selection achieves a gain of $\mathrm{Acc}_N^V - A_1$, while the sampled pools offer at most $\mathrm{Pass@}N - A_1$. We adapt the oracle-normalization of Zhou et al. (2025), replacing greedy decoding with A_1, and define *selection efficiency* as the fraction of available headroom recovered:

$$\mathrm{SE}_N^V = \frac{\mathrm{Acc}_N^V - A_1}{\mathrm{Pass@}N - A_1}, \qquad \mathrm{Pass@}N > A_1. \tag{4}$$

A value of one reaches the sampled oracle; zero matches the reference baseline, and negative values fall below it. Because a small denominator can magnify modest accuracy differences, we interpret selection efficiency alongside absolute accuracy and selection loss.

4 Diagnosing Selection Failures

Section 3 measures how often selection misses an available correct answer. We now examine how these failures can coexist with informative verifier scores. Our diagnosis asks whether discrimination metrics capture the comparisons used during selection, whether a good pairwise ranking determines a correct winner, and which score configurations account for the missed answers. All analyses use the same frozen candidate pools and the selection rule defined above.

4.1 Testing Pooled Discrimination with Unchanged Decisions

Pooled AUROC, denoted AUC_g, measures how often a correct response scores above an incorrect one across the evaluation set, with half credit for ties. It includes comparisons between responses to different questions, even though those responses never compete for selection. It can therefore reflect score differences that do not affect which answer is returned.

We test this discrepancy by subtracting each question's mean score from all of its candidates:

$$\widetilde{s}_{q,i} = s_{q,i} - \mu_q, \qquad \mu_q = \frac{1}{N}\sum_{j=1}^{N} s_{q,j}. \tag{5}$$

This changes neither the ordering nor the ties within a pool, so every selected response remains identical. Yet pooled AUROC can change because scores from different questions are now compared at different offsets (Appendix A.1). We denote the AUROC after centering by AUC_c and report the contrast $\mathrm{GWG}_c = \mathrm{AUC}_g - \mathrm{AUC}_c$. Any change exposes sensitivity of pooled evaluation to score offsets that are irrelevant to selection. Centered AUROC still pools questions, so this contrast does not separate useful from useless verifier information.

4.2 Testing Whether Pairwise Ranking Determines the Winner

We next restrict comparisons to responses answering the same question. Let $\mathcal{Q}_{\mathrm{mix}}$ contain the questions whose pools include both correct and incorrect responses. For each such question, $\mathrm{AUC}_w(q)$ is the fraction of all correct–incorrect pairs in which the correct response scores higher, again assigning half credit to ties. We report $\mathrm{AUC}_w^{\mathrm{macro}}$, the mean of these values with equal weight for each mixed question.

This removes cross-question comparisons, but the metric still summarizes many pairs while selection returns only the highest-scoring response. Consider a constructed pool with two correct and ten incorrect responses, all with distinct scores. Number ranks from highest to lowest. If the correct responses occupy ranks 1 and 4, within-query AUROC is 0.90 and the winner is correct. Placing them at ranks 2 and 3 leaves AUROC at 0.90, but the winner is now incorrect. Thus even identical pairwise quality can lead to opposite selection outcomes. Appendix A.2 also constructs pools whose AUROC approaches one while selection fails.

Within-query AUROC is therefore a measure of local ranking quality; its relationship with selection accuracy must be examined empirically. It does not by itself identify whether an available correct answer is recovered.

4.3 Accounting for Selection Loss at the Top Score

To account for the actual failures, we compare the highest incorrect score with the highest correct score. For each mixed question, define

$$\Delta_q = \max_{i:z_{q,i}=0} s_{q,i} - \max_{j:z_{q,j}=1} s_{q,j}. \tag{6}$$

An available correct answer can be missed in two ways:

1. **An incorrect answer scores strictly higher.** When $\Delta_q > 0$, every highest-scoring response is incorrect. Selection fails regardless of how ties among those responses are resolved.
2. **A tie at the highest score is resolved incorrectly.** When $\Delta_q = 0$, correct and incorrect responses share the maximum. A failure occurs only if the specified tie rule selects an incorrect one.

If $\Delta_q < 0$, every highest-scoring response is correct and selection succeeds. The two failure events therefore account exactly for the selection loss defined in Section 3:

$$\mathrm{Gap}_N^V = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}_{\mathrm{mix}}} \left[\mathbf{1}\{\Delta_q > 0\} + \mathbf{1}\{\Delta_q = 0,\ a_q^V = 0\} \right]. \tag{7}$$

The denominator counts all evaluation questions. All-correct pools cannot lose a correct answer, and all-incorrect pools contribute generation failure rather than selection loss.

For tied maxima, we also record the top-set size and its fraction of correct responses, $p_{q,\mathrm{top}}^V$. This fraction is the expected selection accuracy under uniform random tie-breaking; the primary results use the fixed candidate-index rule. A large tied set is therefore insufficient evidence of failure without its correctness composition and the rule used to choose from it.

These diagnostics connect score evaluation to the decisions that leave correct answers unselected. Centering tests the relevance of pooled comparisons, within-query AUROC assesses local ranking, and the maximum-score analysis accounts for the observed loss. The final accounting identifies which selection events fail, without establishing why the verifier assigned those scores.

5 Experiments

We investigate how much available accuracy is lost during selection, whether discrimination metrics reveal this loss, and which decisions at the score maximum account for it. We then test whether targeted changes to scoring recover the missed answers.

5.1 Experimental Design

Controlled comparisons. We evaluate fixed 400-question subsets of MathVista, MathVerse, MM-Star, and MMMU. Qwen3-VL-8B-Instruct (Bai et al., 2025) generates 32 responses per question

Table 1: Informative discrimination coexists with missed correct answers. Each benchmark uses 400 questions and the same 32 candidates per question for all verifiers. Group headings give the shared oracle availability. Loss is the availability–accuracy gap in percentage points (pp); headroom recovered is $100 \times \mathrm{SE}_{32}$, relative to the empirical single-candidate baseline. Within-query AUROC is macro-averaged over mixed questions. Entries are point estimates; shading identifies selection loss, not statistical significance.

	Selection outcomes			AUROC	
Verifier	Accuracy (%)	Headroom recovered (%)	Lost accuracy (pp)	Pooled	Within-query
MathVista *Correct answer available: 350/400 (87.50%)*					
Qwen pointwise	76.75	30.2	10.75	0.733	0.668
VisualPRM	73.00	5.8	14.50	0.704	0.620
GLM-4.1V	73.50	9.1	14.00	0.757	0.660
MathVerse *Correct answer available: 267/400 (66.75%)*					
Qwen pointwise	42.25	14.9	24.50	0.633	0.583
VisualPRM	42.25	14.9	24.50	0.667	0.574
GLM-4.1V	37.75	−0.8	29.00	0.619	0.564
MMStar *Correct answer available: 347/400 (86.75%)*					
Qwen pointwise	67.25	11.7	19.50	0.647	0.590
VisualPRM	69.00	19.7	17.75	0.695	0.615
GLM-4.1V	66.25	7.2	20.50	0.663	0.580
MMMU *Correct answer available: 296/400 (74.00%)*					
Qwen pointwise	52.25	22.4	21.75	0.746	0.653
VisualPRM	47.75	6.4	26.25	0.645	0.539
GLM-4.1V	47.25	4.6	26.75	0.706	0.606

using a shared prompt and decoding policy. Each verifier receives the same candidates in the same order, so differences in selected accuracy reflect selection on a common set of available answers. These experiments use a controlled evaluation protocol rather than the official benchmark pipelines; generation settings, subset construction, and prompts are in Appendix B.

We compare a Qwen3-VL pointwise judge using the generator's checkpoint, GLM-4.1V (Team et al., 2026), and VisualPRM (Wang et al., 2026). Qwen and GLM score responses individually; VisualPRM processes the candidate pool through its native verification interface. Each returns the highest-scoring response, with exact ties resolved by the smallest candidate index. Score comparisons are made within each verifier's interface.

Evaluation. A shared, frozen rule-based evaluator assigns correctness labels independently of verifier scores. We report candidate availability, selected accuracy, selection loss, and headroom conversion (Section 3), together with the score diagnostics in Section 4. Accuracy and loss use all questions; within-query AUROC and extreme-margin statistics use pools containing both correct and incorrect responses. Confidence intervals resample whole questions, retaining their candidates together, and matched comparisons use paired resampling (Appendix B.7).

5.2 Available Answers Remain Unselected as Sampling Expands

The first question is how much accuracy is lost after generation has already supplied a correct answer. Across the 12 benchmark–verifier settings, selection leaves between 10.75 and 29.00 percentage points of available accuracy unrealized (Table 1). For example, VisualPRM returns 292 correct answers on MathVista even though 350 of the 400 pools contain one. Such missed selections account for approximately 42–61% of final errors across the full matrix (Table 8). The problem therefore extends beyond questions for which generation never succeeds.

The gain over the empirical single-candidate baseline is also limited: headroom conversion ranges from -0.8% to 30.2%. This occurs despite pooled and within-query AUROC point estimates above

Table 2: A larger pool supplies more correct answers without a comparable selection gain. Two independently sampled MathVista blocks and their union use the same 400 questions. Availability and accuracy are percentages; the last row reports percentage-point changes from Block 1 to the union. Full intervals appear in Table 7.

Candidate pool	N	**Correct available** Oracle (%)	**Qwen selected** Accuracy (%)	**VisualPRM selected** Accuracy (%)
Block 1 (1–32)	32	87.50	76.75	73.00
Block 2 (33–64)	32	88.00	77.25	74.00
Union (1–64)	64	89.75	76.50	73.25
Change: union − Block 1	–	+2.25	−0.25	+0.25

0.5 in every setting. Before examining this mismatch, we test whether the selection loss persists when the candidate sample changes.

More candidates increase availability without a comparable accuracy gain. We generate a second, independent 32-candidate MathVista block on the same 400 questions. Both blocks exhibit a substantial gap between availability and selection (Table 2). Combining them increases the number of pools containing a correct response from 350 to 359. Yet Qwen's number of correct selections changes from 307 to 306, and VisualPRM's from 292 to 293. Additional correct answers thus produce little net accuracy gain in this experiment. The comparison tests fresh candidate samples and a larger budget on shared questions; it does not constitute replication on a new question set.

Smaller-budget controls support the same concern. Across 100 random subpool draws at each of $N \in \{2, 4, 8, 16\}$, mean headroom conversion does not improve systematically as more candidates are retained (Table 6). A separate, 120-question GLM-generated MathVista control leaves a 37.5-point selection gap for the Qwen judge, with a 95% interval of [28.3, 46.7] (Appendix C.5). Answer-frequency aggregation also leaves substantial headroom unused (Table 9). These controls extend the finding beyond one candidate sample while retaining the limited scope of the tested generators and selectors.

5.3 Discrimination Summaries Do Not Determine Selection Utility

Can conventional discrimination metrics reveal the missed accuracy? Query centering provides a direct control: it changes the relative score offsets of different questions while preserving every selected answer. On MathVista, Qwen's pooled AUROC changes from 0.733 to 0.472 under this transformation. Across all 12 settings, the decrease ranges from 0.109 to 0.268, with every reported 95% interval above zero (Table 5). A substantial change in the metric can therefore occur with no change in selection. Centered AUROC remains a pooled statistic; the difference does not partition verifier information into useful and useless components.

Restricting comparisons to the same question brings the metric closer to the selection task. Within-query AUROC is positively associated with headroom conversion across the 12 settings, although these associations are descriptive because benchmarks and verifiers recur across cells (Appendix C.2). Positive ranking quality nevertheless coexists with little realized gain. On MathVerse, for example, GLM obtains within-query AUROC 0.564 while its headroom conversion is approximately zero. The construction in Section 4 establishes why this can happen: pairwise quality does not uniquely determine the highest-scoring response. The next analysis examines that response directly.

5.4 The Score Maximum Reveals Distinct Failure Patterns

An available correct answer is missed when an incorrect response strictly outranks every correct one, or when a mixed-quality tie at the highest score is resolved incorrectly (Equation 7). We inspect these events to understand how similar aggregate losses arise from different scoring behaviors.

Table 3 shows a clear contrast on MathVista's 154 mixed questions. Qwen frequently assigns the same maximum score to correct and incorrect responses: mixed ties occur on 66.2% of these questions. VisualPRM almost always separates their extremes, but assigns a strictly higher score to an incorrect

Table 3: Verifiers differ in what happens at the highest score. Statistics use the same 154 MathVista questions containing both correct and incorrect candidates, with $N = 32$. Strict incorrect maxima always cause a selection failure; mixed top ties do so only when resolved incorrectly. Brackets give 95% query-level bootstrap intervals.

Property of the score maximum	Qwen	VisualPRM	GLM-4.1V
Incorrect score strictly higher (%) $\Delta_q > 0$	6.5 [3.2, 10.4]	37.7 [30.5, 45.5]	18.2 [11.7, 24.0]
Correct and incorrect tied at top (%) $\Delta_q = 0$	66.2 [58.4, 73.4]	0.6 [0.0, 1.9]	40.9 [33.1, 49.4]
Mean number of top-scoring candidates	21.99 [20.07, 23.67]	1.09 [1.04, 1.14]	10.05 [8.83, 11.40]

Table 4: Targeted interventions do not establish a net accuracy gain. Each row compares an intervention with its own matched reference using frozen representations. Absolute accuracies are percentages; paired changes and their 95% bootstrap intervals are in percentage points. Reference accuracies belong to the matched protocol in each row; comparisons across intervention families are not controlled.

Intervention	Matched reference	Accuracy (%) Reference → new	Change (pp) [95% CI]
MathVista: 400 questions, 32 candidates per question			
Selection-aware objectives (Full SAVR)	Vanilla head	79.00 → 78.75	−0.25 [−2.25, 1.75]
Candidate-set context (SCES-v1)	Pointwise head	79.00 → 78.25	−0.75 [−2.75, 1.00]
MathVista: 200 validation questions, 16 candidates per question			
Pairwise replacement (J2S-v1)	Original judge	70.00 → 70.00	0.00 [−6.00, 6.00]
Residual correction (J2S-v1)	Original judge	70.00 → 70.00	0.00 [−1.50, 1.50]

response on 37.7%. GLM exhibits both patterns. The contrast recurs across benchmarks (Table 10), describing the tested scoring and extraction interfaces rather than model families in general.

A mixed tie creates the possibility of failure; its resolution determines whether a correct answer is actually missed. To test the role of the fixed-index rule, we replace it with uniform random tie-breaking. Qwen's expected MathVista accuracy changes only from 76.75% to 75.84%, whereas oracle-best and oracle-worst choices within the same top sets span 85.0% to 59.5% (Table 11). The scores leave substantial correctness variation unresolved at the maximum, and randomization does not recover it. This analysis locates the failed decisions without identifying the cause of the underlying scoring errors.

5.5 Targeted Scoring Changes Do Not Establish an Accuracy Gain

The diagnosis motivates three intervention tests: emphasizing selection in the training objective, conditioning scores on the candidate set, and introducing explicit pairwise comparisons. We implement these changes using lightweight modules over frozen candidate representations. SAVR changes the training objective, SCES adds candidate-set context, and J2S replaces or adjusts a pointwise judge using a learned comparator. Each is evaluated against its own matched reference (Appendix E). SAVR and SCES use separately trained pointwise heads as their references. J2S is compared with its original judge on 200 validation questions at $N = 16$.

None of the four matched comparisons establishes an accuracy gain: every paired 95% interval includes zero (Table 4). This does not mean the interventions leave the scores or decisions unchanged. In the objective study, emphasizing the top of the ranking raises within-query AUROC from 0.757 to 0.768 without improving accuracy. In the comparison study, replacing the judge corrects 17 previously wrong selections but breaks 17 previously correct ones; residual correction changes one decision in

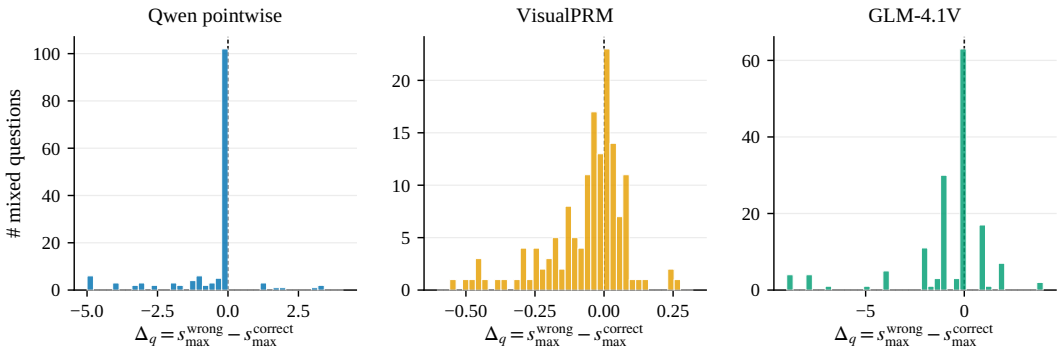

Figure 2: Verifier interfaces exhibit distinct extreme-score distributions. Histograms of Δ_q, the highest incorrect score minus the highest correct score, for the same 154 MathVista questions whose 32-candidate pools contain both correct and incorrect responses. Positive margins indicate strict incorrect maxima; negative margins indicate correct maxima. At zero, correct and incorrect responses share the highest score, and the outcome depends on tie-breaking. The dashed line marks zero. Score scales differ across verifiers.

each direction. Relative evidence can therefore alter the selected answer without producing a net gain.

Figure 2 complements Table 3 by showing how the best correct and incorrect responses compete. Qwen places substantial mass near zero; the exact event statistics confirm that correct and incorrect responses share the maximum on 66.2% of mixed questions. VisualPRM rarely ties correct and incorrect maxima but has a higher rate of strict incorrect maxima (37.7%), while GLM shows both patterns. These observations distinguish unresolved comparisons at the maximum from strictly incorrect orderings. Changing the tie-breaking rule can affect mixed top-score ties, but cannot correct a decision when an incorrect response strictly outranks every correct alternative. The broad contrast recurs across benchmarks (Table 10) and characterizes the tested scoring interfaces. The tested changes provide no clear evidence of reduced selection loss under frozen representations. Their intervals do not establish equivalence or exclude gains from stronger selectors, jointly adapted representations, or additional verification evidence.

Scope and measurement limitations. The primary matrix covers four 400-question subsets and three verifier interfaces; the larger-budget and second-generator controls cover only MathVista. All results depend on frozen correctness labels. In a blinded, two-annotator audit of 200 candidates, human–grader agreement is 84.5% and 82.5%, compared with 91.0% inter-annotator agreement (Appendix B.5). Audit-calibrated independent label perturbations retain positive selection gaps, but do not cover arbitrary correlated grading errors (Appendix B.6). The sampled oracle is an availability bound, and candidate count is only a proxy for inference cost. Our conclusions concern selection under these specified conditions.

6 Conclusion

Additional sampling creates opportunities for correct reasoning, but realizing those opportunities depends on selection. Across the multimodal settings studied here, verifiers leave substantial available accuracy unrealized despite informative correctness scores. Our diagnosis shows that discrimination metrics do not fully characterize the final choice: pooled AUROC can change while every decision remains fixed, and within-query ranking quality does not uniquely determine the winner. We account for the resulting selection loss through two events: an incorrect response strictly outranks every correct alternative, or a mixed-quality tie at the maximum is resolved incorrectly. The gap persists under the tested sampling controls, while lightweight interventions over frozen representations provide no clear accuracy gains. These findings motivate evaluating verifier scores together with the candidate-set decisions they induce, reporting both the availability of correct answers and their recovery through selection. A central challenge for future work is to turn improvements in verification into reliable gains in selected accuracy as sampling expands.

Ethics Statement

This work evaluates multimodal verifiers using existing benchmarks and pretrained models. No personal or sensitive information is collected. Human annotation is used only for a limited audit of model-generated responses. We follow the applicable licenses and usage conditions of the datasets and pretrained models used in this study.

Reproducibility Statement

We provide the necessary details for reproducibility in the paper and appendices. Appendix B describes the datasets, generation settings, prompts, verifier protocols, random seeds, and statistical analyses; Appendix A provides theoretical proofs, and Appendix E details the controlled intervention experiments.

AI Use Statement

The authors used generative AI exclusively for editorial support in preparing the manuscript, such as improving grammar, phrasing, clarity, and readability. It was not involved in the conception of the study, methodological development, experimental design or execution, data processing, result generation, statistical or scientific analysis, or the formulation of conclusions. All scientific contributions, technical details, derivations, experimental findings, and interpretations were created and validated by the authors. Responsibility for the final manuscript and all of its contents rests entirely with the authors.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

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

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, volume 37, pp. 27056–27087, 2024. doi: 10.52202/079017-0850.

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

Florian Eddie Dorner, Yatong Chen, André F Cruz, and Fanny Yang. Roc-n-reroll: How verifier imperfection affects test-time scaling. In *ICLR*, volume 2026, pp. 150895–150936, 2026.

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

Baihe Huang, Shanda Li, Tianhao Wu, Yiming Yang, Ameet Talwalkar, Kannan Ramchandran, Michael Jordan, and Jiantao Jiao. Sample complexity and representation ability of test-time scaling paradigms. In *ICLR*, volume 2026, pp. 54262–54298, 2026.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *EMNLP*, pp. 4334–4353, 2024.

Xuzhao Li, Xuchen Li, Shiyu Hu, Yongzhen Guo, and Wentao Zhang. VerifyBench: A systematic benchmark for evaluating reasoning verifiers across domains. In *AAAI*, volume 40, pp. 31796–31804, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, volume 2024, pp. 39578–39601, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *EMNLP*, pp. 2511–2522, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, volume 2024, pp. 23439–23554, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *ICLR*, volume 2025, pp. 60808–60838, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *ICLR*, volume 2025, pp. 10131–10165, 2025.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. MM-Verify: Enhancing multimodal reasoning with chain-of-thought verification. In *ACL*, pp. 14100–14115, 2025.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *ICLR*, volume 2025, pp. 63277–63303, 2025.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Haochen Li, Jiale Zhu, Jiali Chen, Jiaxing Xu, Jiazheng Xu, Jing Chen, Jinghao Lin, Jinhao Chen, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Ruiliang Lyu, Shangqin Tu, Sheng Yang, Shengbiao Meng, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wei Jia, Wenkai Li, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyu Zhang, Xinyue Fan, Xuancheng Huang, Yadong Xue, Yanfeng Wang, Yanling Wang, Yanzi Wang, Yifan An, Yifan Du, Yiheng Huang, Yilin Niu, Yiming Shi, Yu Wang, Yuan Wang, Yuanchang Yue, Yuchen Li, Yusen Liu, Yutao Zhang, Yuting Wang, Yuxuan Zhang, Zhao Xue, Zhengxiao Du, Zhenyu Hou, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026. URL https://arxiv.org/abs/2507.01006.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce llms step-by-step without human annotations. In *ACL*, pp. 9426–9439, 2024.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. VisualPRM400k: An effective dataset for training multimodal process reward models. In *ICLR*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurPS*, volume 36, pp. 11809–11822, 2023. doi: 10.52202/075280-0517.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, pp. 9556–9567, 2024.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, pp. 169–186, 2024.

Xinchen Zhang, Xiaoying Zhang, Youbin Wu, Yanbin Cao, Renrui Zhang, Ruihang Chu, Ling Yang, Yujiu Yang, and Guang Shi. Generative universal verifier as multimodal meta-reasoner. In *ICLR*, volume 2026, pp. 109211–109243, 2026.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *ACL*, pp. 1009–1024, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *NeurIPS*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

Yilun Zhou, Austin Xu, Peifeng Wang, Caiming Xiong, and Shafiq Joty. Evaluating judges as evaluators: The JETTS benchmark of LLM-as-judges as test-time scaling evaluators. In *ICML*, volume 267, pp. 79436–79471, 2025.

A Theoretical Analysis of Verification–Selection Mismatch

The following results formalize three distinctions underlying our analysis: pooled versus within-query comparison, pairwise ranking versus maximum-score selection, and candidate scaling versus extreme-score competition.

A.1 Query-wise Score Shifts Change Discrimination Metrics Without Changing Deployment Decisions

Proposition 1. For arbitrary $c_q \in \mathbb{R}$, define

$$s'_{q,i} = s_{q,i} + c_q.$$

Then every within-query ordering, exact tie, maximum-score set, selected index, and per-query $\mathrm{AUC}_w(q)$ remain unchanged, whereas pooled AUC_g need not be invariant.

Proof. For candidates i, j from the same query,

$$s'_{q,i} - s'_{q,j} = s_{q,i} - s_{q,j},$$

so all within-query comparisons and ties are preserved. Therefore, under the same tie-breaking rule,

$$T'_q = T_q, \qquad \hat{i}'_q = \hat{i}_q,$$

and every pairwise term contributing to $\mathrm{AUC}_w(q)$ is unchanged.

For candidates from different queries $q \neq q'$,

$$s'_{q,i} - s'_{q',j} = (s_{q,i} - s_{q',j}) + (c_q - c_{q'}),$$

so cross-query orderings can change. Since pooled AUC_g includes comparisons across different queries, it is not invariant to such query-wise shifts. □

Mean centering is a special case with $c_q = -\mu_q$. It can therefore change pooled discrimination metrics while preserving every deployment decision, which motivates separating score evaluation from the decisions induced by the scores.

A.2 Pairwise Ranking Alone Does Not Determine Extreme Selection

Proposition 2. For any integer $K \geq 2$, there exists a mixed candidate pool containing K incorrect candidates and one correct candidate such that

$$\mathrm{AUC}_w(q) = 1 - \frac{1}{K},$$

while top-1 selection is incorrect. Hence, arbitrarily strong within-query pairwise ranking does not uniquely determine the maximum-scoring candidate.

Proof. Construct a candidate pool with one correct candidate having score 0, $K-1$ incorrect candidates having score -1, and one additional incorrect candidate having score 1. The total number of incorrect candidates is therefore K.

The correct candidate scores higher than $K-1$ of the K incorrect candidates, yielding

$$\mathrm{AUC}_w(q) = \frac{K-1}{K} = 1 - \frac{1}{K}.$$

However,

$$s_{\max}^{\mathrm{wrong}} = 1 > 0 = s_{\max}^{\mathrm{correct}},$$

so

$$\Delta_q > 0$$

and the selected response is incorrect. Taking $K \to \infty$ shows that within-query pairwise ranking can approach perfect accuracy while selection still fails. □

A.3 Candidate Scaling Creates Extreme Competition Under Fixed Correct Scores

Proposition 3. Fix the best-correct score at c. Suppose $s_1^-, \ldots, s_m^-$ are i.i.d. incorrect-candidate scores with CDF F_-. Then

$$\Pr\left(\max_{1\le j\le m} s_j^- > c\right) = 1 - F_-(c)^m, \tag{8}$$

which is non-decreasing in m.

Proof. By independence,

$$\Pr\left(\max_{1\le j\le m} s_j^- \le c\right) = \prod_{j=1}^{m} \Pr(s_j^- \le c) = F_-(c)^m.$$

Taking the complement gives Eq. equation 8. Because

$$0 \le F_-(c) \le 1,$$

the probability is non-decreasing with m. □

This result isolates one conditional effect of candidate scaling: if the best-correct score and the incorrect-score distribution remain fixed, adding incorrect candidates cannot reduce the probability of a strict incorrect maximum. This corresponds to the failure event $\Delta_q > 0$, and does not describe mixed-quality ties at $\Delta_q = 0$.

The result is deliberately conditional and does not imply that increasing the candidate budget must decrease verifier accuracy or selection efficiency. Larger pools may simultaneously improve the best-correct score, introduce additional correct candidates, or alter the joint score distribution. Therefore, our scaling experiments examine whether limited headroom conversion persists as candidate availability increases, rather than assuming a universal monotonic scaling law.

Summary. Propositions 1 and 2 show that discrimination summaries and pairwise ranking statistics do not fully characterize the final selection decision. Proposition 3 provides a complementary explanation for why larger candidate pools can introduce additional extreme-score competition. Together, these results motivate evaluating verifier-guided scaling through the candidate-set decisions induced by verifier scores.

B Experimental Details

B.1 Candidate Generation

We evaluate verifier-guided test-time scaling under a frozen candidate-selection protocol. For each query q, a fixed generation policy first produces

$$\mathcal{Y}_q^{(N)} = (y_{q,1}, \ldots, y_{q,N}),$$

and all verifier evaluations operate on these frozen candidates. Thus, matched verifier comparisons isolate selection behavior from candidate-generation differences.

We evaluate four multimodal reasoning benchmarks: MathVista (Lu et al., 2024), MathVerse (Zhang et al., 2024), MMStar (Chen et al., 2024), and MMMU (Yue et al., 2024). Unless otherwise specified, candidates are generated using Qwen3-VL-8B-Instruct (Bai et al., 2025) with temperature 0.8, top-p $= 0.95$, and a maximum generation length of 512 tokens. The primary experiments use fixed 400-question subsets and $N = 32$ candidates per query. Candidate sampling uses fixed seed initialization for reproducibility. We do not claim bit-exact reproducibility across different hardware environments.

The MathVista, MathVerse, and MMStar subsets are frozen before generation and preserve benchmark structure through the corresponding stratified sampling procedures used in our experimental pipeline. MathVerse is restricted to the Vision Intensive version, MMStar uses proportional allocation over its

six L1 capability categories, and MMMU uses a fixed 400-question validation subset with single-image inputs. The same question IDs are used for generation, verifier scoring, correctness evaluation, and all reported diagnostics; no verifier-specific resampling or post-hoc filtering is performed.

For MathVista, the 400-question evaluation pool is divided into a 50-question exploratory subset and a disjoint 350-question confirmation subset. Repeated-subpool robustness draws candidate subsets at $N \in \{2, 4, 8, 16\}$ from the frozen $N = 32$ primary pools.

To test candidate-sampling replication and larger-budget behavior, we generate a second independently sampled MathVista block containing candidates 33–64. This block uses the same frozen decoding configuration with an independent seed base of 1041. We report the primary block $\mathcal{Y}^{1:32}$, the second block $\mathcal{Y}^{33:64}$, and their union $\mathcal{Y}^{1:64}$. Both blocks use the same 400 questions, so the second block provides a candidate-sampling replication rather than an independent question-set replication. Only Qwen pointwise and VisualPRM are evaluated in this $N = 64$ robustness analysis.

B.2 Generation Prompt

We use a unified generation template across all benchmarks rather than the official benchmark-specific query templates. Each formatted question is appended with:

> Solve the problem using the provided context and image when available.
> Write a short reasoning trace.
> Put the final answer on its own line as:
> Answer: <value>

The template is applied consistently to MathVista, MathVerse, MMStar, and MMMU to construct a controlled candidate distribution before verifier evaluation. Because this protocol differs from official benchmark evaluation pipelines, reported $\mathrm{Pass@}N$ and selection results should be interpreted under this controlled setting rather than as official benchmark scores.

B.3 Verifier Scoring Protocol

All verifiers score the same frozen candidate pools and do not alter generation. We evaluate three heterogeneous scoring interfaces: a Qwen3-VL pointwise verifier (Bai et al., 2025), VisualPRM (Wang et al., 2026), and GLM-4.1V (Team et al., 2026). The primary 4×3 matrix uses the same Qwen-generated candidate pools for all three interfaces.

Qwen pointwise. The Qwen judge uses the same Qwen3-VL-8B-Instruct checkpoint as the primary generator, making this a self-scoring configuration at the model-weight level. This configuration is included as one representative pointwise verification interface and is analyzed together with independent verifier interfaces.

Each query–candidate pair is judged independently with a fixed rubric covering visual grounding, reasoning validity, and answer quality. The model is instructed to return JSON containing the three criterion scores and an overall score, each on a 0–10 scale. The rubric defines requested evaluation criteria but does not provide candidate-specific correctness labels. Decoding is greedy (`do_sample=False`) with at most 192 new tokens.

The scalar verifier score is the model-reported JSON `score`. If that field cannot be parsed, we use the mean of the available three criterion scores; if JSON parsing fails entirely, we fall back to the first numerical value in the generated text. The resulting value is clamped to $[0, 10]$. No further normalization or rounding is applied before ranking.

GLM-4.1V. GLM-4.1V uses the same pointwise 0–10 judging rubric as Qwen so that the requested evaluation criteria are held fixed across the two pointwise judges. Decoding is greedy with a maximum of 1024 new tokens. Because GLM produces an explicit thinking segment, score parsing is applied only to the text following the closing `</think>` tag, preventing intermediate numbers in the reasoning trace from being interpreted as verifier scores.

The same JSON parser, fallback rules, and $[0, 10]$ clamping procedure used for Qwen are then applied. No further score normalization or rounding is performed.

VisualPRM. VisualPRM is evaluated through its native `select_best_response(..., return_scores=True)` interface. The model jointly receives the question, image, and candidate responses and returns one scalar score per response. We use these returned candidate scores directly; the aggregation of process-level evidence is therefore performed by the official VisualPRM interface rather than by a custom last-step, mean, minimum, or product rule. VisualPRM is loaded in bfloat16, and the returned scalar values are stored as floating-point numbers without additional rounding, calibration, or normalization.

Score comparison and ties. Raw score magnitudes are verifier-specific and are never compared across interfaces. All reported AUROC, ranking, and selection quantities are computed separately within each verifier.

Maximum-score sets use equality of stored verifier scores: $i \in T_q^V$ iff

$$s_{q,i} = \max_j s_{q,j}.$$

The deployed selector scans candidates in candidate-index order and updates the winner only under a strict score increase, so stored-score ties are resolved by the smallest candidate index. Pairwise AUROC likewise assigns tie credit only when stored scores are equal. For the extreme-margin analysis, $\Delta_q = 0$ is implemented numerically as $|\Delta_q| < 10^{-12}$, which corresponds to equality at the stored score precision used here.

No post-hoc score rounding, decimal truncation, softmax normalization, or tie tolerance is introduced before ranking. Qwen and GLM scores are model-generated decimal values stored as Python floating-point numbers, whereas VisualPRM uses its native returned floating-point scores. The different tie regimes observed in Section 5.4 therefore characterize these tested scoring interfaces, including their score generation and extraction procedures, rather than a downstream quantization step introduced by our analysis.

The separate generation-policy control in Appendix C.5 explicitly changes the generator; all other matched verifier comparisons retain the same frozen candidate pools.

B.4 Correctness Evaluation

Each generated candidate receives a binary correctness label $z_{q,i} \in \{0, 1\}$ from a frozen unified rule-based extractor and matcher shared across all four benchmarks. The evaluator does not use GPT-based judging or any evaluated verifier score. It performs answer extraction and normalization before matching against benchmark ground truth and supports multiple-choice, numerical, textual, fractional, and normalized LaTeX answer formats. For multiple-choice tasks, both option-letter and content-based matching are supported where appropriate.

The evaluator applies benchmark-format matching rules only and does not access verifier scores, candidate rankings, or selection outcomes. The same frozen correctness logic is used throughout the study. Labels are not modified after observing verifier scores, selection outcomes, or human audit results.

B.5 Human Audit of Correctness Labels

Because the reported diagnostics rely on binary correctness labels, we conduct a pre-specified human audit of 200 generated candidates, with 50 candidates per benchmark. Each benchmark contributes 25 label-stratified random candidates (S1) and 25 decision-critical candidates (S2). The S2 stratum targets cases most directly involved in the deployment diagnostics: selected candidates, maximum-scoring correct candidates, maximum-scoring incorrect candidates, top-set ties, and parse failures.

Two annotators independently label all 200 examples while blinded to the frozen grader outputs. A 100-example subset is re-annotated by both annotators to assess repeatability. Table **??** summarizes overall agreement.

Inter-annotator agreement is 91.0% with Cohen's $\kappa = 0.813$ on the full set and 98.0% with $\kappa = 0.938$ on the double-annotated subset.

[Tables unchanged]

The audit is used to characterize measurement uncertainty rather than to replace the primary evaluation protocol. We retain all frozen labels unchanged and use observed disagreements only for the sensitivity analysis below; no label is corrected post hoc after observing verifier outcomes.

B.6 Audit-Calibrated Label-Noise Sensitivity

We test whether the main conclusions are sensitive to the observed human–grader disagreements without changing any frozen primary label. For each benchmark, we estimate disagreement rates $p_{1\to0}$ and $p_{0\to1}$ from the audit and define an audit-calibrated conservative-rate setting by taking the larger rate observed across the S1 and S2 strata.

Candidate labels are then perturbed independently for 200 Monte Carlo draws per benchmark–verifier cell (seed 41), and pool-level $\mathrm{Pass@32}$, Acc_{32}^{V}, Gap_{32}^{V}, and SE_{32}^{V} are recomputed.

[Table unchanged]

Under this audit-calibrated independent-perturbation model, all benchmark–verifier cells remain positive across the sampled perturbations and no verifier reaches high headroom conversion. This analysis characterizes sensitivity to observed disagreements rather than providing a formal worst-case guarantee over arbitrary or correlated label errors. All primary results continue to use the original frozen evaluator outputs.

B.7 Statistical Analysis

All statistical analyses treat the question as the independent sampling unit. Candidate responses belonging to the same question are retained together in all question-level resampling procedures. Global verifier metrics are computed over candidate responses, whereas within-query metrics are computed within eligible queries and then macro-averaged.

Query-centered discrimination uses

$$\widetilde{s}_{q,i} = s_{q,i} - \frac{1}{N}\sum_{j=1}^{N} s_{q,j},$$

and the extreme margin is

$$\Delta_q = \max_{i:z_{q,i}=0} s_{q,i} - \max_{j:z_{q,j}=1} s_{q,j}.$$

Confidence intervals follow the resampling procedures used for each analysis pipeline. Query-centered GWG_c confidence intervals use 400 question-clustered bootstrap resamples with seed 41. Extreme-score binary-event statistics on mixed queries use 1,000 question-clustered bootstrap resamples.

Final SE_{32} uncertainty estimates and paired $\Delta\mathrm{SE}$ comparisons use 1,000 question-clustered bootstrap resamples with seed 41.

Repeated-subpool robustness is a separate candidate-sampling analysis: for each $N \in \{2, 4, 8, 16\}$, we draw 100 random candidate subsets from each frozen $N = 32$ pool and report the mean selection efficiency across draws. These candidate-subset draws are not bootstrap confidence intervals.

The independently sampled second MathVista block and the $N = 64$ union use 1,000 question-clustered bootstrap resamples, with paired same-question resampling for block-to-block differences. All matched intervention $\Delta\mathrm{Acc}$ confidence intervals likewise use 1,000 question-clustered bootstrap resamples with seed 41. Audit-calibrated label-noise sensitivity uses 200 independent perturbation draws per benchmark–verifier cell.

All reported bootstrap confidence intervals are percentile intervals based on the 2.5th and 97.5th percentiles.

C Additional Diagnostic Results

This section provides complementary diagnostics supporting the main verification–selection analysis. All verifier comparisons use frozen candidate pools and treat the question as the statistical unit. The purpose of these experiments is not to introduce additional benchmark metrics, but to identify when commonly used verification statistics align with or diverge from the final candidate-selection decision.

C.1 Complete Query-Centered Diagnostics

The main paper reports the deployment-oriented 4×3 matrix. Here we provide complementary score-structure diagnostics omitted from the main table. Table 5 reports pooled AUROC, query-centered pooled AUROC, their contrast GWG_c, and query-macro within-query AUROC for all 12 benchmark–verifier settings.

Table 5: Complete query-centered discrimination diagnostics. AUC_c is pooled AUROC after subtracting each query's mean verifier score, and $\mathrm{GWG}_c = \mathrm{AUC}_g - \mathrm{AUC}_c$. The centering transformation leaves all within-query orderings and deployed decisions unchanged. Confidence intervals for GWG_c are question-clustered bootstrap intervals.

Dataset	Verifier	AUC_g	AUC_c	GWG_c [95% CI]	$\mathrm{AUC}_w^{\mathrm{macro}}$
MathVista-400	Qwen pointwise	0.733	0.472	0.260 [0.215, 0.303]	0.668
	VisualPRM	0.704	0.515	0.189 [0.136, 0.231]	0.620
	GLM-4.1V	0.757	0.489	0.268 [0.230, 0.306]	0.660
MathVerse-400	Qwen pointwise	0.633	0.522	0.110 [0.074, 0.144]	0.583
	VisualPRM	0.667	0.517	0.149 [0.104, 0.196]	0.574
	GLM-4.1V	0.619	0.510	0.109 [0.077, 0.140]	0.564
MMStar-400	Qwen pointwise	0.647	0.491	0.156 [0.120, 0.190]	0.590
	VisualPRM	0.695	0.518	0.178 [0.137, 0.223]	0.615
	GLM-4.1V	0.663	0.478	0.184 [0.146, 0.218]	0.580
MMMU-400	Qwen pointwise	0.746	0.497	0.249 [0.214, 0.285]	0.653
	VisualPRM	0.645	0.501	0.145 [0.099, 0.189]	0.539
	GLM-4.1V	0.706	0.525	0.181 [0.152, 0.210]	0.606

Every GWG_c confidence interval is strictly positive: query centering changes pooled AUROC by 0.109–0.268 across all 12 settings while preserving all within-query orderings and deployed decisions.

The centered pooled values themselves lie in the range 0.472–0.525. We emphasize that this transformation is a diagnostic perturbation rather than a corrected verifier metric: AUC_c remains a pooled statistic and should not be interpreted as a direct measurement of within-query selection quality.

The MathVista-350 confirmation subset, which excludes the 50 development queries, shows the same centered-score pattern for the two primary verifiers:

$$\mathrm{GWG}_c^{\mathrm{Qwen}} = 0.250\,[0.206, 0.289], \qquad \mathrm{GWG}_c^{\mathrm{VisualPRM}} = 0.190\,[0.139, 0.241],$$

with $\mathrm{AUC}_w^{\mathrm{macro}} = 0.665$ and 0.629, respectively. This subset therefore confirms that the centering effect is not driven only by the 50 development queries.

For completeness, deployment-level uncertainty on the primary MathVista-400 pool gives

$$\mathrm{SE}_{32}^{\mathrm{Qwen}} = 0.302\,[0.166, 0.453], \qquad \mathrm{SE}_{32}^{\mathrm{VisualPRM}} = 0.058\,[-0.110, 0.227], \qquad \mathrm{SE}_{32}^{\mathrm{GLM}} = 0.091\,[-0.043, 0.235].$$

Paired question-level bootstrap further gives

$$\Delta\mathrm{SE}_{32}^{\mathrm{Qwen-VisualPRM}} = 0.244\,[0.057, 0.446], \qquad \Delta\mathrm{SE}_{32}^{\mathrm{Qwen-GLM}} = 0.211\,[0.049, 0.387].$$

Thus, the query-centering result and the observed MathVista differences in headroom conversion both persist under question-level resampling uncertainty.

C.2 Verification Metrics versus Selection Utility

Figure 3 provides the complete 12-point view of pooled and within-query discrimination versus SE_{32}. The descriptive Pearson correlations are 0.474 for AUC_g and 0.604 for AUC_w^{macro}.

These correlations indicate that verifier discrimination contains meaningful information about downstream selection. However, discrimination statistics alone do not fully characterize deployment behavior: benchmark–verifier pairs with comparable within-query ranking quality can still exhibit different levels of headroom conversion.

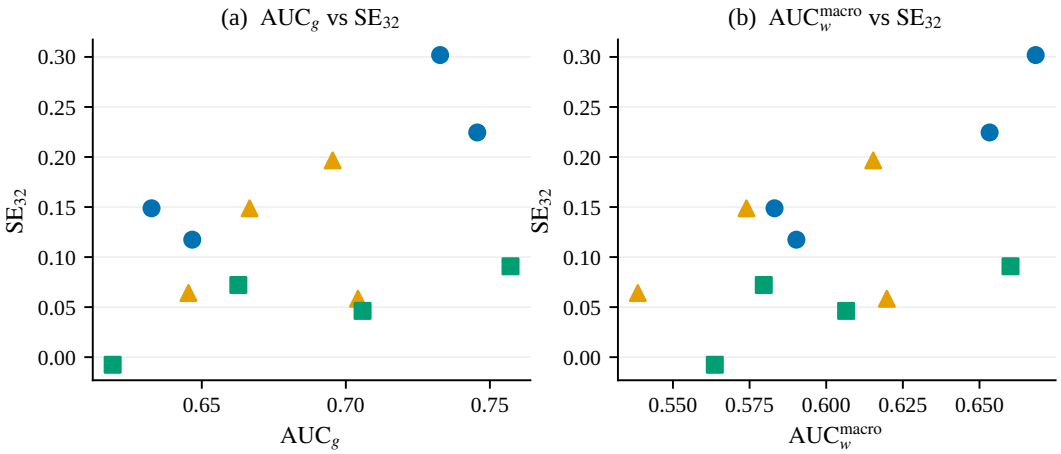

Figure 3: Verification discrimination is associated with, but does not fully characterize, selection utility. Each point is a benchmark–verifier pair from the complete 4×3 matrix. The left panel uses pooled AUROC and the right panel uses macro within-query AUROC. The descriptive correlations are $\rho = 0.474$ and $\rho = 0.604$, respectively. In particular, comparable within-query discrimination can still yield substantially different top-1 selection efficiency.

The counterexamples are visible in the full matrix. For example, MathVerse–GLM has $\mathrm{AUC}_w^{macro} = 0.564$ but $\mathrm{SE}_{32} = -0.008$, while on the same MMMU candidate pool Qwen and GLM achieve $\mathrm{AUC}_w^{macro} = 0.653$ and 0.606 but $\mathrm{SE}_{32} = 0.224$ and 0.046, respectively. These examples motivate the extreme-score analysis in Appendix D, where we examine the maximum-score events that directly determine final selection.

C.3 Scaling and Candidate-Sampling Robustness

The primary experiments use frozen $N = 32$ candidate pools. To examine whether the observed behavior depends on a particular candidate subset or sampling prefix, we repeatedly draw random subpools at $N \in \{2, 4, 8, 16\}$. Table 6 reports the mean selection efficiency over 100 draws at each budget.

Across all benchmark–verifier settings, increasing the candidate budget consistently improves oracle availability, while the corresponding increase in verifier-selected outcomes is limited and inconsistent. The purpose of this analysis is therefore not to establish a universal scaling trend, but to test whether additional candidate availability alone is sufficient to close the verification–selection gap.

This control varies candidate subsets within the same frozen $N = 32$ pools. It therefore isolates candidate-subset variation while keeping the underlying question set and generated candidates fixed. It is complementary to, rather than a substitute for, the independently sampled second MathVista candidate block in Appendix C.4, which tests candidate-sampling replication on the same queries and extends the analysis to $N = 64$.

C.4 Independent Candidate-Block Replication and $N = 64$ Scaling

We next evaluate the same diagnosis under a second independently sampled 32-candidate MathVista block on the same 400 queries. This experiment provides candidate-sampling replication rather than

Table 6: Repeated-subpool robustness. Each entry reports mean selection efficiency over 100 random subpool draws from the frozen $N = 32$ candidate pool. Increasing the candidate budget increases available correct candidates, but does not systematically translate into higher headroom conversion.

Benchmark	Verifier	$N = 2$	$N = 4$	$N = 8$	$N = 16$
MathVista	Qwen	0.301	0.297	0.278	0.269
MathVista	VisualPRM	0.195	0.159	0.126	0.093
MathVista	GLM-4.1V	0.261	0.203	0.158	0.122
MathVerse	Qwen	0.215	0.190	0.173	0.164
MathVerse	VisualPRM	0.157	0.149	0.160	0.142
MathVerse	GLM-4.1V	0.152	0.099	0.053	0.031
MMStar	Qwen	0.187	0.158	0.136	0.140
MMStar	VisualPRM	0.218	0.198	0.198	0.200
MMStar	GLM-4.1V	0.174	0.113	0.085	0.063
MMMU	Qwen	0.353	0.306	0.287	0.243
MMMU	VisualPRM	0.088	0.103	0.111	0.096
MMMU	GLM-4.1V	0.248	0.221	0.177	0.140

question-set replication. Combining the two blocks further extends the available candidate budget from $N = 32$ to $N = 64$.

Table 7 reports the two $N = 32$ blocks and their union. The oracle-availability point estimate increases from 0.8750 for the primary block to 0.8975 for the $N = 64$ union, while selected accuracy changes only marginally: Qwen moves from 0.7675 to 0.7650, and VisualPRM from 0.7300 to 0.7325.

Correspondingly, the conversion of additional candidate availability into selected correct responses remains limited. Qwen selection efficiency changes from 0.3019 to 0.2492, while VisualPRM remains low (0.0583 to 0.0651). These observations extend the limited headroom-conversion pattern to a larger candidate budget without implying that selection efficiency must monotonically decrease as N increases.

Table 7: Independent candidate-block replication and MathVista $N = 64$ scaling. The two 32-candidate blocks are sampled independently on the same 400 queries and share the normalization baseline $A_1 = 0.7210$. Because both verifiers evaluate the same candidate block, $\mathrm{Pass@}N$ is common across verifiers within each block. Numbers in brackets are 95% question-clustered bootstrap confidence intervals.

Pool	Verifier	Acc@N	Pass@N	$\mathbf{Gap_N}$	$\mathbf{SE_N}$
$\mathcal{Y}^{1:32}$	Qwen	0.7675 [0.7275, 0.8100]	0.8750 [0.8425, 0.9025]	0.1075 [0.0775, 0.1375]	0.3019 [0.1625, 0.4526]
	VisualPRM	0.7300 [0.6900, 0.7725]		0.1450 [0.1125, 0.1800]	0.0583 [−0.1215, 0.2235]
$\mathcal{Y}^{33:64}$	Qwen	0.7725 [0.7325, 0.8125]	0.8800 [0.8500, 0.9075]	0.1075 [0.0775, 0.1400]	0.3238 [0.1754, 0.4711]
	VisualPRM	0.7400 [0.7000, 0.7800]		0.1400 [0.1100, 0.1750]	0.1194 [−0.0557, 0.2698]
$\mathcal{Y}^{1:64}$	Qwen	0.7650 [0.7250, 0.8075]	0.8975 [0.8700, 0.9225]	0.1325 [0.1000, 0.1650]	0.2492 [0.1267, 0.3788]
	VisualPRM	0.7325 [0.6900, 0.7775]		0.1650 [0.1300, 0.2025]	0.0651 [−0.0906, 0.2095]

The two independently sampled $N = 32$ candidate blocks also yield comparable deployment quantities. Paired question-level bootstrap gives

$$\Delta\mathrm{Gap}_{32}^{(1:32)-(33:64)} = -0.0003\,[-0.0350, 0.0325], \qquad \Delta\mathrm{SE}_{32}^{(1:32)-(33:64)} = -0.0188\,[-0.2264, 0.1768]$$

for Qwen, and

$$\Delta\mathrm{Gap}_{32}^{(1:32)-(33:64)} = 0.0057\,[-0.0225, 0.0350], \qquad \Delta\mathrm{SE}_{32}^{(1:32)-(33:64)} = -0.0636\,[-0.2264, 0.0874]$$

for VisualPRM. All four paired intervals contain zero, so the two candidate blocks are statistically compatible with the same qualitative limited-conversion pattern under this paired comparison.

Taken together, the second block provides a candidate-sampling replication on the same questions, while the union extends the diagnosis to $N = 64$. The $\mathrm{Pass@32}$ and $\mathrm{Pass@64}$ confidence intervals overlap, so we treat the increase in oracle availability as a point-estimate change rather than a statistically separated effect. Accordingly, this experiment supports persistence of limited headroom conversion at a larger candidate budget, but does not establish a monotonic scaling law.

C.5 Generation-Policy Control

The primary benchmark matrix uses Qwen3-VL-8B-Instruct candidate pools. To examine whether the observed verification–selection mismatch depends only on this candidate distribution, we additionally evaluate a pre-specified GLM-generated MathVista control with 120 queries and 32 candidates per query.

Oracle availability increases from 0.301 for a single candidate to $\mathrm{Pass@32} = 0.675$, while the best verifier-selected accuracy remains approximately 0.300. The resulting $\mathrm{Gap}_{32} = 0.375$ has 95% CI [0.283, 0.467].

This smaller control reproduces the qualitative separation between available correct candidates and verifier-selected outcomes under an additional generation policy. The result provides supporting evidence that the observed phenomenon is not limited to the primary candidate distribution, while not implying a universal generator-independent scaling law.

C.6 Generation- vs. Selection-Limited Failures

To identify which stage limits final performance, we decompose each final error into two mutually exclusive cases: a generation-limited case, where no correct candidate is present in the candidate pool, and a selection-limited case, where at least one correct candidate exists but the verifier selects an incorrect response. Table 8 reports the complete decomposition.

Table 8: Generation- and selection-limited final errors at $N = 32$.

Benchmark	Verifier	Final errors	Generation-limited	Selection-limited
MathVista	Qwen	93	50	43
MathVista	VisualPRM	108	50	58
MathVista	GLM-4.1V	106	50	56
MathVerse	Qwen	231	133	98
MathVerse	VisualPRM	231	133	98
MathVerse	GLM-4.1V	249	133	116
MMStar	Qwen	131	53	78
MMStar	VisualPRM	124	53	71
MMStar	GLM-4.1V	135	53	82
MMMU	Qwen	191	104	87
MMMU	VisualPRM	209	104	105
MMMU	GLM-4.1V	211	104	107

Selection-limited errors constitute a substantial fraction of final failures in every setting, ranging from approximately 42% to 61% of the final errors across the 12 cells. This decomposition shows that increasing candidate availability alone does not fully explain the observed scaling behavior: a considerable portion of failures occur after correct candidates are already present but are not selected by the verifier.

Therefore, the bottleneck is not solely candidate coverage; the final decision rule induced by the verifier remains a critical factor in converting available correct candidates into successful outputs.

C.7 Answer-Frequency Aggregation Baseline

A natural alternative to verifier-based selection is answer-frequency aggregation, which does not require an explicit correctness verifier. We compute the modal parsed answer within each 32-candidate pool, using the smallest candidate index only to break frequency ties. Table 9 reports the resulting performance and headroom conversion.

Table 9: Answer-frequency aggregation at $N = 32$.

Benchmark	A_1	Majority Acc.	Pass@32	Majority Gap	Majority SE_{32}
MathVista-400	0.7210	0.7525	0.875	0.1225	0.204
MathVerse-400	0.3797	0.4100	0.667	0.2575	0.105
MMStar-400	0.6466	0.6550	0.8675	0.2125	0.038
MMMU-400	0.4595	0.4600	0.740	0.2800	0.0017

Answer-frequency aggregation provides a simple selection mechanism based on candidate agreement rather than verifier scores. However, it still converts only a limited fraction of the available headroom across the evaluated benchmarks. On MMMU, for example, majority aggregation is nearly identical to single-sample accuracy despite a 28-point increase in oracle availability. This indicates that the observed scaling limitation is not resolved by a simple self-consistency-style aggregation strategy (Wang et al., 2023).

D Extreme Score Geometry

Verifier-guided deployment is ultimately an extreme-score decision problem: the selected output is determined by the highest verifier score within a candidate set rather than by average pairwise ranking quality. We therefore analyze the extreme margin Δ_q defined in the main paper. A positive margin corresponds to a strict incorrect extreme, zero corresponds to a correct–incorrect tie at the maximum score, and a negative margin corresponds to a correct extreme.

This view separates three qualitatively different verifier behaviors: recovering a correct maximum, producing an ambiguous maximum-score set, and assigning the highest score to an incorrect candidate.

D.1 Cross-Benchmark Extreme-Margin Statistics

Table 10 reports the two extreme-margin events that most directly characterize the selection behavior across all 12 benchmark–verifier cells.

Table 10: Cross-benchmark extreme-score geometry on mixed questions.

Benchmark	Verifier	n_{mixed}	$P(\Delta_q = 0)$	$P(\Delta_q > 0)$
MathVista	Qwen	154	0.662	0.065
MathVista	VisualPRM	154	0.006	0.377
MathVista	GLM-4.1V	154	0.409	0.182
MathVerse	Qwen	205	0.776	0.102
MathVerse	VisualPRM	205	0.000	0.478
MathVerse	GLM-4.1V	205	0.571	0.244
MMStar	Qwen	179	0.743	0.084
MMStar	VisualPRM	179	0.006	0.397
MMStar	GLM-4.1V	179	0.464	0.240
MMMU	Qwen	192	0.698	0.125
MMMU	VisualPRM	192	0.000	0.547
MMMU	GLM-4.1V	192	0.474	0.307

Across benchmarks, the same broad interface-level contrast repeatedly appears: Qwen exhibits a tie-dominated extreme geometry, VisualPRM produces few maximum-score ties but a larger fraction of strict incorrect extremes, and GLM-4.1V occupies an intermediate regime. These results indicate that the selection gap does not arise from a single universal score geometry; instead, different verifier interfaces fail through different extreme-score mechanisms.

The corresponding detailed MathVista maximum-score-set statistics are reported in Table 3 in the main paper. The appendix analysis focuses on cross-benchmark recurrence and the robustness of the conclusions under alternative tie-breaking assumptions.

D.2 TIE-BREAKING SENSITIVITY

Because Qwen and GLM frequently produce tied maximum scores, we test whether the reported selection gap is substantially affected by the deterministic index-based tie-breaking rule. Table 11 compares the deployed index-first accuracy with the expected accuracy under uniform random tie-breaking for all 12 benchmark–verifier cells.

Table 11: Tie-breaking sensitivity at $N = 32$. Fixed-index and random-expected accuracies are generally close, indicating that the particular deterministic index rule is not the primary source of the observed gap.

Benchmark	Verifier	Tied / 400	Index	Random expected
MathVista	Qwen	388	0.7675	0.7584
MathVista	VisualPRM	49	0.7300	0.7288
MathVista	GLM-4.1V	355	0.7350	0.7357
MathVerse	Qwen	364	0.4225	0.4215
MathVerse	VisualPRM	11	0.4225	0.4225
MathVerse	GLM-4.1V	344	0.3775	0.3858
MMStar	Qwen	390	0.6725	0.6788
MMStar	VisualPRM	57	0.6900	0.6887
MMStar	GLM-4.1V	361	0.6625	0.6596
MMMU	Qwen	376	0.5225	0.5190
MMMU	VisualPRM	33	0.4775	0.4770
MMMU	GLM-4.1V	335	0.4725	0.4840

The conclusion is not that ties are harmless. Oracle best/worst resolution within the same maximum-score set spans 0.595–0.850 accuracy for MathVista–Qwen and 0.345–0.680 for MMMU–Qwen. These ranges show that maximum-score sets may contain candidates with substantially different correctness labels. Consequently, randomizing the final tie-breaking rule cannot recover information that is absent from the verifier-defined maximum-score set.

D.3 DISTRIBUTION OF EXTREME MARGINS

The histogram provides a distributional view of the same three extreme-score regimes in the primary MathVista setting and complements the cross-benchmark event rates in Table 10.

Rather than summarizing verifier quality through a single ranking statistic, the extreme-margin distribution directly characterizes the events that govern deployment outcomes: correct extremes ($\Delta_q < 0$), ambiguous maximum-score sets ($\Delta_q = 0$), and strict incorrect extremes ($\Delta_q > 0$). This decomposition provides a decision-level explanation for why similar discrimination statistics can correspond to different selection behavior.

E CONTROLLED INTERVENTION DETAILS

The main results motivate three controlled tests under frozen Qwen3-VL-8B-Instruct candidate representations $h_i \in \mathbb{R}^{4096}$: objective alignment, candidate-set context, and explicit relative comparison. These experiments are designed as diagnostic controls rather than proposed verifier architectures. The generation policy and candidate representations remain fixed, while only lightweight scoring modules are modified to test whether specific missing ingredients can resolve the observed verification–selection mismatch.

Table 12: Protocols for the three controlled intervention families. All methods operate on frozen candidate representations and do not retrain the generation policy. Accuracy differences are meaningful only relative to the matched reference within each row and should not be compared across rows.

Intervention	Representation	Evaluation set	N	Matched reference
SAVR	frozen $f(h)$	MathVista-400	32	Vanilla
SCES	frozen $f(h)$	MathVista-400	32	Vanilla (pointwise)
J2S	frozen $f(h)$	MathVista-200 (val)	16	Judge (pointwise), $\alpha^* = 0.1$

E.1 Testing Whether Selection-aware Objectives Resolve the Mismatch

We first examine whether explicitly optimizing objectives closer to the final selection target can improve deployment outcomes. We train lightweight scoring heads with increasingly selection-oriented objectives. The baseline optimizes independent correctness prediction, while additional objectives emphasize listwise ordering and the correct–incorrect extreme margin.

Table 13: Selection-aware objective comparison on MathVista-400.

Method	AUC_w	AUC_g	Acc_{32}	SE_{32}
Vanilla (L_{base})	0.757	0.863	0.7900	0.448
+List	0.742	0.831	0.7850	0.416
+Top	0.768	0.850	0.7875	0.432
Full SAVR	0.748	0.832	0.7875	0.432

The +Top objective improves within-query ranking from 0.757 to 0.768, showing that the modified objective can affect verifier discrimination. However, this improvement does not translate into better final selection: Full SAVR obtains $\Delta\mathrm{Acc@32} = -0.0025$ relative to Vanilla, with 95% CI $[-0.0225, 0.0175]$.

Thus, under this frozen-representation interface, improving objectives that more explicitly reflect selection structure does not provide clear evidence of improved deployment accuracy.

E.2 Testing Whether Candidate-set Context Resolves the Mismatch

We next examine whether the missing information arises from the absence of candidate-set context. We train a set-conditioned rescoring model that jointly observes candidate representations while preserving the same deployed top-1 selection rule.

The best SCES model reaches $\mathrm{Acc}_{32} = 0.7825$, compared with 0.790 for the matched pointwise reference. The paired difference is

$$\Delta\mathrm{Acc@32} = -0.0075, \qquad 95\%\ \mathrm{CI} = [-0.0275, 0.0100].$$

Under the tested frozen-representation interface, adding candidate-set context alone does not provide clear evidence of improved top-1 selection. This result does not rule out richer joint selectors or representation-level adaptation; it only evaluates whether lightweight post-hoc set conditioning is sufficient to resolve the mismatch.

E.3 Testing Whether Relative Evidence Improves Final Decisions

Finally, we test whether explicit relative comparisons can provide additional decision-relevant information. We train a lightweight pairwise comparator that directly models preferences between candidate pairs.

The comparator achieves $\mathrm{AUC}_{pair} = 0.724$, indicating measurable relative preference information. At $N = 16$, however, the original judge, pairwise replacement, and the best residual combination all obtain $\mathrm{Acc@16} = 0.700$.

Question-clustered paired bootstrap gives

$$\Delta_{\mathrm{replace}} = 0.000\,[-0.060, 0.060],$$

and

$$\Delta_{\mathrm{residual}} = 0.000\,[-0.015, 0.015].$$

The replacement changes individual decisions: it corrects 17 previously wrong decisions while also changing 17 previously correct decisions. Therefore, the pairwise signal is not redundant, but it does not yield a net accuracy improvement under this evaluation setting. The residual combination changes only one decision in each direction and is largely decision-consistent with the original judge.

These results indicate that measurable relative evidence alone is insufficient to establish a reliable deployment improvement under the frozen interface.

E.4 SUMMARY

Table 14: Matched paired accuracy differences for the controlled intervention families. Confidence intervals containing zero are treated as inconclusive, not as evidence of equivalence.

Intervention	Comparison	ΔAcc [95% CI]
SAVR	Full − Vanilla, $N = 32$	-0.0025 [-0.0225, 0.0175]
SCES-v1	SCES − pointwise, $N = 32$	-0.0075 [-0.0275, 0.0100]
J2S-v1	Replace − Judge, $N = 16$	0.000 [-0.060, 0.060]
J2S-v1	Residual − Judge, $N = 16$	0.000 [-0.015, 0.015]

Across the three intervention families, none of the matched comparisons provides clear evidence of improved top-1 accuracy. Since candidate availability is fixed within each comparison, these controls also do not demonstrate a reduction in selection loss.

The conclusion is intentionally narrow: under frozen candidate representations, lightweight post-hoc modifications in objective alignment, candidate-set conditioning, or relative comparison do not establish a reliable improvement in final selection. These results do not exclude richer selectors, jointly adapted representations, or methods that acquire additional evidence beyond the frozen score-space setting.