# MIXTURE OF GRADERS: ADAPTIVE STRATEGY ROUTING FOR LLM-BASED MATHEMATICAL EVALUATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

LLM judges for mathematical reasoning are sensitive to the grading protocol they use. We compare 13 protocols, spanning direct scoring, structured chain-of-thought, and multi-call verification or aggregation, on two new expert-annotated competition-mathematics benchmarks with disjoint problem sources and candidate models: Putnam-AXIOM-Grading (540 problems; 1,080 candidate solutions) and OPC-Grading (500 problems; 1,000 candidate solutions). Across four open-weight judges, no single protocol is best everywhere, and the best choice can flip between benchmarks, a pattern that holds even under controlled prompt paraphrases and Holm-corrected pairwise tests. Building on this, we introduce Mixture of Graders (MoG), a judge-specific router that uses only the problem and reference solution to predict a distribution over grading protocols and combine their scores adaptively for each instance. MoG outperforms the strongest fixed protocol for every open-weight judge on both benchmarks, reducing relative MSE by 6.7 to 27.8% on Putnam-AXIOM-Grading and 3.0 to 13.1% on OPC-Grading. These results show that, for reference-based grading of LLM-generated competition-mathematics solutions, the most useful protocol depends on the instance, and learned routing can exploit this. Code is available at `https://anonymous.4open.science/r/mixture-of-graders-01CC/README.md`.

## 1 INTRODUCTION

Mathematical benchmarks increasingly shape model comparison and development: small score differences can decide which systems are called state-of-the-art and which methods are trained further (Hendrycks et al., 2021; Glazer et al., 2025). But a benchmark score depends on how outputs are graded, not only on the model being evaluated. Grading is especially hard for mathematical reasoning, where partial credit must separate valid intermediate ideas from missing cases, unsupported lemmas, invalid implications, and flawed derivations (Xia et al., 2024; Lightman et al., 2023; Zheng et al., 2025; Uesato et al., 2022). LLM judges are now central to this grading, both for model comparison and for alignment (Zheng et al., 2023; Gu et al., 2024; Chiang et al., 2024; Lambert et al., 2024).

Yet LLM judges are sensitive to prompt format, scoring instructions, and repeated sampling (Liu et al., 2023; Zheng et al., 2023; Shi et al., 2025). They can miss errors in problems they can solve themselves (Zheng et al., 2023), overthink or loop during verification (Khalifa et al., 2025), and assign poorly calibrated partial credit (Mahdavi et al., 2025). Prior work addresses this with many grading protocols, including structured evaluation steps (Liu et al., 2023), verification (Dhuliawala et al., 2024), debate (Chan et al., 2024), and multi-judge aggregation (Verga et al., 2024), typically comparing them by average performance. But if reasonable protocols give different scores to the same solution, benchmark comparisons partly measure the protocol rather than mathematical ability.

We evaluate 13 protocols with four open-weight judges on two expert-annotated benchmarks of LLM-generated competition-mathematics solutions: Putnam-AXIOM-Grading (Gulati et al., 2025) and OPC-Grading, which we build from 500 Open Proof Corpus problems across 13 olympiad and university competitions (Dekoninck et al., 2026). Both use the same partial-credit rubric but differ in problem sources and candidate models. The protocols range from direct scoring and chain-of-thought variants to comparative prompting, evidence-constrained scoring, and multi-call pipelines



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.

(G-Eval, Chain-of-Verification, Debate, and multi-judge aggregation). We also test robustness to controlled prompt paraphrases and apply Holm-corrected pairwise comparisons.

No fixed protocol is best across judges and benchmarks. On Putnam, Comparative is the best fixed protocol for three judges and Bullet-Point for one; on OPC, Debate is best for three judges and Bullet-Point for one. Debate shows the sharpest reversal: it is only mid-pack on Putnam (third- to ninth-best of 13) but the best for three of four judges on OPC. More reasoning or computation also does not reliably help. Unconstrained chain-of-thought (Wei et al., 2022) can raise variance without improving agreement (Haldar & Hockenmaier, 2025b; Khalifa et al., 2025), and in our experiments multi-call verification and aggregation cost more without consistently beating simpler prompts.

If the best protocol shifts across benchmarks, it may also shift across problems within a benchmark. This raises a different question from asking which protocol is best on average: *can a fixed judge be improved by choosing its grading protocol per problem?* To measure the available headroom, we use an in-hindsight oracle that uses expert labels to pick the best protocol for each instance, giving an empirical ceiling on single-protocol selection from our pool. Relative to the best fixed protocol, the oracle reduces MSE by 64.4 to 87.5% on Putnam and 38.2 to 65.7% on OPC. To capture part of this gain without labels, we introduce **Mixture of Graders (MoG)**, a judge-specific router that maps the problem and reference solution to a distribution over protocols and uses it to form an instance-specific weighted grade. The judge model stays fixed and the router never observes the candidate response.

MoG outperforms the strongest fixed protocol for every open-weight judge on both benchmarks, reducing relative MSE by 6.7 to 27.8% on Putnam-AXIOM-Grading and 3.0 to 13.1% on OPC-Grading. Every paired 95% confidence interval excludes zero, even against fixed comparators selected on the evaluation fold. MoG recovers 8.7 to 31.9% of the oracle headroom on Putnam and 7.9 to 20.0% on OPC, where a single protocol already wins most pairwise comparisons. Giving the router the candidate response brings no consistent benefit across the eight benchmark and judge settings.

**Contributions.**

- **Benchmarks with label-reliability analysis.** We introduce Putnam-AXIOM-Grading (540 problems; 1,080 candidate solutions) and OPC-Grading (500 problems; 1,000 candidate solutions), with two candidates per problem in both. Independent double annotation of 500 solutions quantifies label uncertainty.
- **A controlled comparison of grading protocols.** We characterize agreement, stability, and cost across 13 protocols, test robustness to prompt wording and annotator choice, and quantify instance-level oracle headroom.
- **Adaptive protocol routing.** We train and evaluate MoG against the strongest fixed protocols and examine cross-generator transfer and the value of candidate-response features.

## 2 Related Work

**LLM judges and structured evaluation.** LLMs are widely used to evaluate open-ended tasks (Dubois et al., 2025; Liu et al., 2023), but agreement with human preferences can coexist with position and verbosity biases (Zheng et al., 2023). Structured approaches include explicit evaluation steps (Liu et al., 2023), learned calibration of multidimensional LLM outputs in LLM-Rubric (Hashemi et al., 2024), dynamic rubric generation (Du et al., 2024), locked rubrics (Hong et al., 2026), and fine-grained skill-level evaluation (Ye et al., 2024). A separate line trains evaluators directly, including PandaLM (Wang et al., 2023b), Prometheus (Kim et al., 2023), and JudgeLM (Zhu et al., 2023), although their generalization beyond the training distribution can remain limited (Huang et al., 2025).

**Chain-of-thought, verification, and multi-step judging.** CoT and self-consistency can improve problem solving (Wei et al., 2022; Wang et al., 2023a), but their benefits do not transfer uniformly to evaluation. Reasoning-based judgments can exhibit increased variance (Haldar & Hockenmaier, 2025a), generated rationales can be unfaithful (Lanham et al., 2023; Turpin et al., 2023), and CoT gains can be limited when evaluation criteria are explicit (Yamauchi et al., 2025). Self-correction

without external feedback can also be unreliable (Huang et al., 2023). Other approaches separate drafting from verification (Dhuliawala et al., 2024), train step-level verifiers (Lightman et al., 2024), or aggregate perspectives through debate and panels (Chan et al., 2024; Verga et al., 2024). ThinkPRM demonstrates benefits from trained generative verification with long CoT (Khalifa et al., 2025).

**Adaptive routing and evaluation protocols.** FrugalGPT, Hybrid LLM, RouteLLM, and BEST-Route study quality–cost tradeoffs through model selection, cascades, and test-time compute allocation (Chen et al., 2024; Ding et al., 2024; Ong et al., 2024; Ding et al., 2025), while RouterBench provides a framework for evaluating routing systems (Hu et al., 2024). Route-To-Reason jointly selects models and reasoning strategies (Pan et al., 2026), Symbolic-MoE studies expert selection for reasoning (Chen et al., 2025), and SEAR connects structured evaluation signals to gateway routing (Zhang et al., 2026b). Adaptive evaluation is more directly addressed by Auto-Prompt Ensemble, which learns auxiliary evaluation dimensions from failures and uses confidence-gated aggregation (Li et al., 2025a), and LLM Jury-on-Demand, which predicts judge reliability for instance-specific selection and score weighting (Li et al., 2025b). RACER selects between reasoning and non-reasoning judgments under budget and distribution-shift constraints, including inference modes of the same base model (Zhang et al., 2026a). Our focus is learning weights over heterogeneous grading protocols instantiated with a fixed judge. MoG conditions these weights only on the problem and reference solution, not the candidate response, and minimizes MSE against expert partial-credit labels; stability and cost are analyzed separately.

**Mathematical partial-credit grading.** Mathematical evaluation complements final-answer benchmarks such as MATH and GSM8K (Hendrycks et al., 2021; Cobbe et al., 2021) with process supervision, reasoning-quality assessment, and error localization (Lightman et al., 2023; Xia et al., 2024; Zheng et al., 2025). RefGrader addresses partial-credit calibration through reference-derived, problem-specific rubrics and multi-step grading workflows, and also studies averaging predictions across workflows (Mahdavi et al., 2025). Putnam-AXIOM (Gulati et al., 2025) and the Open Proof Corpus (Dekoninck et al., 2026) provide resources for competition-level reasoning and proof evaluation.

# 3 METHODS

## 3.1 TASK AND DATA

We study reference-based partial-credit grading of LLM-generated competition-mathematics solutions. Each grading instance consists of a problem statement, a reference solution, a model-generated candidate solution, and an expert partial-credit label under a shared five-component rubric. We use two benchmarks with the same grading task but different problem sources and candidate-model populations: Putnam-AXIOM-Grading and OPC-Grading.

**Putnam-AXIOM-Grading.** Putnam-AXIOM-Grading extends Putnam-AXIOM (Gulati et al., 2025) into a partial-credit grading benchmark. It contains 540 problems, covering all 522 problems in Putnam-AXIOM together with additional problems from the 2024 and 2025 Putnam competitions. Each problem is paired with two model-generated candidate solutions, from QWEN2.5-72B-MATH-INSTRUCT and MISTRAL-LARGE-INSTRUCT-2411, yielding 1,080 grading instances. Each record contains the problem statement, a reference solution derived from the official or benchmark solution, a frozen candidate solution, and a partial-credit label.

**OPC-Grading.** To evaluate the same grading task on a distinct mathematical distribution, we construct OPC-Grading from the Open Proof Corpus (Dekoninck et al., 2026). We select 500 problems spanning 13 olympiad and university competitions, including the Balkan Mathematical Olympiad Shortlist, European Girls' Mathematical Olympiad, IMC, and IMO Shortlist. Each problem is paired with two fully expert-graded candidate solutions, from O3 and QWEN3-235B-A22B, yielding 1,000 grading instances. OPC-Grading uses the same annotation rubric as Putnam-AXIOM-Grading while changing both the source of the mathematical problems and the models generating the candidate solutions.

Table 1: The 13 grader protocols by design axis. Multi-call protocols are in **bold**; all others use a single call. Full prompts in Appendix S.1.

| Axis | Protocols |
|---|---|
| Reasoning amount | No-CoT (direct score); Brief-CoT (1–2 sentences); Full-CoT (free-form, per component) (Wei et al., 2022) |
| Reasoning structure | Structured-CoT (summarize → check → decide); Bullet-Point (terse per-criterion) |
| Evidence anchoring | Comparative (candidate vs. reference) (Zheng et al., 2023); Quote-Forcing (claims must quote the output) (Turpin et al., 2023) |
| Revision & verification | Self-Critique (Huang et al., 2023); **G-Eval** (generated checklist) (Liu et al., 2023); **CoVe** (verification questions) (Dhuliawala et al., 2024); **Overseer** (audit, up to 5 retries) (Lightman et al., 2024) |
| Multi-agent | **Debate** (pro/con + arbiter) (Chan et al., 2024); **Multi-Judge** (3 personas + chair) (Verga et al., 2024) |

**Rubric and output format.** Both benchmarks use the five-component analytic rubric in Figure 2. The rubric scores logical coherence, faithfulness to the task, methodological alignment, intermediate correctness, and error awareness/self-correction, with each component receiving a score in $\{0, 0.5, 1\}$ for a total score between 0 and 5. The same rubric is provided to human annotators and LLM graders. LLM graders return component scores and the total score in structured JSON enclosed by `<json>` tags, following common LLM-as-a-judge practice (Zheng et al., 2023; Liu et al., 2023).

**Annotation reliability.** We independently double-grade 500 solutions, split evenly across the two benchmarks. Quadratic-weighted agreement is $\kappa = 0.857$ on Putnam-AXIOM-Grading and $\kappa = 0.815$ on OPC-Grading, with $\kappa = 0.842$ overall and MAE 0.540 on the 0–5 scale. This indicates strong ordinal agreement despite some variation in exact partial-credit allocation. Appendix O tests robustness to annotator choice, audits disagreements with a held-out frontier evaluator, and gives a representative disagreement example.

### 3.2 Candidate grader pool

We evaluate a pool $\mathcal{S}$ of 13 grading protocols spanning single-call prompts and multi-call pipelines. Every protocol receives the same problem, reference solution, candidate solution, and rubric; only the procedure used to reason about and aggregate the judgment changes. The protocols span five design axes—reasoning amount, reasoning structure, evidence anchoring, revision/verification, and multi-agent aggregation—and are summarized in Table 1. Full prompts appear in Appendix S.1 and are shared across benchmarks. The pool is designed to capture distinct grader behaviors rather than exhaust all possible protocols; Appendix L analyzes its complementarity and redundancy.

### 3.3 Fixed-strategy evaluation protocol

We evaluate each candidate grader as a fixed protocol applied uniformly across instances. We use four open-weight judge models—Qwen2.5-32B-Instruct, DeepSeek-R1-Distill-Qwen-32B, Qwen2.5-72B-Instruct, and Meta-Llama-3.1-70B-Instruct—served with batched inference via `vLLM` (Kwon et al., 2023). For every benchmark, candidate-solution set, judge model, and grader protocol, we run five independent trials with seeds 43–47, temperature 0.7, top_p = 1, maximum output length 2048, and stop sequences at JSON closing tags. We use stochastic decoding intentionally to measure protocol instability under repeated judge sampling (Tan et al., 2024; Thakur et al., 2024; Haldar & Hockenmaier, 2025a). Overseer-guided retry permits up to five retries, with overseer responses capped at 800 tokens.

For each strategy–instance–run cell, we extract the final numeric score by reading the structured JSON field (`score` or `score_total`); if that fails, a deterministic regex fallback recovers an explicitly printed final score. We retain the extracted score, repeated-run scores, token usage, and extraction status for every cell.

We evaluate fixed strategies for two metrics, agreement and stability. *Agreement* is the mean squared error (MSE) against the human reference label, using each strategy's mean extracted score across repeated runs. *Stability* is the per-instance variance of total scores across repeated runs, averaged over instances; lower variance indicates more consistent grading under sampling (Haldar & Hockenmaier, 2025a; Thakur et al., 2024). We also track end-to-end token usage. For statistical comparison, we report 95% problem-clustered bootstrap confidence intervals for MSE and run variance. Within each benchmark–judge panel, all strategy comparisons use shared problem-level bootstrap samples to preserve pairing. We apply Holm–Bonferroni correction separately by panel and metric, with Benjamini–Hochberg as a sensitivity analysis, and report a Holm statistical top set consisting of the numerical winner plus strategies not significantly worse than it (Appendix M).

### 3.4 Adaptive routing with Mixture of Graders

The fixed-strategy sweep yields a judge-specific strategy–instance score matrix with entries $S_{q,s} = |\mathcal{R}_{q,s}|^{-1} \sum_r \hat{y}_{q,s,r}$, the mean extracted score across valid repeated runs; we also retain repeated-run variance $\sigma^2_{q,s}$ and mean token cost $C_{q,s}$. As a diagnostic upper bound, we define an in-hindsight oracle that selects the strategy maximizing $U_q(s) = -((S_{q,s} - y_q)/M_q)^2 - \lambda\sigma^2_{q,s}/M_q^2 - \gamma C_{q,s}/\bar{C}$, where $y_q$ is the human reference label, $M_q$ the benchmark maximum, and $\bar{C}$ normalizes token cost. The oracle is not deployable (it uses $y_q$), but it measures the headroom available to adaptive choice. MoG is trained from the full strategy-score matrix rather than by imitating hard oracle labels.

**Router formulation and objective.** MoG maps $x_q = (\text{problem}_q, \text{reference}_q)$ to a distribution $p_\theta(\cdot \mid x_q)$ over the 13 strategies and predicts the expected score $\tilde{y}_q = \sum_s p_\theta(s \mid x_q)\, S_{q,s}$. We train it to minimize $N^{-1} \sum_q (\tilde{y}_q - y_q)^2$ on $[0, 1]$-normalized scores, so the router shifts mass toward strategies that move $\tilde{y}_q$ toward the human label rather than imitating one hard oracle choice among near-ties. The candidate solution and student identity are deliberately excluded, so MoG is a problem/reference-conditioned policy that cannot learn artifacts of a particular student's output style.

**Architecture and evaluation.** The router is a ModernBERT-large encoder (inputs truncated to 1024 tokens) with mean pooling and a 13-way linear head, trained separately per judge; hyperparameters are in Appendix B. On Putnam we train one router per judge and student response set with 5-fold cross-validation over all 540 problems, pooling the two response sets ($Q_{\text{eff}} = 1{,}080$); on OPC we use grouped 5-fold cross-validation by problem, stratified by competition, pooling both candidate families ($Q_{\text{eff}} = 1{,}000$). In each fold, the fixed baseline is the strategy with the lowest MSE *on the held-out fold itself*, which favors the baseline. Significance uses a paired bootstrap (10,000 resamples) over per-instance squared-error differences. For OPC, $S_{q,s}$ averages only trials that parsed for all 13 strategies (Appendix C), and, as on Putnam, we report all MSEs on the native 0–5 scale.

## 4 Results and Analysis

### 4.1 Fixed strategies are useful but not sufficient

The fixed-strategy sweep shows that grader protocol materially affects both MSE and stability, and that the best protocol depends on the judge and, even more, on the benchmark. On Putnam-AXIOM-Grading (Table 2), Comparative prompting is the strongest practical default: it is single-call, ranks first or second among fixed strategies for every judge, and achieves the best fixed-strategy MSE for three of four judges. But it is not globally optimal: Bullet-Point is best for Llama-3.1-70B-Instruct, and No-CoT trails Comparative by only .015 for both Qwen judges. On OPC-Grading (Table 3), the ranking shifts substantially. Debate is the numerical winner for three of four judges and the sole member of the Holm top set for both Qwen judges, while Bullet-Point wins for Llama; Comparative is not in any OPC top set. Debate's reversal is the clearest example of distribution dependence: on Putnam it ranks between third and ninth of 13 fixed protocols and has the highest run-to-run variance for three of four open-weight judges.

Adding reasoning or pipeline complexity does not solve this variation. Multi-call strategies sometimes win on MSE, but they do not consistently dominate simpler single-call graders and often cost

Table 2: MSE and variance ($\overline{\sigma^2}$) on Putnam-AXIOM-Grading, averaged across the Qwen2.5-72B and Mistral-Large-2411 student models (native 0–5 scale). Both metrics are lower-is-better. Marking is based on each metric *within each model*: best is **bold**, second-best is underlined, third-best is marked with $^\dagger$. OPC-Grading results are in Table 3.

| | | **Qwen2.5-32B** | | **Qwen2.5-72B** | | **DeepSeek-R1-32B** | | **Llama-3.1-70B** | |
|---|---|---|---|---|---|---|---|---|---|
| **Type** | **Strategy** | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ |
| *Ours* | MoG | **.630** | .417 | **.586** | .333 | **.789** | .445 | **.737** | .305 |
| *Baseline* | No-CoT | .839$^\dagger$ | **.152** | .827$^\dagger$ | .127 | 1.314 | .318 | 1.143 | .214 |
| *Single-Stage* | Brief-CoT | .963 | .350 | 1.377 | .230 | 1.093 | .300 | 1.131 | .348 |
| | Full-CoT | 1.285 | .284 | .987 | .139$^\dagger$ | 1.185 | .288 | 1.333 | .229$^\dagger$ |
| | Structured-CoT | 1.185 | .436 | 2.250 | .232 | 1.043 | .303 | 1.387 | .314 |
| | Bullet-Point | .922 | .212$^\dagger$ | 1.011 | .157 | 1.025 | .269 | .790 | .269 |
| | Comparative | .824 | .246 | .812 | .170 | .850 | .286 | .995$^\dagger$ | .255 |
| | Quote-Forcing | 1.181 | .467 | 1.100 | .245 | 1.037 | .297 | 1.001 | .356 |
| | Self-Critique | 1.036 | .322 | 1.267 | .266 | 1.405 | .282 | 1.142 | .379 |
| *Multi-Step* | G-Eval | 1.572 | .913 | 1.748 | .735 | .955$^\dagger$ | .559 | 1.886 | .569 |
| | CoVe | 1.043 | .308 | .886 | .179 | 1.090 | .324 | 1.048 | .288 |
| | Multi-Judge | 1.107 | .188 | .975 | **.090** | 1.056 | **.212** | 1.234 | **.164** |
| | Debate | 1.076 | 1.076 | 1.143 | .586 | 1.006 | .708 | 1.149 | .813 |
| | Overseer | 1.318 | .286 | 1.020 | .140 | 1.284 | .274$^\dagger$ | 1.404 | .233 |

substantially more: G-Eval, CoVe, Debate, Multi-Judge, and Overseer-guided Retry use roughly 2–7$\times$ more tokens than single-call prompts on Putnam-AXIOM-Grading (Table 11). Their gains are also uneven: Debate wins most OPC panels but is only mid-pack on Putnam, while G-Eval has the highest Putnam MSE for two of four open-weight judges. Conversely, low-variance strategies such as No-CoT or Multi-Judge often fail to minimize MSE. Fixed protocols therefore expose a three-way tradeoff between accuracy, stability, and cost: they are useful baselines, but no single fixed strategy resolves this tradeoff across judge models and benchmark distributions.

Among single-call variants, the *structure* of elicited reasoning matters more than its presence: terse, rubric-aligned formats (Comparative, Bullet-Point) are consistently competitive, while unconstrained variants (Brief-, Full-, and Structured-CoT, Self-Critique) generally raise variance without lowering MSE.

### 4.2 Fixed-strategy conclusions are statistically and prompt robust

Numerical rankings alone can overstate small differences between grader protocols. Across Putnam-AXIOM-Grading and OPC-Grading, Holm-corrected paired tests resolve 480/624 MSE comparisons (76.9%). The statistical top set is a singleton in 4/8 open-weight benchmark–judge panels and contains only two or three strategies in the remaining four. The numerical winner varies across settings: Comparative wins three panels (the two Qwen judges and DeepSeek on Putnam), Debate wins three panels (the two Qwen judges and DeepSeek on OPC), and Bullet-Point wins two panels (Llama on both benchmarks). For three of four judges, the Putnam and OPC top sets are disjoint. Thus, individual protocols can be robustly best within a panel, while no fixed protocol dominates across benchmark and judge distributions. Full panel-level top sets appear in Appendix M.

We also test whether these conclusions are artifacts of wording. For each strategy we evaluate the original prompt and three operation-preserving paraphrases (rubric, structure, stage order, and output schema frozen) with all four judges on 300-candidate subsets of each benchmark, five runs per cell. Between-strategy variation in per-run squared error is 14.77$\times$ within-strategy prompt variation (95% CI [11.83, 15.25]), with the ratio exceeding one in all eight benchmark–judge conditions. Rankings are largely preserved (mean Spearman 0.894; 88.4% of pairwise orderings preserved): exact winner identity is unchanged in 16/24 comparisons (66.7%), while the original winner remains in the paraphrased top three in 23/24 and every paraphrased winner was already in the original top three. Wording can swap close competitors, but it does not explain the broader protocol-level effects (Appendix N).

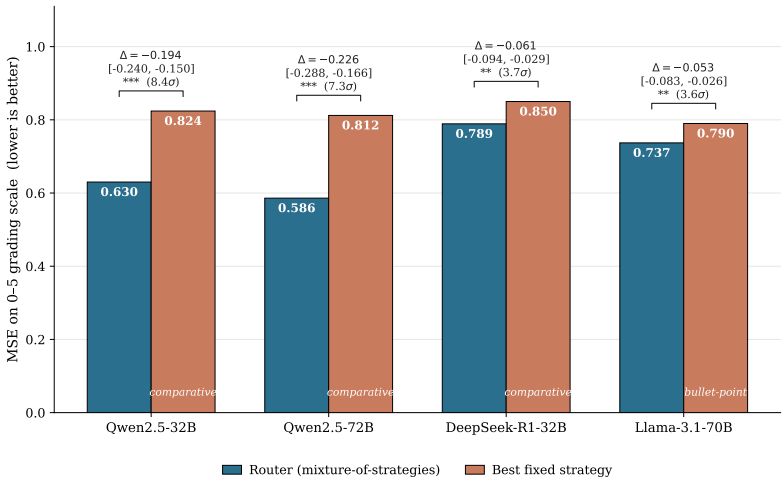



Figure 1: MoG vs. the best fixed strategy on Putnam-AXIOM-Grading (pooled 5-fold CV, $Q_{\text{eff}} = 1{,}080$ per judge; MSE on the 0–5 scale). The best fixed strategy is each judge's lowest-MSE protocol in Table 2. Brackets give $\Delta = \text{MSE}_{\text{MoG}} - \text{MSE}_{\text{fixed}}$ with 95% paired bootstrap CIs and, in parentheses, $\Delta$ in bootstrap standard errors ($\sigma$); all exclude zero by $\geq 3.6\sigma$. $^{**}p < 10^{-3}$; $^{***}p < 10^{-4}$.

### 4.3 Mixture of Graders improves over fixed strategy selection

The fixed-strategy sweep leaves substantial adaptive headroom. Under the primary MSE-aligned oracle setting ($\lambda = 0, \gamma = 0$), the oracle-selected strategy differs from the validation-selected best fixed strategy in 83.9% of held-out Putnam model-instance cases (Table 8) and 62.9% of OPC cases. This translates into large oracle gains: oracle selection reduces MSE by 64.4%–87.5% on Putnam-AXIOM-Grading relative to the best fixed strategy (Table 9) and 38.2%–65.7% on OPC. Sensitivity sweeps over the stability penalty $\lambda$ and token-cost penalty $\gamma$ show that this headroom persists across objective choices, and oracle selections do not collapse to a single fixed protocol (Appendix K).

MoG converts this headroom into deployable gains. On Putnam-AXIOM-Grading, it achieves the lowest MSE for every open-weight judge (Table 2). It reduces MSE by 6.7 to 27.8% relative to the strongest fixed strategy and recovers 8.7 to 31.9% of the gap between that strategy and the MSE-only oracle ($\lambda = 0$). All paired bootstrap CIs exclude zero by at least 3.6 bootstrap standard errors (Figure 1), even though the fixed baseline is selected on the evaluation fold.

The fixed-strategy picture on OPC is sharper than on Putnam. Holm correction resolves 64–72 of 78 pairwise comparisons per judge (276/312 overall, versus 204/312 on Putnam), and the top set is a singleton for three of four judges ((Table 16)). Yet the protocol that wins is different: Debate for both Qwen judges and DeepSeek, Bullet-Point for Llama.

The full strategy rankings in Table 3 show how judge-specific these preferences are. Debate is the best fixed strategy for Qwen2.5-32B, Qwen2.5-72B, and DeepSeek-R1-32B but only eighth of 13 for Llama-3.1-70B. Bullet-Point, Llama's best strategy, ranks second for DeepSeek, sixth for Qwen2.5-32B, and ninth for Qwen2.5-72B. Comparative, the strongest default on Putnam, never ranks first on OPC (second, fourth, third, and seventh). No-CoT, second-best for both Qwen judges on Putnam, ranks between second and tenth.

MoG nevertheless beats the best fixed protocol for every judge, by 3.0–13.1% relative MSE, with every 95% paired bootstrap CI excluding zero. It also beats all 13 fixed strategies and the uniform mixture (by 27–59%), and closes 7.9–20.0% of the gap to the oracle. The routers are concentrated (mean maximum routing probability 0.71–0.87; Table 7), so they are best read as the judge's best fixed protocol adjusted problem by problem. Gains are smaller than on Putnam, as expected when one protocol already resolves most pairwise comparisons, but they hold on a disjoint problem source and candidate population.

Table 3: All 13 fixed strategies on **OPC-Grading** ($Q_{\text{eff}} = 1{,}000$ per judge; native 0–5 scale). MSE and $\overline{\sigma^2}$ are as in Table 2; both are lower-is-better. For MoG, $\overline{\sigma^2}$ holds routing probabilities fixed and so reflects judge sampling noise only. MoG beats every fixed strategy and the uniform mixture on MSE for every judge (paired bootstrap, $p \leq .0005$; CIs in Table 4). Best per metric and judge, including MoG, is **bold**, second underlined, third $^{\dagger}$.

| | | Qwen2.5-32B | | Qwen2.5-72B | | DeepSeek-R1-32B | | Llama-3.1-70B | |
|---|---|---|---|---|---|---|---|---|---|
| **Type** | **Strategy** | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ | MSE | $\overline{\sigma^2}$ |
| *Ours* | MoG | **1.330** | .746 | **.960** | .608 | **2.020** | .567 | **.905** | **.267** |
| *Baseline* | No-CoT | 2.348 | **.096** | 2.110$^{\dagger}$ | .187$^{\dagger}$ | 3.278 | .230$^{\dagger}$ | 1.858 | .311 |
| *Single-Stage* | Brief-CoT | 2.010 | .336 | 3.038 | .205 | 2.638 | .275 | 1.430 | .562 |
| | Full-CoT | 2.685 | .182$^{\dagger}$ | 2.420 | .297 | 3.398 | .235 | 2.165 | .367 |
| | Structured-CoT | 3.365 | .350 | 4.663 | **.177** | 2.718 | .353 | 3.130 | .403 |
| | Bullet-Point | 2.378 | .301 | 2.943 | .188 | 2.243$^{\dagger}$ | .264 | .965 | .436 |
| | Comparative | 1.928$^{\dagger}$ | .216 | 2.353 | .184 | 2.370 | .324 | 1.890 | .384 |
| | Quote-Forcing | 3.178 | .345 | 2.928 | .312 | 3.273 | .243 | 1.063$^{\dagger}$ | .553 |
| | Self-Critique | 2.448 | .240 | 3.078 | .227 | 2.858 | .287 | 2.303 | .451 |
| *Multi-Step* | G-Eval | 3.720 | .740 | 3.718 | .738 | 2.558 | .443 | 1.888 | 1.383 |
| | CoVe | 2.313 | .292 | 2.388 | .303 | 3.155 | .373 | 1.650 | .533 |
| | Multi-Judge | 2.608 | .142 | 2.350 | .226 | 3.473 | **.188** | 2.035 | .335$^{\dagger}$ |
| | Debate | 1.398 | 1.093 | 1.105 | .947 | 2.083 | .855 | 1.933 | 1.274 |
| | Overseer | 2.725 | .193 | 2.438 | .297 | 3.480 | .223 | 2.188 | .373 |

Because the strategy–score matrix must be built on expert-labeled training problems, we also ask how many such problems MoG needs. We retrain the QWEN2.5-72B-INSTRUCT routers on random subsets of each fold's training problems (400 on OPC, 432 on Putnam), keeping the held-out folds fixed. MoG beats the best fixed protocol at every size tested, down to 10 training problems per fold, and with 100 problems it already achieves 95% (OPC) and 94% (Putnam) of the full-data MSE reduction. Thus, for this judge, the main training sets are substantially larger than the amount of labeled data needed to recover most of MoG's observed gain (Appendix R, Figure 7).

### 4.4 ROUTING GENERALIZES ACROSS STUDENT-MODEL OUTPUTS

A natural concern with response-blind routing is that the learned policy may depend on idiosyncrasies of the student-model response distribution. To test this, we train MoG on Putnam-AXIOM-Grading instances generated by a QWEN2.5-72B-MATH-INSTRUCT student model and evaluate it without retraining on the parallel MISTRAL-LARGE-INSTRUCT-2411 response set, using the same 540 problems and the same QWEN2.5-32B-INSTRUCT judge. Because the router never sees the candidate solution, this directly tests whether problem/reference-conditioned strategy preferences transfer across student-output distributions.

Under the expected-router deployment mode,

$$\tilde{y}_q = \sum_s p_\theta(s \mid x_q) S_{q,s},$$

the transferred router achieves MSE $0.538$ on Mistral responses, outperforming the uniform mixture over strategies $(0.623)$ and all fixed strategies in point estimate, including Comparative prompting $(0.589)$, the strongest fixed default (Table 10). It also significantly outperforms the uniform mixture ($p < 0.05$), and recovers a median $11.5\%$ of the oracle headroom.

### 4.5 CANDIDATE-RESPONSE FEATURES DO NOT CONSISTENTLY IMPROVE ROUTING

A natural hypothesis is that the best protocol depends on the candidate solution's error type, so a response-aware router should outperform our response-blind design. We test this directly by adding the candidate solution to the router input while holding folds, targets, hyperparameters, seeds, and context-window size fixed. Across all four judges on both Putnam and OPC, seven of eight comparisons are statistical ties; the sole significant difference is a small OPC improvement for DeepSeek

($\Delta\text{MSE} = -0.018$ on the 0–5 scale). There is no consistent direction across judges or benchmarks, and the 95% CIs bound plausible effects to within about 0.08 MSE on the 0–5 scale. The response-aware router does change its routing distribution across candidates, so the null is not explained by ignoring the extra input. In our setting, response-blindness therefore does not impose a measurable accuracy penalty and retains the practical advantage of transfer across candidate-generator distributions. Full results appear in Appendix P.

## 5 LIMITATIONS

We evaluate four open-weight judges; rankings and routing gains may differ for other architectures, including proprietary models and reasoning-specialized models beyond DeepSeek-R1. Our prompt-robustness study tests four controlled instantiations per strategy, but it still does not exhaust the space of possible implementations. The benchmarks are reference-based, LLM-generated competition-mathematics solutions; our results do not establish transfer to naturally occurring human student work, reference-free grading, or other domains.

MoG trades additional computation for improved grading agreement. The headline results use the expected-mixture estimator and therefore require the constituent protocol scores at evaluation time; they should be interpreted as an accuracy result rather than a compute-efficiency result. A single-strategy routing variant could instead select and execute only one grading protocol per instance, although its accuracy–cost trade-off is not evaluated in our headline experiments. MoG also requires a judge-specific strategy–score matrix for training which is expensive though our training-size ablation shows that most of the observed gain can be recovered using substantially fewer labeled and fully swept training problems (Appendices Q and R).

Finally, expert partial-credit scores retain some calibration uncertainty. Expanded double annotation, annotator-sensitivity analysis, and a strong-model disagreement audit show that the broad fixed-strategy conclusions are stable, while exact scores and marginal comparisons can still vary with annotation (Appendix O).

## 6 CONCLUSION

We study LLM-as-a-judge for difficult mathematical grading and ask whether a single fixed grader protocol is the right abstraction. Across two competition-mathematics benchmarks with disjoint problem sources and candidate models, the identity of the best fixed protocol varies by judge and, most sharply, by benchmark: Comparative is a strong default on Putnam-AXIOM-Grading, while Debate is best for most judges on OPC-Grading. This is not primarily a prompt-wording artifact: controlled paraphrasing shows that strategy choice explains far more per-run squared-error variation than wording choice, and Holm-corrected comparisons resolve most pairwise protocol differences. A complementary strategy pool exposes substantial instance-level oracle headroom, while expanded double annotation shows that the comparative protocol conclusions remain stable despite meaningful uncertainty in exact partial-credit labels.

Motivated by this gap, we introduce **Mixture of Graders**, a MODERNBERT-LARGE router over evaluation protocols. MoG improves over the strongest fixed strategy for every open-weight judge on both Putnam-AXIOM-Grading and OPC-Grading, transfers across candidate generators, and does not consistently benefit from seeing the candidate response itself. The resulting claim is deliberately scoped: for reference-based grading of LLM-generated competition-mathematics solutions, grader-protocol utility is instance-conditional, and learned routing can exploit that variation. The contribution is therefore not a novel encoder architecture but a shift in evaluation design—from searching for one universal judging prompt to treating the grading protocol itself as a routable decision.

## REFERENCES

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25cc3adf8c85f7c70989cb8a97a691a7-Paper-Conference.pdf.

Justin Chih-Yao Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, and Mohit Bansal. Symbolic mixture-of-experts: Adaptive skill-based routing for heterogeneous reasoning, 2025. URL https://arxiv.org/abs/2503.05641.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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

Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, Kaloyan Tsvetkov, Nikolay Drenchev, Lazar Todorov, et al. The open proof corpus: A large-scale study of llm-generated mathematical proofs. In *International Conference on Learning Representations*, volume 2026, pp. 22214–22244, 2026.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruehle, Laks V. S. Lakshmanan, and Ahmed Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *ICLR 2024*, February 2024. URL https://www.microsoft.com/en-us/research/publication/hybrid-llm-cost-efficient-and-quality-aware-query-routing/.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Hipolito Garcia, Menglin Xia, L. Lakshmanan, Qingyun Wu, and Victor Ruehle. Best-route: Adaptive llm routing with test-time optimal compute. *ArXiv*, abs/2506.22716, June 2025. URL https://www.microsoft.com/en-us/research/publication/best-route-adaptive-llm-routing-with-test-time-optimal-compute/.

Hanyu Du, Parvez Rashid, and Qian Jia. Interactive rubric generator for instructor's assessment using prompt engineering and large language models. In *Proceedings of the IEEE Frontiers in Education Conference*. IEEE, 2024. URL https://ieeexplore.ieee.org/abstract/document/10893448.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators, 2025. URL https://arxiv.org/abs/2404.04475.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2025. URL https://arxiv.org/abs/2411.04872.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024. URL https://arxiv.org/abs/2411.15594.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno Dumont, Elyas Obbad, and Sanmi Koyejo. Putnam-axiom: A functional and static benchmark for measuring higher level mathematical reasoning in llms, 2025. URL https://arxiv.org/abs/2508.08292.

Ishita Haldar and Julia Hockenmaier. Rating roulette: Do language models give consistent ratings to different parts of a document? In *Proceedings of the 2025 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*, 2025a. doi: 10.1145/3701716.3715530. URL https://dl.acm.org/doi/10.1145/3701716.3715530.

Rajarshi Haldar and Julia Hockenmaier. Rating roulette: Self-inconsistency in llm-as-a-judge frameworks, 2025b. URL https://arxiv.org/abs/2510.27106.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13806–13834. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.745. URL http://dx.doi.org/10.18653/v1/2024.acl-long.745.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS 2021 Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Yufei Hong, Hao Yao, Biao Shen, Wenxin Xu, and Hao Wei. Rulers: Locked rubrics and evidence-anchored scoring for robust llm evaluation. *arXiv preprint arXiv:2601.08654*, 2026. URL https://arxiv.org/abs/2601.08654.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system, 2024. URL https://arxiv.org/abs/2403.12031.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5880–5895, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.306. URL https://aclanthology.org/2025.findings-acl.306/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023. URL https://arxiv.org/abs/2310.01798.

Muhammad Khalifa et al. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*, 2023. URL https://arxiv.org/abs/2310.08491.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP)*, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Jiajie Li, Huayi Zhang, Peng Lin, Jinjun Xiong, and Wei Xu. Auto-prompt ensemble for llm judge, 2025a. URL https://arxiv.org/abs/2510.06538.

Xiaochuan Li, Ke Wang, Girija Gouda, Shubham Choudhary, Yaqun Wang, Linwei Hu, Joel Vaughan, and Freddy Lecue. Who judges the judge? LLM jury-on-demand: Building trustworthy LLM evaluation systems, 2025b. URL https://arxiv.org/abs/2512.01786.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.20050.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2511–2522, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Hamed Mahdavi et al. RefGrader: Automated grading of mathematical competition proofs using agentic workflows. *arXiv preprint arXiv:2510.09021*, 2025.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2024. URL https://arxiv.org/abs/2406.18665.

Zhihong Pan, Kai Zhang, Yuze Zhao, and Yupeng Han. Adaptive model and strategy routing for cost-efficient llm services. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 5568–5578, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792556. URL https://doi.org/10.1145/3774904.3792556.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge, 2025. URL https://arxiv.org/abs/2406.07791.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. *arXiv preprint arXiv:2410.12784*, 2024.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. *arXiv preprint arXiv:2406.12624*, 2024. URL https://arxiv.org/abs/2406.12624.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

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

Patrick Verga, Sebastian Hofstätter, Nazneen Rajani, Mohit Iyyer, et al. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024. URL https://arxiv.org/abs/2404.18796.

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 *International Conference on Learning Representations (ICLR)*, 2023a. URL https://arxiv.org/abs/2203.11171.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shi-Bo Zhang, and Yue Zhang. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *ICLR 2024*, June 2023b. URL https://www.microsoft.com/en-us/research/publication/pandalm-an-automatic-evaluation-benchmark-for-llm-instruction-tuning-optimization/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2201.11903.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*, 2024. URL https://arxiv.org/abs/2404.05692.

Yusuke Yamauchi, Taro Yano, and Masafumi Oyamada. An empirical study of LLM-as-a-judge: How design choices impact evaluation reliability. *arXiv preprint arXiv:2506.13639*, 2025.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. Flask: Fine-grained language model evaluation based on alignment skill sets. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 55361–55414, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/f41b4a6b202adcd8e150a9d4f124d8f6-Paper-Conference.pdf.

Wenbo Zhang, Lijinghua Zhang, Liner Xiang, and Hengrui Cai. Reasoning is not free: Robust adaptive cost-efficient routing for LLM-as-a-judge, 2026a. URL https://arxiv.org/abs/2605.10805.

Zecheng Zhang, Han Zheng, and Yue Xu. Sear: Schema-based evaluation and routing for llm gateways, 2026b. URL https://arxiv.org/abs/2603.26728.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning, 2025. URL https://arxiv.org/abs/2412.06559.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. 2023.

## A AI USAGE DISCLOSURE

Generative AI tools were used to assist with literature discovery, implement and debug experimental code, and edit portions of the manuscript for clarity and presentation. All AI-assisted code was reviewed and tested by the authors, and all experimental results, citations, claims, and manuscript text were independently checked by the authors. The authors take full responsibility for the final content of this work.

## B ROUTER TRAINING DETAILS

The router is a MODERNBERT-LARGE encoder whose input is the concatenated problem statement and reference solution; the candidate solution is never included. Token embeddings are mean-pooled and passed to a 13-way linear head, followed by a softmax over strategies. Inputs are truncated to 1,024 tokens on OPC-Grading and Putnam-AXIOM-Grading. We train with AdamW (learning rate $2 \times 10^{-5}$ for the encoder and $10^{-3}$ for the head), batch size 16, two epochs of head-only warmup, and six epochs of joint fine-tuning, using router seed 0. The objective is the mixture-MSE loss of Section 3.4 on $[0, 1]$-normalized scores. We train a separate router for each judge, because the strategy–score matrix, oracle headroom, and best fixed strategy are all judge-specific.

## C OPC-GRADING ROUTING DIAGNOSTICS

This appendix reports diagnostics for MoG on OPC-Grading for all four open-weight judges. All numbers come from a single router-training seed. Table 4 gives confidence intervals for every gap in Table 3.

Table 4: 95% paired bootstrap confidence intervals (10,000 resamples) for the gap MoG − comparator on OPC-Grading (native 0–5 scale), complementing Table 3. Negative values favor MoG.

| **Comparator** | **Qwen2.5-32B** | **Qwen2.5-72B** | **DeepSeek-R1-32B** | **Llama-3.1-70B** |
|---|---|---|---|---|
| No-CoT | [−1.150, −.888] | [−1.298, −1.000] | [−1.388, −1.130] | [−1.073, −.830] |
| Brief-CoT | [−.783, −.583] | [−2.263, −1.885] | [−.720, −.518] | [−.613, −.440] |
| Full-CoT | [−1.500, −1.213] | [−1.628, −1.300] | [−1.510, −1.245] | [−1.390, −1.135] |
| Structured-CoT | [−2.218, −1.853] | [−3.978, −3.430] | [−.808, −.590] | [−2.418, −2.030] |
| Bullet-Point | [−1.175, −.918] | [−2.170, −1.793] | [−.310, −.135] | [−.093, −.028] |
| Comparative | [−.705, −.485] | [−1.553, −1.233] | [−.455, −.248] | [−1.105, −.865] |
| Quote-Forcing | [−2.020, −1.680] | [−2.155, −1.780] | [−1.383, −1.130] | [−.210, −.103] |
| Self-Critique | [−1.243, −.988] | [−2.315, −1.920] | [−.940, −.738] | [−1.545, −1.245] |
| G-Eval | [−2.595, −2.190] | [−2.998, −2.513] | [−.648, −.428] | [−1.115, −.853] |
| CoVe | [−1.115, −.850] | [−1.593, −1.268] | [−1.263, −1.010] | [−.845, −.645] |
| Multi-Judge | [−1.408, −1.140] | [−1.553, −1.228] | [−1.588, −1.315] | [−1.253, −1.008] |
| Debate | [−.108, −.028] | [−.215, −.080] | [−.100, −.025] | [−1.155, −.898] |
| Overseer | [−1.538, −1.253] | [−1.643, −1.308] | [−1.595, −1.328] | [−1.413, −1.153] |
| Uniform mixture | [−1.083, −.863] | [−1.545, −1.255] | [−.828, −.650] | [−.723, −.555] |
| Oracle | [+.498, +.630] | [+.523, +.645] | [+.663, +.810] | [+.500, +.610] |

**Data quality.** All 13 strategies have full instance coverage for every judge, and no routing units are dropped. Because $S_{q,s}$ averages only over trial indices that parsed for every strategy, lower parse rates cost more cells (Table 5). Qwen2.5-32B is the cleanest judge. DeepSeek-R1-32B, the only reasoning model, and Qwen2.5-72B lose the most cells, so their score estimates are the noisiest.

Table 5: OPC-Grading preprocessing by judge and candidate family. "Discarded" counts valid trial cells removed so that all 13 strategies share the same set of trial indices.

| Judge | Candidates | Parse rate | Cells kept | Discarded | % discarded |
|---|---|---|---|---|---|
| Qwen2.5-32B | O3 | 99.99% | 32,435 | 60 | 0.2% |
| | QWEN3-235B | 99.99% | 32,487 | 12 | 0.0% |
| Qwen2.5-72B | O3 | 98.14% | 25,467 | 6,490 | 20.3% |
| | QWEN3-235B | 98.14% | 23,816 | 8,016 | 25.2% |
| DeepSeek-R1-32B | O3 | 99.09% | 28,639 | 3,496 | 10.9% |
| | QWEN3-235B | 99.09% | 29,692 | 2,580 | 8.0% |
| Llama-3.1-70B | O3 | 99.87% | 31,941 | 516 | 1.6% |
| | QWEN3-235B | 99.87% | 31,928 | 528 | 1.6% |

**Per-response-set results.** For three of four judges, MoG MSE is similar across the two candidate families (Table 6). DeepSeek-R1-32B is the exception: its two families differ by .985, more than three times the largest spread for the other judges, so its pooled MSE averages two quite different regimes. Across folds, MoG beats the fold's best fixed strategy in 10/10 folds for Qwen2.5-72B (per-fold gaps $-.255$ to $-.020$) and 9/10 for Llama-3.1-70B ($-.173$ to $.000$). Individual folds ($Q = 100$) are too small to support claims on their own, but the consistent sign shows the pooled result is not carried by one fold.

Table 6: MoG MSE on OPC-Grading by candidate family (native 0–5 scale).

| Judge | O3 | QWEN3-235B-A22B | Pooled |
|---|---|---|---|
| Qwen2.5-32B | 1.370 | 1.290 | 1.330 |
| Qwen2.5-72B | .988 | .933 | .960 |
| DeepSeek-R1-32B | 1.528 | 2.513 | 2.020 |
| Llama-3.1-70B | 1.053 | .760 | .905 |

**What the router learns.** All four routers concentrate their probability mass: the mean entropy of every routing distribution is far below the uniform distribution's 2.565 nats, and the argmax falls on only 2–6 of the 13 strategies (Table 7). The routers therefore behave like the judge's best fixed strategy adjusted per problem, rather than spreading mass evenly over the pool. DeepSeek-R1-32B's router is the most concentrated by argmax count (two strategies), consistent with its gain being the smallest of the four.

Table 7: Concentration of the learned OPC routing distributions.

| Judge | Mean max $p$ | Mean entropy (nats) | Distinct argmax |
|---|---|---|---|
| Qwen2.5-32B | 0.873 | 0.386 | 6 / 13 |
| Qwen2.5-72B | 0.797 | 0.661 | 4 / 13 |
| DeepSeek-R1-32B | 0.756 | 0.537 | 2 / 13 |
| Llama-3.1-70B | 0.705 | 0.762 | 4 / 13 |

**Input truncation.** At the 1,024-token router limit, 28.6% of OPC router inputs are truncated (median length 723 tokens, 90th percentile 1,780, maximum 3,590), so on roughly a quarter of problems the encoder does not see the full reference solution. The reported gains are therefore obtained despite this truncation; a longer context window may improve them.

# D ADAPTIVE STRATEGY HEADROOM RESULTS

Table 8: How often the validation-selected best fixed strategy matches or differs from the oracle-selected strategy under the primary MSE-aligned setting $\lambda = 0, \gamma = 0$.

| Benchmark | Model | Best fixed | Match | Different |
|---|---|---|---|---|
| Putnam-AXIOM-Grading | DeepSeek-R1-32B | Comparative | 15.5% | 84.5% |
| | Llama-3.1-70B | Bullet-point | 16.9% | 83.1% |
| | Qwen2.5-32B | Comparative | 16.5% | 83.5% |
| | Qwen2.5-72B | Comparative | 15.7% | 84.3% |
| OPC-Grading | DeepSeek-R1-32B | Debate | 31.2% | 68.8% |
| | Llama-3.1-70B | Bullet-point | 25.4% | 74.6% |
| | Qwen2.5-32B | Debate | 40.5% | 59.5% |
| | Qwen2.5-72B | Debate | 51.2% | 48.8% |

Table 9: MSE-aligned strategy-selection headroom on held-out instances. $\Delta U$ reports the utility improvement of the oracle selector over the validation-selected best fixed strategy under the primary setting $\lambda = 0, \gamma = 0$. MSE is reported on the native 0–5 scale. Higher $\Delta U$ and Pearson are better; lower MSE is better. The best fixed strategy is selected on each fold's validation data, so its pooled MSE can differ slightly from the lowest pooled MSE in Table 2 (Qwen2.5-72B: .829 vs. .812 for Comparative, because a different protocol is selected on some folds).

| Benchmark | Model | Best fixed | $\Delta U$ | Oracle MSE | Fixed MSE | MSE red. | Oracle $\rho$ | Fixed $\rho$ |
|---|---|---|---|---|---|---|---|---|
| Putnam | DeepSeek-R1-32B | Comparative | .022 | .303 | .850 | 64.4% | .925 | .789 |
| | Llama-3.1-70B | Bullet-point | .024 | .181 | .790 | 77.1% | .946 | .753 |
| | Qwen2.5-32B | Comparative | .027 | .157 | .824 | 81.0% | .954 | .772 |
| | Qwen2.5-72B | Comparative | .029 | .104 | .829 | 87.5% | .969 | .781 |
| OPC | DeepSeek-R1-32B | Debate | .032 | 1.287 | 2.083 | 38.2% | .822 | .717 |
| | Llama-3.1-70B | Bullet-point | .025 | .352 | .965 | 63.6% | .928 | .757 |
| | Qwen2.5-32B | Debate | .025 | .770 | 1.398 | 44.9% | .865 | .670 |
| | Qwen2.5-72B | Debate | .029 | .379 | 1.105 | 65.7% | .921 | .677 |

# E MoG Transfer Ablation

Table 10: Router transfer to a held-out student model on Putnam-AXIOM-Grading. A router trained on Qwen2.5-72B-Math-Instruct student responses is evaluated, without retraining, on Mistral student responses graded by the same Qwen-2.5-32B judge. We report MSE of the routed score against human grades on the native 0–5 Putnam total-score scale, with $95\%$ paired bootstrap confidence intervals over $n = 540$ problems ($B = 10{,}000$ replicates). The bottom block reports both router deployment modes alongside the uniform mixture and the oracle upper bound, plus the paired contrast $\Delta$MSE against the strongest fixed default, Comparative ($p$ is one-sided). $^{\dagger}$ marks rows where Router (expected) significantly outperforms the strategy on MSE (paired bootstrap, $p < 0.05$). The best non-oracle MSE is **bolded**.

| Type | Strategy | MSE ↓ | 95% CI | ΔMSE vs. Comp. ($p$) |
|---|---|---|---|---|
| Baseline | No-CoT$^{\dagger}$ | 0.7004 | [0.6155, 0.7900] | — |
| Single-Turn | Brief-CoT$^{\dagger}$ | 0.8266 | [0.7282, 0.9351] | — |
| | Full-CoT$^{\dagger}$ | 1.2334 | [1.1099, 1.3568] | — |
| | Structured-CoT$^{\dagger}$ | 0.9810 | [0.8467, 1.1207] | — |
| | Bullet-Point$^{\dagger}$ | 0.6845 | [0.5928, 0.7872] | — |
| | *Comparative* | *0.5886* | [0.5099, 0.6696] | — |
| | Quote-Forcing$^{\dagger}$ | 0.9659 | [0.8430, 1.0974] | — |
| | Self-Critique$^{\dagger}$ | 0.7301 | [0.6336, 0.8324] | — |
| Multi-Step | G-Eval$^{\dagger}$ | 1.4807 | [1.3115, 1.6635] | — |
| | CoVe$^{\dagger}$ | 1.0202 | [0.9180, 1.1248] | — |
| | Multi-Judge$^{\dagger}$ | 0.9583 | [0.8543, 1.0673] | — |
| | Debate$^{\dagger}$ | 0.9436 | [0.8231, 1.0710] | — |
| | Overseer$^{\dagger}$ | 1.2566 | [1.1349, 1.3865] | — |
| Routed | Uniform mixture | 0.6228 | [0.5444, 0.7066] | — |
| | Router (expected) | **0.5383** | [0.4713, 0.6105] | $-0.0502$ ($p$=0.059) |
| Upper bound | Oracle | 0.1505 | [0.1213, 0.1830] | — |

*Headroom captured by Router (expected)*: median $11.5\%$ of the gap between the best fixed strategy and the oracle (95% CI $[-3.2\%,\ +23.7\%]$); $P(\text{Router} < \text{best fixed}) = 0.941$ across paired bootstrap replicates.

# F TOKEN CALCULATION FORMULAS

**Token Cost Summary**

*Per call:*

$$T_{\text{call}} = T_{\text{in}}(\text{call}) + T_{\text{out}}(\text{call}) \tag{1}$$

*Per strategy (summed over problems $i$):*

$$T_{\text{strategy}} = \sum_i T_{\text{strategy}}(i) \tag{2}$$

*Per-problem token costs by strategy:*

**Single-turn** (`no_cot`, `brief_cot`, `structured_cot`, `full_cot`, `self_critique`, `bullet_point`, `comparative`, `quote_forcing`):

$$T = T_{\text{in}}(\text{prompt}) + T_{\text{out}}(\text{output}) \tag{3}$$

**G-Eval** (two-stage checklist then scoring):

$$T = \left[T_{\text{in}}(\text{step}_1) + T_{\text{out}}(\text{checklist})\right] + \left[T_{\text{in}}(\text{step}_2(\text{checklist})) + T_{\text{out}}(\text{output})\right] \tag{4}$$

**Debate** (pro, con, then arbiter):

$$\begin{aligned} T = &\left[T_{\text{in}}(\text{pro}) + T_{\text{out}}(\text{pro_arg})\right] \\ &+ \left[T_{\text{in}}(\text{con}) + T_{\text{out}}(\text{con_arg})\right] \\ &+ \left[T_{\text{in}}(\text{arbiter}(\text{pro_arg},\ \text{con_arg})) + T_{\text{out}}(\text{output})\right] \end{aligned} \tag{5}$$

**CoVe** (draft, verify, answer, finalise):

$$\begin{aligned} T = &\left[T_{\text{in}}(\text{draft}) + T_{\text{out}}(\text{draft})\right] \\ &+ \left[T_{\text{in}}(\text{verify}(\text{draft})) + T_{\text{out}}(\text{questions})\right] \\ &+ \left[T_{\text{in}}(\text{answer}(\text{questions})) + T_{\text{out}}(\text{answers})\right] \\ &+ \left[T_{\text{in}}(\text{final}(\text{draft},\ \text{Q\&A})) + T_{\text{out}}(\text{output})\right] \end{aligned} \tag{6}$$

**Multi-Judge** (three persona judges then chair):

$$T = \sum_{j=1}^{3} \left[T_{\text{in}}(\text{persona}_j) + T_{\text{out}}(\text{judgment}_j)\right] + T_{\text{in}}(\text{chair}(j_1, j_2, j_3)) + T_{\text{out}}(\text{output}) \tag{7}$$

**Overseer-Retry** (summed over all iterations $k$):

$$T = \sum_k \left( \left[T_{\text{in}}(\text{judge}_k) + T_{\text{out}}(\text{judge_out}_k)\right] + \left[T_{\text{in}}(\text{overseer}_k) + T_{\text{out}}(\text{overseer_out}_k)\right] \right) \tag{8}$$

where $k$ iterates over all entries in `overseer_checks`.

## G TOKEN USAGE ANALYSIS

Table 11: Total token usage of judge strategies across models on **Putnam-AXIOM-Grading**. We report total tokens consumed per strategy per model. Highest tokens per model are **bolded**; second-highest are underlined. Strategies are grouped by type: baseline, single-turn variants, and multi-step pipelines.

| Type | Strategy | DeepSeek-R1-32B Tokens | Qwen2.5-32B Tokens | Qwen2.5-72B Tokens | Llama-3.1-70B Tokens |
|---|---|---|---|---|---|
| *Baseline* | No-CoT | 866,526 | 850,780 | 851,797 | 844,746 |
| *Single-Turn* | Brief-CoT | 856,775 | 855,903 | 855,932 | 849,895 |
| | Full-CoT | 1,037,450 | 1,006,790 | 1,009,740 | 1,001,598 |
| | Structured-CoT | 988,835 | 933,031 | 932,164 | 928,397 |
| | Bullet-Point | 886,734 | 890,603 | 890,824 | 885,537 |
| | Comparative | 958,600 | 895,427 | 894,719 | 888,687 |
| | Quote-Forcing | 983,522 | 891,478 | 896,288 | 887,367 |
| | Self-Critique | 1,103,194 | 916,705 | 918,074 | 914,197 |
| *Multi-Step* | G-Eval | 2,363,471 | 1,960,149 | 1,905,160 | 2,085,316 |
| | CoVe | 5,125,495 | 3,774,706 | 3,799,769 | 3,860,194 |
| | Multi-Judge | **6,070,101** | **4,804,855** | **4,937,384** | 4,825,306 |
| | Debate | 4,744,275 | 3,532,677 | 4,190,834 | 3,695,543 |
| | Overseer | 4,545,332 | 2,884,088 | 3,499,382 | **5,407,469** |

## H COMPARATIVE PROMPTING ABLATIONS

Given comparative prompting's strong performance on Putnam-AXIOM-Grading (Table 2), we conduct targeted ablations to identify its active ingredients. We test five variants against the original (A0) on all four open-source judges on Putnam-AXIOM-Grading (Table 12).

**Template ablations** We test three modifications to the comparative template: (A1) **Comparative no-compare** uses the same four-step template but replaces "compare to ground truth" with "evaluate the solution's correctness," while keeping the ground truth in context. This tests whether explicitly directing attention to the reference is the active ingredient. (A2) **Difference-only** strips the template to a minimal prompt: "List the differences between the model output and the ground truth, then score." This tests whether the bare act of difference-finding suffices without the template structure. (A4) **Comparative shuffled** uses the same four components as A0 but in a randomized order (error analysis → scoring rationale → approach comparison → differences). This tests whether the specific progressive-narrowing ordering matters.

**Attention-direction ablations** We test two variants that redirect the judge's focus: (A5) **Quote-forcing reference** requires the judge to cite direct quotes from the *ground truth* rather than the student solution, testing whether anchoring attention specifically to the reference drives performance. (A3) **Reference checklist** auto-generates 3–5 binary Yes/No verification questions from the ground truth, then has the judge answer them before scoring—similar to G-Eval but with checklist items derived from the reference rather than freely generated.

**Results** The ablation results (Table 12) fail to support any single-ingredient explanation for comparative prompting's effectiveness:

- **Attention anchoring is not the driver.** If explicitly directing attention to the reference were the key mechanism, A5 (quote-forcing reference) should approach A0. Instead, A5 underperforms A0 for every judge. Conversely, A1 performs competitively despite *not* explicitly referencing the ground truth in its instructions, suggesting that the reference need only be *present in context*, not explicitly called out.
- **Template structure is not necessary.** A2 (difference-only), which removes all template structure, performs comparably to A0, exceeding it for three of four judges.
- **Task framing ("compare" vs. "evaluate") does not explain the gap.** A1 and A2 differ precisely on this dimension—A1 frames the task as evaluation while A2 frames it as difference-finding—yet perform similarly, ruling out cognitive framing as the primary mechanism.
- **Component ordering has limited effect.** A4 (shuffled order) matches or exceeds A0 for every judge, suggesting the progressive-narrowing structure of the original template is not critical.
- **Structured reference decomposition hurts.** A3 (reference checklist) has the highest variance for every judge and the lowest correlation for three of four (A5 is lower for Llama-3.1-70B), mirroring G-Eval's poor showing in the main experiments—suggesting that decomposing the reference into binary checks destroys holistic reasoning.

**Interpretation** The pattern that emerges is that *all high-performing variants share a common property*: they allow the judge to holistically reason about both solutions with minimal output constraints. The strategies that degrade performance—A3 (reference checklist) and A5 (quote-forcing reference)—are precisely those that impose rigid intermediate output requirements, forcing the judge to fragment its reasoning into constrained artifacts (binary answers, quoted evidence) before scoring. We term this active ingredient *evaluation-focused holistic reasoning*: the prompt keeps the judge's attention on the substantive task of assessment rather than diverting it to mechanical subtasks.

Table 12: Ablation results for comparative prompting on Putnam-AXIOM-Grading. We report Pearson correlation with human grades ($\rho$, higher is better) and mean within-condition variance ($\sigma^2$, lower is better) across five independent trials. Best results per model–metric are **bolded**; second-best are underlined. Strategies are ordered by conceptual grouping: original comparative, template ablations, and attention-direction ablations. The ablation uses its own set of five trials, so A0's variances differ slightly from Comparative's in Table 2.

| | Pearson ($\rho$) ↑ | | | | Variance ($\sigma^2$) ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| Strategy | Qwen-32B | Qwen-72B | DS-R1-32B | Llama-70B | Qwen-32B | Qwen-72B | DS-R1-32B | Llama-70B |
| BASELINE | | | | | | | | |
| Comparative (A0) | .715 | .733 | .725 | .677 | .239 | .179 | .316 | .293 |
| TEMPLATE ABLATIONS | | | | | | | | |
| Comp. no compare (A1) | .716 | .714 | .728 | .681 | **.219** | .184 | .293 | **.263** |
| Difference only (A2) | .718 | .736 | .704 | .688 | .254 | **.158** | .322 | .261 |
| Comp. shuffled (A4) | **.732** | **.761** | **.740** | **.689** | .251 | .165 | **.279** | .297 |
| ATTENTION-DIRECTION ABLATIONS | | | | | | | | |
| Quote-force ref. (A5) | .690 | .718 | .684 | .654 | .361 | .185 | .334 | .368 |
| Ref. checklist (A3) | .678 | .614 | .660 | .661 | .419 | .390 | .427 | .408 |

# I GRADING RUBRIC

**Reasoning-quality rubric (MODEL_OUTPUT vs. GROUND_TRUTH_SOLUTION).**

*Task:* Evaluate the `MODEL_OUTPUT`'s *reasoning quality* against the `GROUND_TRUTH_SOLUTION`. Focus *only* on reasoning quality (not writing style).

**Criteria (each scored 0–1; 0.5 allowed):**

1. **Logical Coherence (0–1)**: Are steps locally valid and globally consistent?
2. **Faithfulness to Task (0–1)**: Does the reasoning address the question (no hallucinated subgoals or irrelevant paths)?
3. **Methodological Alignment (0–1)**: Is the approach compatible with (or reasonably equivalent to) the canonical method in the ground truth? If different, is it still valid?
4. **Intermediate Correctness (0–1)**: Are key intermediate results (derivations, lemmas, sub-calculations) correct?
5. **Error Awareness / Self-Correction (0–1)**: Does the reasoning notice and repair mistakes or check edge cases? (If model does not make any nontrivial mistakes, do not penalize here).

**Scoring:** Final score is the sum across the five criteria (0–5; half-points allowed).

Figure 2: Rubric used by annotators and judge LLMs for both Putnam-AXIOM-Grading and OPC-Grading.

# J DEEPSEEK R1 ANALYSIS

DeepSeek-R1-Distill-Qwen-32B departs from the other judges in how it responds to direct scoring. On Putnam-AXIOM-Grading, No-CoT has the second-highest MSE of any single-call strategy for this judge (1.314, behind Self-Critique at 1.405; Table 2), whereas it is the second-best fixed strategy for both Qwen judges. Here we present a fine-grained analysis of *how* the Full-CoT and No-CoT conditions differ for this model, based on a comparison of all 540 Putnam-AXIOM-Grading outputs from a single representative trial.

**DeepSeek-R1 cannot suppress reasoning.** The most fundamental observation is that DeepSeek-R1-Distill-Qwen-32B generates extensive chain-of-thought reasoning in *both* conditions. Even under the No-CoT prompt—which instructs the model to output only a JSON score with no reasoning—536 of 540 outputs (99.4%) contain a `</think>` token demarcating an internal reasoning scratchpad. These scratchpads average 2,016 characters, only 10% shorter than the Full-CoT scratchpads (2,243 characters). After the `</think>` token, the No-CoT outputs contain only the JSON score block ($\sim$230 characters). For reasoning-optimized models, "direct scoring" is therefore not just less effective but behaviorally unnatural—the model *cannot help but deliberate*.

The critical difference lies not in whether the model reasons, but in what happens *after* the scratchpad. Under Full-CoT, 465 of 540 outputs (86.2%) exhibit a **two-phase structure**: a free-form scratchpad (enclosed within the model's internal think tokens), followed by a structured `<THINKING>` summary that distills the scratchpad's observations into per-criterion assessments, followed by the JSON score. Under No-CoT, the model proceeds directly from its scratchpad to the JSON score, with no intermediate consolidation step.

**The distillation step corrects leniency on surface-level criteria.** This architectural difference has measurable consequences. Comparing per-component score means across the two conditions reveals an asymmetric pattern: No-CoT assigns higher scores on *methodological alignment* ($+0.10$ shift, from 0.55 to 0.65) and *logical coherence* ($+0.05$ shift), while assigning *lower* scores on *error awareness* ($-0.12$ shift). The first two criteria are precisely those assessable from a surface reading of the student solution: "Does the method seem reasonable?" and "Do the steps flow logically?" The No-CoT scratchpad tends to answer these affirmatively based on the student's internal consistency, without weighing the ground truth.

The error awareness shift is a striking counterpoint: No-CoT is actually *harsher* on this criterion (mean 0.16 vs. 0.28 under Full-CoT), but this harshness appears nearly random—the Pearson correlation between No-CoT's error awareness scores and human total scores is only $r = 0.39$, compared to $r = 0.64$ under Full-CoT, the largest per-component gap we observe ($\Delta r = +0.25$). Substantial gaps also appear for logical coherence ($\Delta r = +0.08$) and methodological alignment ($\Delta r = +0.07$).

**No-CoT compresses scores toward the midrange.** The net effect of these per-component shifts is a compression of the total score distribution. Under No-CoT, 65.3% of scores fall in the 1.5–4.0 range, compared to 48.7% under Full-CoT and 48.4% for human grades. Correspondingly, No-CoT produces fewer extreme scores: only 15.9% of scores are $\leq 1$ (vs. 21.2% for Full-CoT, 36.9% for humans) and only 17.8% are $\geq 4.5$ (vs. 29.2% for Full-CoT). The reduced spread (standard deviation 1.37 vs. 1.56 for Full-CoT, 1.43 for humans) directly reduces Pearson correlation: on the 448 instances where both conditions parse successfully, Full-CoT achieves $r = 0.69$ versus $r = 0.64$ for No-CoT.

**Qualitative mechanism: noticing errors without penalizing them.** The quantitative patterns are explained by a consistent qualitative mechanism visible in individual outputs (Figures 3–5). In the No-CoT condition, the model's scratchpad frequently *identifies* discrepancies with the ground truth but then *rationalizes them away* in the same breath, defaulting to a lenient assessment of the student solution's internal logic. Three illustrative failure modes emerge:

1. **Internal-consistency bias** (Figure 3). The scratchpad notes that the student derives $I_k = I_{5-k}$ instead of the correct $I_{4-k}$, but then writes: "However, the steps are locally valid and globally consistent *within its own reasoning*, so I'll give a 1." The judge evaluates the solution as a standalone argument rather than against the reference.

2. **Premise-blindness** (Figure 4). The student's internal mathematics is correct (a quadratic is properly solved), but the entire argument rests on a false premise contradicted by the ground truth. The No-CoT scratchpad never confronts this, writing: "The calculations are accurate. So, Intermediate Correctness is 1." The Full-CoT scratchpad immediately cross-references: "doesn't align with the ground truth, which shows all $u$'s are zero."
3. **Failure to propagate errors across criteria** (Figure 5). The No-CoT scratchpad notices that the student's answer is wrong but confines the penalty to a single criterion (Intermediate Correctness), leaving Logical Coherence, Faithfulness, and Methodological Alignment at full marks. The Full-CoT `<THINKING>` summary propagates the error backward, recognizing that an incorrect conclusion undermines the coherence and faithfulness of the reasoning that led to it.

In each case, the Full-CoT condition avoids the failure because its two-phase structure—scratchpad followed by `<THINKING>` summary—forces the model to *consolidate* its observations into rubric-calibrated judgments. The scratchpad serves as exploratory analysis; the summary serves as deliberate assessment. Without this distillation step, the model's initial impressions pass through to the final score unchecked.

**Implications.** DeepSeek-R1 deliberates regardless of the prompt, but the *quality of the transition from deliberation to scoring* depends on whether the prompt elicits an explicit output-side reasoning structure. The Full-CoT prompt does not make the model think more—the scratchpads are similar in length and content—but it provides a scaffold for converting messy, exploratory reasoning into calibrated judgments. For practitioners using reasoning-specialized models as judges, this suggests that the key design choice is not whether to allow CoT (the model will reason anyway) but whether to require a structured output format that forces distillation of that reasoning before scoring.

**Problem (Putnam #227)**

Let $p(x) = 2 + 4x + 3x^2 + 5x^3 + 3x^4 + 4x^5 + 2x^6$. For $k$ with $0 < k < 5$, define $I_k = \int_0^\infty \frac{x^k}{p(x)}\,dx$. For which $k$ is $I_k$ smallest?

**Ground Truth Solution (abridged)**

Substitute $x = 1/t$ to show $I_k = I_{4-k}$. Then $I_k = (I_k + I_{4-k})/2 = \int_0^\infty \frac{(x^k+x^{4-k})/2}{p(x)}\,dx > \int_0^\infty \frac{x^2}{p(x)}\,dx = I_2$ by AM–GM, since $(x^k + x^{4-k})/2 \geq x^2$. Answer: $k = \boxed{2}$.

**Student Solution (abridged)**

Notes $p(x) = x^6 p(1/x)$ (correct). Claims $I_k = I_{5-k}$ (should be $I_{4-k}$). Approximates $p(x) \approx 23$ near $x = 1$, argues minimum at $k = 5/2$. Assumes $k$ must be integer (not stated in problem). Checks $k = 2$ vs. $k = 3$, concludes $k = 2$. Gets the right answer through flawed reasoning.

**No-CoT Judge (score: 4.5 / 5)**

"The model's reasoning seems to flow well [...] I notice that the model assumes symmetry around 5, not 4 as in the ground truth. However, the steps are locally valid and globally consistent within its own reasoning, so I'll give a 1."

"The model correctly identifies $p(x)$ as symmetric, but the substitution in the ground truth relates $I_k$ to $I_{4-k}$, not $I_{5-k}$. The model's symmetry is slightly off. However, the main conclusion $k$=2 is correct, so intermediate steps are mostly correct. Score: 1."

"Both methods are valid but different. The model's approach is reasonable, so alignment is present. Score: 1."

LC: 1 | FT: 1 | MA: 1 | IC: 1 | EA: 0.5 | **Total: 4.5**

**Full-CoT Judge (score: 1.0 / 5)**

*Scratchpad:*
"Wait, that's a discrepancy. The ground truth uses $4-k$, but the model says $5-k$. Hmm, that might be an error."

"Wait, but why does the model assume $k$ must be an integer? The problem states $0 < k < 5$, but $k$ doesn't have to be an integer [...] That could be a mistake."

⟨*THINKING*⟩ *summary:*
"Methodological Alignment (0): The approach differs from the ground truth [...] leading to invalid steps."
"Intermediate Correctness (0): Key steps, like the substitution and integer assumption, are incorrect, undermining the reasoning."

LC: 0.5 | FT: 0.5 | MA: 0 | IC: 0 | EA: 0 | **Total: 1.0**

Figure 3: **"Locally valid within its own reasoning" (Human: 1.0).** Both judges notice that the student derives $I_k = I_{5-k}$ instead of the correct $I_{4-k}$. The No-CoT judge excuses this: because the steps are internally consistent, it awards full marks on four criteria. The Full-CoT judge's scratchpad flags the same discrepancy ("*Wait, that's a discrepancy*") and the unjustified integer assumption, then the structured <THINKING> summary consolidates these into zero-score penalties on methodology and correctness—matching the human grade exactly. The key failure mode: **without a distillation step, No-CoT evaluates the student solution as a standalone argument rather than against the reference standard.**

**Problem (Putnam #92)**

Suppose $u_0, u_1, u_2, \ldots$ is a sequence of real numbers with $u_n = \sum_{k=1}^{\infty} u_{n+k}^2$ for all $n \geq 0$. If $\sum u_n$ converges, find the maximum value of $u_k$ for all $k$.

**Ground Truth (abridged)**

Shows $\{u_n\}$ is non-increasing and non-negative. Chooses $p$ with $\sum_{n>p} u_n < 1$, then derives $u_p \leq u_p \cdot \sum_{n>p} u_n < u_p$, a contradiction unless $u_p = 0$. Hence all $u$'s are zero. Answer: $\boxed{0}$.

**Student Solution (abridged)**

Sets $n = 0$: $u_0 = \sum_{k=1}^{\infty} u_k^2$. Sets $n = 1$: $u_1 = \sum_{k=2}^{\infty} u_k^2$. Subtracts to get $u_0 = u_1^2 + u_1$, a quadratic. Solves: $u_1 = (-1 + \sqrt{1 + 4u_0})/2$. Claims $u_0 = 1/4$ is possible, giving $u_1 = 1/2$. Answer: $\boxed{1/2}$ (incorrect—the correct answer is 0).

**No-CoT Judge (score: 4.5 / 5)**

"The model correctly derives the quadratic equation and solves it, finding $u_1 = 1/2$ when $u_0 = 1/4$. The calculations are accurate. So, Intermediate Correctness is 1."

"The model correctly identifies that the maximum occurs at $u_1$ when $u_0$ is $1/4$, leading to $u_1 = 1/2$. This directly addresses the task without irrelevant paths. So, Faithfulness is 1."

"The ground truth uses induction and an inequality approach, while the model uses a quadratic equation [...] Both methods aim to find the maximum value, and the model's approach is valid. So, Methodological Alignment is 1."

LC: 1 | FT: 1 | MA: 1 | IC: 1 | EA: 0.5 | **Total: 4.5**

**Full-CoT Judge (score: 1.0 / 5)**

*Scratchpad:*
"However, the reasoning doesn't align with the ground truth, which shows all $u$'s are zero. The model's approach is inconsistent because it arrives at a non-zero maximum, contradicting the correct solution. So, Logical Coherence is 0."

"The model's intermediate steps, like solving the quadratic, are mathematically correct. However, the premise that $u_0$ can be $1/4$ is flawed because the ground truth shows all terms must be zero."

*⟨THINKING⟩ summary:*
"Logical Coherence (0): The argument is inconsistent. It derives a maximum of $1/2$ but fails to consider that all terms must be zero."

LC: 0 | FT: 0.5 | MA: 0 | IC: 0.5 | EA: 0 | **Total: 1.0**

Figure 4: **The internal-consistency trap (Human: 1.5).** The student solution is internally valid mathematics—the quadratic really does yield $u_1 = 1/2$—but the entire argument rests on the false premise that $u_0 = 1/4$ is achievable. No-CoT evaluates the solution as a standalone argument ("the calculations are accurate"; "the model's approach is valid") without confronting the ground truth's proof that all terms are zero. Full-CoT's scratchpad immediately cross-references: "*doesn't align with the ground truth, which shows all $u$'s are zero*," then the `<THINKING>` phase assigns consequences. **This example illustrates the core failure: No-CoT checks whether the student's math is self-consistent, not whether its premises are correct.**

**Problem (Putnam #62)**

Let $f(x)$ be a cubic polynomial with roots $x_1, x_2, x_3$. Assume that $f(2x)$ is divisible by $f'(x)$. Compute the ratios $x_1 : x_2 : x_3$.

**Ground Truth (abridged)**

Writes $f(2x) = f'(x)(px + q)$, compares coefficients to get $b = 4a^2/3$, $c = -16a^3/27$. Factors $f(x)$ to find roots $1, -2 + 2\sqrt{3}\,i, -2 - 2\sqrt{3}\,i$ (up to scaling by $a$). Ratios involve **complex numbers**.

**Student Solution (abridged)**

Notes that roots of $f'(x)$, when doubled, must be roots of $f(x)$. Uses Vieta's formulas. Assumes all roots are real, concludes $x_1 : x_2 : x_3 = 2 : 1 : 1$ without justification. Does not consider complex roots.

**No-CoT Judge (score: 3.5 / 5)**

"The MODEL_OUTPUT's steps are locally valid and globally consistent. It outlines a logical progression [...] Logical Coherence: 1."

"The approach is compatible with the canonical method. Both the model and ground truth involve setting up integrals [sic] and solving for c [sic]. Methodological Alignment: 1."

"The model concludes the ratio is 2:1:1. However, the ground truth shows the roots are $1, -2+2\sqrt{3}\,i, -2-2\sqrt{3}\,i$, which don't align [...] The model's intermediate step about the ratio might be incorrect, so I'll give 0.5."

LC: 1 | FT: 1 | MA: 1 | IC: 0.5 | EA: 0 | **Total: 3.5**

**Full-CoT Judge (score: 1.5 / 5)**

*Scratchpad:*
"However, the model uses Vieta's formulas [...] but it doesn't apply them as thoroughly as the ground truth. The ground truth sets up equations by comparing coefficients, which is a more precise approach [...] 0.5."

⟨*THINKING*⟩ *summary:*
"Logical Coherence (0.5): [...] the leap to the ratio 2:1:1 is abrupt and lacks detailed justification."
"Faithfulness (0.5): [...] oversimplifies by assuming real roots and a specific ratio without considering complex roots as in the ground truth."
"Methodological Alignment (0.5): Uses Vieta's formulas [...] but applies it less rigorously than the ground truth."

LC: 0.5 | FT: 0.5 | MA: 0.5 | IC: 0 | EA: 0 | **Total: 1.5**

Figure 5: **Failure to propagate error backward through rubric criteria (Human: 1.5).** The No-CoT judge evaluates each criterion independently: "Does the reasoning flow logically?" (yes, on the surface)—and awards 1. It even *notices* the wrong answer ("the ground truth shows the roots are $1, -2+2\sqrt{3}\,i, \ldots$ which don't align") but confines the penalty to Intermediate Correctness (0.5) without revising the upstream criteria. Full-CoT recognizes that the wrong answer retroactively undermines Logical Coherence ("the leap to 2:1:1 is abrupt"), Faithfulness ("oversimplifies by assuming real roots"), and Methodology ("applies it less rigorously")—propagating the error backward to produce a correctly calibrated total. Note also that No-CoT *hallucinates* problem content: it refers to "setting up integrals and solving for $c$," neither of which appear in this problem.

# K Additional Strategy-Selection Headroom Results

This appendix reports robustness analyses for the MSE-aligned strategy-selection headroom results in Section 4. We include only analyses that answer distinct questions beyond the main table: sensitivity to the stability penalty $\lambda$, sensitivity to the token-cost penalty $\gamma$, and the distribution of oracle-selected strategies.

## K.1 Sensitivity to stability penalty

We first sweep the stability penalty $\lambda$ while fixing $\gamma = 0$. Oracle selection continues to improve over the validation-selected best fixed strategy across all tested $\lambda$ values, indicating that adaptive headroom is not tied to a single stability weighting.

Table 13: Putnam-AXIOM-Grading sensitivity to the stability penalty $\lambda$ with $\gamma = 0$. $\Delta$ columns report utility improvement under the corresponding $\lambda$. MSE is measured on the native 0–5 Putnam scale.

| $\lambda$ | Model | Best fixed | $\Delta$ vs fixed | $\Delta$ vs comp. | Oracle MSE | Fixed MSE | Comp. MSE |
|---|---|---|---|---|---|---|---|
| 0 | DeepSeek-R1-32B | Comparative | 0.022 | 0.022 | 0.303 | 0.850 | 0.850 |
| 0 | Llama-3.1-70B | Bullet-point | 0.024 | 0.033 | 0.181 | 0.790 | 0.995 |
| 0 | Qwen2.5-32B | Comparative | 0.027 | 0.027 | 0.157 | 0.824 | 0.824 |
| 0 | Qwen2.5-72B | Comparative | 0.029 | 0.028 | 0.104 | 0.829 | 0.812 |
| 0.25 | DeepSeek-R1-32B | Comparative | 0.022 | 0.022 | 0.314 | 0.850 | 0.850 |
| 0.25 | Llama-3.1-70B | Bullet-point | 0.024 | 0.032 | 0.192 | 0.790 | 0.995 |
| 0.25 | Qwen2.5-32B | Comparative | 0.026 | 0.026 | 0.173 | 0.847 | 0.824 |
| 0.25 | Qwen2.5-72B | Comparative | 0.029 | 0.028 | 0.115 | 0.842 | 0.812 |
| 0.5 | DeepSeek-R1-32B | Comparative | 0.023 | 0.023 | 0.329 | 0.850 | 0.850 |
| 0.5 | Llama-3.1-70B | Bullet-point | 0.025 | 0.033 | 0.204 | 0.790 | 0.995 |
| 0.5 | Qwen2.5-32B | Comparative | 0.025 | 0.026 | 0.188 | 0.839 | 0.824 |
| 0.5 | Qwen2.5-72B | Comparative | 0.028 | 0.027 | 0.131 | 0.845 | 0.812 |
| 1 | DeepSeek-R1-32B | Comparative | 0.026 | 0.026 | 0.355 | 0.850 | 0.850 |
| 1 | Llama-3.1-70B | Bullet-point | 0.027 | 0.035 | 0.235 | 0.790 | 0.995 |
| 1 | Qwen2.5-32B | Comparative | 0.025 | 0.028 | 0.219 | 0.839 | 0.824 |
| 1 | Qwen2.5-72B | Comparative | 0.027 | 0.028 | 0.155 | 0.827 | 0.812 |
| 2 | DeepSeek-R1-32B | Comparative | 0.033 | 0.033 | 0.402 | 0.850 | 0.850 |
| 2 | Llama-3.1-70B | Bullet-point | 0.033 | 0.040 | 0.294 | 0.790 | 0.995 |
| 2 | Qwen2.5-32B | No-CoT | 0.026 | 0.033 | 0.286 | 0.839 | 0.824 |
| 2 | Qwen2.5-72B | No-CoT | 0.028 | 0.031 | 0.200 | 0.827 | 0.812 |

### K.2 SENSITIVITY TO TOKEN-COST PENALTY

We next sweep the token-cost penalty $\gamma$ while fixing $\lambda = 1$. Gains shrink as token cost is penalized more heavily, but remain positive across all tested settings, indicating that the headroom is not simply an artifact of selecting expensive multi-call strategies.

Table 14: Putnam-AXIOM-Grading sensitivity to the cost penalty $\gamma$ with $\lambda = 1$. $\Delta$ columns report utility improvement under the corresponding $\gamma$; token counts are reconstructed approximate token counts.

| $\gamma$ | Model | Best fixed | $\Delta$ vs fixed | $\Delta$ vs comp. | Oracle tokens | Fixed tokens | Comp. tokens |
|---|---|---|---|---|---|---|---|
| 0 | DeepSeek-R1-32B | Comparative | 0.041 | 0.041 | 4775 | 2582 | 2582 |
| 0 | Llama-3.1-70B | Bullet-point | 0.033 | 0.046 | 3514 | 1956 | 2057 |
| 0 | Qwen2.5-32B | Comparative | 0.039 | 0.039 | 3362 | 2121 | 2121 |
| 0 | Qwen2.5-72B | Comparative | 0.037 | 0.037 | 3830 | 2118 | 2118 |
| 0.01 | DeepSeek-R1-32B | Comparative | 0.039 | 0.039 | 3009 | 2582 | 2582 |
| 0.01 | Llama-3.1-70B | Bullet-point | 0.031 | 0.044 | 2300 | 1956 | 2057 |
| 0.01 | Qwen2.5-32B | Comparative | 0.037 | 0.037 | 2397 | 2121 | 2121 |
| 0.01 | Qwen2.5-72B | No-CoT | 0.031 | 0.034 | 2822 | 1677 | 2118 |
| 0.05 | DeepSeek-R1-32B | Comparative | 0.038 | 0.038 | 2448 | 2582 | 2582 |
| 0.05 | Llama-3.1-70B | Bullet-point | 0.030 | 0.044 | 2063 | 1956 | 2057 |
| 0.05 | Qwen2.5-32B | No-CoT | 0.024 | 0.037 | 1977 | 1680 | 2121 |
| 0.05 | Qwen2.5-72B | No-CoT | 0.023 | 0.031 | 2214 | 1677 | 2118 |
| 0.1 | DeepSeek-R1-32B | Brief-CoT | 0.035 | 0.040 | 2364 | 2063 | 2582 |
| 0.1 | Llama-3.1-70B | Bullet-point | 0.029 | 0.044 | 1952 | 1956 | 2057 |
| 0.1 | Qwen2.5-32B | No-CoT | 0.021 | 0.040 | 1858 | 1680 | 2121 |
| 0.1 | Qwen2.5-72B | No-CoT | 0.019 | 0.032 | 1946 | 1677 | 2118 |
| 0.25 | DeepSeek-R1-32B | Brief-CoT | 0.028 | 0.049 | 2219 | 2063 | 2582 |
| 0.25 | Llama-3.1-70B | Bullet-point | 0.032 | 0.050 | 1832 | 1956 | 2057 |
| 0.25 | Qwen2.5-32B | No-CoT | 0.016 | 0.052 | 1756 | 1680 | 2121 |
| 0.25 | Qwen2.5-72B | No-CoT | 0.014 | 0.043 | 1753 | 1677 | 2118 |

### K.3 ORACLE STRATEGY FREQUENCIES

The oracle does not collapse to a single strategy: Putnam selections are distributed across nearly all candidate strategies for each model, and the distribution is strongly model-dependent. This supports the interpretation that strategy utility is instance-conditional rather than captured by a single global ranking.

Table 15: Putnam-AXIOM-Grading oracle-selected strategy counts under $\lambda = 0, \gamma = 0$. Counts are over held-out test instances.

| Model | Brief | Bullet | Comp. | CoVe | Debate | Full | G-Eval | Multi | No-CoT | Overseer | Quote | Self | Struct. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DeepSeek-R1-32B | 64 | 77 | 103 | 57 | 120 | 66 | 138 | 83 | 36 | 54 | 89 | 35 | 78 |
| Llama-3.1-70B | 53 | 124 | 79 | 70 | 81 | 92 | 76 | 55 | 73 | 62 | 49 | 83 | 103 |
| Qwen2.5-32B | 66 | 91 | 102 | 86 | 139 | 56 | 49 | 40 | 90 | 57 | 70 | 61 | 93 |
| Qwen2.5-72B | 43 | 56 | 98 | 60 | 214 | 57 | 121 | 49 | 82 | 49 | 64 | 48 | 59 |

## L Strategy Pool Complementarity Analysis

The 13 protocols are intended to cover distinct judge behaviors rather than to assert that every protocol is equally useful. We therefore ask how much of the full-pool oracle headroom can be recovered by smaller subsets. Across the eight Putnam– and OPC–judge conditions, using losses averaged across the four controlled prompt variants, a complementarity-selected five-strategy pool recovers 90.4% of the full-pool oracle headroom on held-out problems (95% CI [88.5%, 92.1%]), compared with 78.3% for the five individually strongest strategies. Its paired held-out loss is lower by 0.00264 (95% CI [0.00178, 0.00362]), and it performs at least as well as 96.2% of all size-five alternatives on average. The utility curve has a strong-core-plus-long-tail shape: three strategies recover 80.2% of full-pool headroom, five recover 90.4%, and seven recover 96.3%.

A leave-one-strategy-out analysis gives a complementary view. Removing any one of the 13 strategies increases oracle loss, with a positive problem-clustered 95% bootstrap lower bound for every removal. Eleven removals harm coverage in all eight benchmark–judge conditions, while the remaining two harm seven of eight. Contributions are highly unequal, but no strategy is empirically redundant conditional on the other twelve. These results motivate treating the full set as a discovery pool while recognizing that most deployable headroom can be captured by a substantially smaller complementary subset.

Table 16: Holm- and BH-corrected fixed-strategy MSE comparisons.

| Benchmark | Judge | Numerical winner | Holm top set | Holm resolved | BH top set | BH resolved |
|---|---|---|---|---|---|---|
| Putnam | Qwen2.5-32B | Comparative | Comparative, No-CoT | 49/78 | Comparative, No-CoT | 59/78 |
| Putnam | Qwen2.5-72B | Comparative | Comparative, No-CoT, CoVe | 58/78 | Comparative, No-CoT | 66/78 |
| Putnam | DeepSeek-R1-32B | Comparative | Comparative, G-Eval | 44/78 | Comparative, G-Eval | 55/78 |
| Putnam | Llama-3.1-70B | Bullet-Point | Bullet-Point | 53/78 | Bullet-Point | 65/78 |
| OPC | Qwen2.5-32B | Debate | Debate | 70/78 | Debate | 73/78 |
| OPC | Qwen2.5-72B | Debate | Debate | 64/78 | Debate | 71/78 |
| OPC | DeepSeek-R1-32B | Debate | Debate, Bullet-Point | 70/78 | Debate, Bullet-Point | 71/78 |
| OPC | Llama-3.1-70B | Bullet-Point | Bullet-Point | 72/78 | Bullet-Point | 72/78 |

## M Statistical Comparisons of Fixed Strategies

Within each benchmark–judge panel, we perform all 78 pairwise strategy comparisons using shared problem-clustered bootstrap resamples. We apply Holm–Bonferroni correction as the primary family-wise procedure and Benjamini–Hochberg (BH) correction as a false-discovery-rate sensitivity analysis, separately by panel and metric. Table 16 reports the numerical MSE winner, the Holm and BH statistical top sets, and the number of pairwise comparisons resolved by each procedure.

Across the eight panels, Holm resolves 480/624 MSE comparisons (76.9%), while BH resolves 532/624 (85.3%). Both procedures yield singleton MSE top sets in 4/8 panels and leave all numerical winners unchanged. Their MSE top sets are identical except for Putnam Qwen2.5-72B, where CoVe is included under Holm but not BH. For run variance, Holm and BH resolve 471/624 (75.5%) and 528/624 (84.6%) comparisons, respectively, with singleton top sets in 7/8 and 8/8 panels. Thus, the absence of a universal winner is not caused merely by statistically indistinguishable methods: many within-panel orderings are resolved, but the identity of the best protocol changes across judges and benchmarks.

## N PROMPT PARAPHRASE ROBUSTNESS

For each strategy, we generated six candidate paraphrases while holding fixed the rubric, structural tags, examples, stage order, quantitative constraints, frozen literals, and output schema. We screened candidates for operation preservation and length within 75–125% of the original, then selected three for lexical and syntactic diversity. The original prompt (P0) and paraphrases P1–P3 were evaluated for all four open-weight judges, with five repeated runs on 300-candidate subsets of Putnam-AXIOM-Grading and OPC-Grading.

Strategy choice dominates wording choice. Between-strategy variation in per-run squared error is 14.77× within-strategy prompt variation across the benchmark–judge conditions (95% CI [11.83, 15.25]); the ratio is above one in all eight conditions. A direct distance-based analysis agrees: the mean MSE difference between strategies is 3.66× the difference between wordings of the same strategy. Across 24 original-versus-paraphrase comparisons, the mean Spearman correlation between the 13-strategy rankings is 0.894, 88.4% of pairwise orderings are preserved, and exact winner identity remains unchanged in 16/24 comparisons (66.7%). The original winner remains in the paraphrased top three in 23/24 comparisons (95.8%), while the paraphrased winner belongs to the original top three in all 24 comparisons. When the winner changes, the original winner trails by a median normalized MSE of only 0.0049. Across all 32 prompt–condition panels, Comparative, Bullet-Point, or Debate ranks first in 31 (96.9%).

## O EXPANDED ANNOTATION RELIABILITY AND DISAGREEMENT AUDIT

We expanded blind independent double-grading from the original small validation subset to 500 solutions: 250 Putnam and 250 OPC candidates. Table 17 reports adjacent-score agreement, quadratic-weighted kappa (QWK), and MAE on the 0–5 scale.

Table 17: Expanded independent double-grading results.

| Benchmark | $N$ | Within 0.5 | Within 1 | QWK | MAE |
|---|---|---|---|---|---|
| Putnam | 250 | 75.2% | 91.2% | .857 | .518 |
| OPC | 250 | 74.8% | 87.6% | .815 | .562 |
| Overall | 500 | 75.0% | 89.4% | .842 | .540 |

**Representative disagreement.** Figure 6 shows a representative case in which the annotators agree on the mathematical deficiency but differ in the amount of partial credit assigned.

**Putnam 1958, Problem 10 — Qwen2.5-72B-Math-Instruct**

**Problem.** "If a square of unit side be partitioned into two sets, what is the minimum possible value of the larger diameter among the two sets? (The diameter of a set is defined as the least upper bound of distances between pairs of points in that set.)"

**Reference solution (relevant excerpt)**

Suppose that the square is partitioned into two sets $S$ and $T$ of diameter less than $\sqrt{5}/2$, and choose the notation so that $A \in S$. Then $F \in T$, $D \in S$, $E \in T$, and $C \in S$. Thus $A$ and $C$ are in the same subset $S$, but

$$|AC| = \sqrt{2} > \sqrt{5}/2,$$

contradicting the fact that the diameter of $S$ is less than $\sqrt{5}/2$.

...

On the other hand, one can clearly partition the square into two rectangular sets of diameter $\sqrt{5}/2$.

...

Thus the final answer is

$$\boxed{\sqrt{5}/2}.$$

**What the reference establishes:**

- a **lower bound**: no partition can make both diameters smaller than $\sqrt{5}/2$;
- a matching **construction**: two $1 \times \frac{1}{2}$ rectangles.

**Candidate response (relevant excerpt)**

"One such partition is to draw a line through the center of the square, parallel to one of its sides. This will divide the square into two congruent rectangles, each with a side of length 1 and a side of length $\frac{1}{2}$."

...

"Any other partition will either result in one set having a diameter equal to or greater than the diagonal of a rectangle with sides 1 and $\frac{1}{2}$, or will result in one set having a diameter equal to or greater than the diagonal of the original square. Therefore, the partition into two congruent rectangles provides the minimum possible value."

**Shared annotator diagnosis:** the construction is correct and establishes the **upper bound**, but the claim beginning "Any other partition..." is asserted without a valid argument. The response therefore **does not prove optimality**.

**Where the annotators differ in partial credit**

| Rubric component | Grader 1 | Grader 2 |
|---|---|---|
| Methodological Alignment | 0 | 0.5 |
| Intermediate Correctness | 0.5 | 1.0 |

Both annotators identify the same mathematical omission; Grader 2 gives more credit for the correct construction and calculation. The disagreement is therefore about **partial-credit severity and rubric allocation**, not about whether the optimality proof is missing.

Figure 6: **Representative human-annotator disagreement.** The reference proves both the lower bound and a matching construction, whereas the candidate gives the construction but does not justify the lower bound. Both annotators identify this same omission but assign different partial credit to the affected rubric components.

The larger study gives a more conservative estimate of annotation reliability than a smaller initial pilot annotation. Rather than treating expert scores as exact ground truth, we therefore test whether scientific conclusions depend on the annotator. Recomputing fixed-strategy analyses independently against Grader 1 and Grader 2 gives mean strategy-rank Spearman correlation 0.962 across the eight benchmark–judge panels (range 0.940–0.995). Numerical winners agree in 7/8 panels, and 586/624 pairwise strategy-effect directions (93.9%) are unchanged. In the sole winner mismatch, Comparative and No-CoT belong to the Holm top set under both annotators. Absolute MSEs and marginal

top-set membership can therefore change with partial-credit calibration, but the broad protocol comparisons are stable.

We additionally ask whether a strong held-out model systematically exposes human labeling mistakes. GPT-5.2 independently graded all 500 double-annotated solutions without seeing either human score. Its agreement with Grader 1 is QWK 0.825, MAE 0.600, and 90.0% within one point; with Grader 2 it is QWK 0.797, MAE 0.640, and 84.0% within one point. GPT-5.2 is systematically stricter by roughly 0.31–0.32 points. Among the 74 cases where the humans agree within 0.5 points but GPT-5.2 falls at least one point outside their interval, we separately audit the 15 largest disagreements per benchmark. The audit supports the humans in 14/30 cases, GPT-5.2 in 12/30, and an intermediate or ambiguous assessment in 4/30. Primary causes include partial-credit calibration (8), missing-justification severity (7), rubric allocation (5), human-missed substantive errors (2), valid alternative methods (3), GPT-5.2-missed errors (4), and rubric ambiguity (1). Strong-model disagreement therefore does not uniformly favor either the humans or the model.

## P Response-Aware Routing Ablation

We add the candidate solution to the router input while holding fold assignments, targets, hyperparameters, random seeds, and context-window size fixed. Table 18 reports MSE on the native 0–5 scale; negative gaps indicate that response-aware routing is better.

Table 18: Response-aware versus response-blind MoG (MSE on the native 0–5 scale; lower is better). Blind is the main MoG result (Tables 2 and 3). Gap = response-aware − response-blind, estimated from paired runs with 95% paired bootstrap CIs; response-aware MSE is Blind + Gap.

| Corpus | Judge | Blind | Response-aware | Gap | 95% CI |
|---|---|---|---|---|---|
| OPC | Qwen2.5-32B | 1.330 | 1.355 | $+.025$ | $[-.003, +.065]$ |
| OPC | Qwen2.5-72B | .960 | .988 | $+.028$ | $[-.015, +.075]$ |
| OPC | DeepSeek-R1-32B | 2.020 | 2.002 | $-.018$ | $[-.033, -.000]$ |
| OPC | Llama-3.1-70B | .905 | .895 | $-.010$ | $[-.025, +.003]$ |
| Putnam | Qwen2.5-32B | .630 | .622 | $-.008$ | $[-.023, +.005]$ |
| Putnam | Qwen2.5-72B | .586 | .581 | $-.005$ | $[-.023, +.010]$ |
| Putnam | DeepSeek-R1-32B | .789 | .797 | $+.008$ | $[-.018, +.033]$ |
| Putnam | Llama-3.1-70B | .737 | .755 | $+.018$ | $[-.018, +.055]$ |

Seven of eight comparisons are statistical ties. The sole significant difference is the very small DeepSeek-on-OPC improvement ($-0.018$); there is no consistent direction across judges or benchmarks. Diagnostics confirm that the response-aware router genuinely uses the candidate: routing distributions for the two candidates of the same problem differ by mean $L_1$ distance up to 0.32, versus exactly zero for the response-blind router. Yet response awareness changes agreement with the per-instance oracle-best strategy by at most two percentage points and decreases it in four of the six conditions for which we computed this diagnostic. This suggests that the extra response information changes routing decisions without reliably making them more accurate.

## Q Strategy-Matrix Construction Cost

Constructing the strategy–score matrix is the primary setup cost of MoG. For $N = 1{,}000$ candidates and one judge, a one-trial pass over all 13 strategies requires 24,070 model calls and approximately 62.46M tokens at our observed sequence lengths. The five-trial protocol used to reduce cell noise and estimate variance requires 120,350 calls and approximately 312.28M tokens. Relative to a one-trial Comparative sweep, the one-trial matrix costs 24.07× as many calls and about 23.7× as many tokens; relative to Debate, it costs 8.02× as many calls and 6.89× as many tokens. These are one-time benchmark–judge-specific costs. The reported expected-mixture formulation combines the constituent protocol scores using the learned routing distribution; a hard-routing variant can instead execute only one selected protocol plus the router encoder pass. Therefore, Comparative is the natural choice for one-off grading, while MoG is most defensible when the benchmark will be reused extensively or when improved agreement is worth the up-front matrix construction cost.

## R TRAINING-SET SIZE ABLATION

Every training problem must carry an expert label and be graded under all 13 strategies before it can supervise the router (Appendix Q), so the size of the training set directly determines MoG's setup cost. The main experiments train each router on the full training split of its fold (400 problems on OPC, 432 on Putnam). Here we measure how MoG's accuracy changes as that set shrinks.

**Setup.** We use the QWEN2.5-72B-INSTRUCT judge on both benchmarks, with the same two response sets per benchmark as in the main experiments (O3 and QWEN3-235B-A22B on OPC-Grading; QWEN2.5-72B-MATH-INSTRUCT and MISTRAL-LARGE-INSTRUCT-2411 on Putnam-AXIOM-Grading). The 5-fold splits are unchanged, and the held-out problems of each fold (100 on OPC, 108 on Putnam) are never modified, so every training size is evaluated on exactly the same instances as the main run. For each fold and each $N \in \{10, 25, 50, 100, 200\}$, the training set is a random subset of $N$ of the fold's training problems. We repeat this for five trials: trial $t$ shuffles the fold's training problems once with seed $t$ and takes the first $N$, so subsets are nested within a trial (the $N$=10 problems are among the $N$=25 problems, and so on) and independent across trials. The full training set (400 problems on OPC, 432 on Putnam) is the main-paper run, which we reuse rather than retrain.

**Training.** Every (size, trial, fold) combination trains a new router from scratch on the pooled response instances for the selected training problems, giving $5 \times 5 \times 5 = 125$ routers per benchmark. For each selected problem, both candidate-generator responses are included in the training set. The recipe is identical to the main runs: MODERNBERT-LARGE with mean pooling and a 13-way linear head, the mixture-MSE objective of Section 3.4, AdamW (learning rate $2 \times 10^{-5}$ for the encoder and $10^{-3}$ for the head), batch size 16, two head-only epochs followed by six joint epochs, and router seed 0. Inputs are truncated to 1,024 tokens on OPC and Putnam, as in the corresponding main runs. Because batch size and epoch count are fixed, smaller training sets also receive fewer optimizer steps: 8, 16, 32, 56, 104, and 200 steps for $N$ = 10, 25, 50, 100, 200, and 400. We did not retune the schedule for small $N$, so the curve reflects both less data and fewer updates.[1]

**Evaluation.** For each (size, trial), we concatenate the held-out predictions of the five folds and the two response sets, giving $Q_{\text{eff}} = 1{,}000$ instances on OPC and $1{,}080$ on Putnam, each predicted by a router that never trained on it. MoG is scored as the expected mixture $\tilde{y}_q = \sum_s p_\theta(s \mid x_q) S_{q,s}$. The best fixed strategy is chosen by lowest MSE on the same held-out instances; because the test set never changes, it is Debate on OPC and Comparative on Putnam at every $N$. We compare MoG against it with the paired bootstrap of Section 3.4 (10,000 resamples over per-instance squared-error differences). Table 19 reports the mean and standard deviation across the five trials. The standard deviation reflects which training problems were drawn; uncertainty from test sampling is captured by each trial's own bootstrap CI. The full-data run is a single run and has no standard deviation. All MSEs are on the native 0–5 scale.

Table 19: MoG with reduced training sets (QWEN2.5-72B-INSTRUCT; native 0–5 scale; lower MSE is better). Gap = MoG − best fixed strategy on the same held-out instances ($Q_{\text{eff}} = 1{,}000$ on OPC, $1{,}080$ on Putnam) (Debate, MSE 1.105, on OPC; Comparative, MSE .812, on Putnam); negative values favor MoG. Entries are mean ± standard deviation over five random training subsets. "CI < 0" counts trials whose 95% paired bootstrap CI for the gap lies entirely below zero.

| | **OPC-Grading** | | | **Putnam-AXIOM-Grading** | | |
|---|---|---|---|---|---|---|
| $N$ | MoG MSE | Gap vs. Debate | CI < 0 | MoG MSE | Gap vs. Comp. | CI < 0 |
| 10 | 1.045 ± .063 | −.060 ± .063 | 4/5 | .663 ± .048 | −.150 ± .048 | 5/5 |
| 25 | 1.018 ± .010 | −.088 ± .010 | 5/5 | .623 ± .013 | −.188 ± .013 | 5/5 |
| 50 | 1.008 ± .015 | −.098 ± .015 | 5/5 | .618 ± .018 | −.195 ± .018 | 5/5 |
| 100 | .970 ± .018 | −.138 ± .018 | 5/5 | .598 ± .008 | −.213 ± .008 | 5/5 |
| 200 | .953 ± .010 | −.153 ± .010 | 5/5 | .595 ± .005 | −.218 ± .005 | 5/5 |
| Full (main run) | .960 | −.145 | — | .586 | −.226 | — |

[1] As in the main runs, the checkpoint epoch is chosen by MSE on the held-out fold. This selection is applied identically at every $N$, so it does not favor any training size.

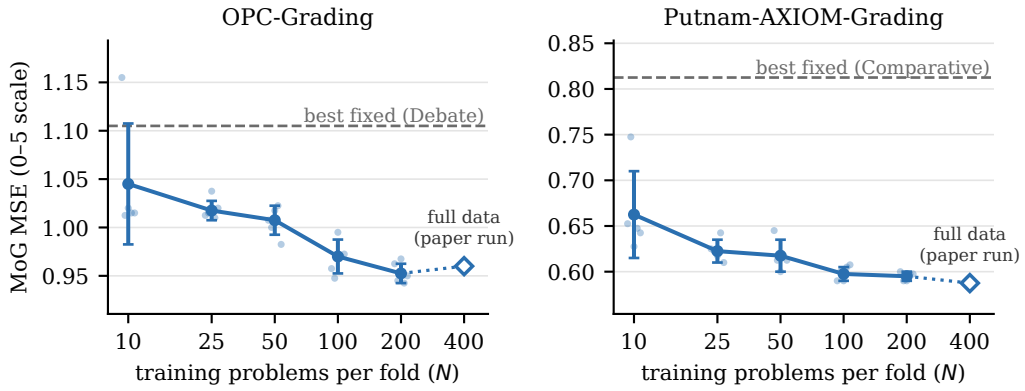



Figure 7: **MoG beats the best fixed strategy even with 10 training problems per fold, and saturates by $N{\approx}100$.** MoG MSE (native 0–5 scale; lower is better) for QWEN2.5-72B-INSTRUCT as a function of the number of training problems per fold, with the held-out test set fixed. Solid points and error bars show the mean $\pm$ one standard deviation over five random training subsets; faint points are individual subsets, each pooled over all held-out instances ($Q_{\text{eff}} = 1{,}000$ on OPC, 1,080 on Putnam). The hollow diamond is the main run (full training set, a single run). The dashed line is the best fixed strategy on the same instances.

**Results.** MoG beats the best fixed strategy at every training size (Table 19, Figure 7). On Putnam, all 25 subsampled routers have a 95% CI entirely below zero. On OPC, 24 of 25 do; the exception is a single $N{=}10$ trial whose gap is positive ($+.050$). Even at $N{=}10$, the mean relative MSE reduction is 5.4% on OPC and 18.5% on Putnam, against 13.1% and 27.8% with the full training set.

Most of the benefit arrives early. At $N{=}100$, MoG reduces MSE by 12.4% on OPC and 26.2% on Putnam, which is 95% and 94% of the full-data gap ($-.138$ vs. $-.145$ and $-.213$ vs. $-.226$). At $N{=}200$, the gap essentially matches the full-data result ($-.153$ vs. $-.145$ on OPC and $-.218$ vs. $-.226$ on Putnam). The OPC mean at $N{=}200$ is slightly better than the single full-data run, which is within the spread we observe across training subsets.

Sensitivity to the particular training problems is confined to the smallest size. The standard deviation across trials is .063 (OPC) and .048 (Putnam) at $N{=}10$, and at most .018 for every $N \geq 25$. With very few problems, an unlucky draw can leave the router with a poor estimate of which strategies help; by $N{=}25$ that risk has largely disappeared.

**Implications.** In this setting, about 100 labeled and fully swept training problems per fold, a quarter of the training set used in our main experiments, recover most of MoG's improvement over the best fixed protocol. This reduces the one-time cost of building the strategy–score matrix for the training set by roughly a factor of four. This ablation covers one judge; we expect the qualitative pattern to hold for the others but have not verified it.

# S PROMPTS

## S.1 GRADING PROMPTS (PUTNAM-AXIOM-GRADING AND OPC-GRADING)

**Base Context (Common to All Strategies)**

*Note: All strategies typically begin with this context block.*
<CONTEXT> [Rubric Text Omitted] </CONTEXT>
<PROBLEM> {problem_statement} </PROBLEM>
<GROUND_TRUTH_SOLUTION> {ground_truth_solution}
</GROUND_TRUTH_SOLUTION>
<MODEL_OUTPUT> {model_output} </MODEL_OUTPUT>

### SINGLE-TURN STRATEGIES

**Strategy: No CoT**

<INSTRUCTIONS> Provide your scores DIRECTLY in the specified JSON format inside <json> tags. Do NOT include any reasoning.
Your response MUST include a <json>...</json> block and NOTHING after it.
Here is an example response: <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 1, "score_error_awareness": 1, "score_total": 5 } </json> </INSTRUCTIONS>

**Strategy: Brief CoT**

<INSTRUCTIONS> First, write ONLY 1-2 SHORT sentences inside <THINKING> tags summarizing your overall assessment. Do NOT explain your entire process. Just state your assessment directly. Then provide your scores in the specified JSON format inside <json> tags.
Example format: <THINKING> The solution correctly applies integration by parts but makes an arithmetic error in step 3. Otherwise reasoning is sound. </THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 0, "score_error_awareness": 0, "score_total": 3 } </json> </INSTRUCTIONS>

**Strategy: Structured CoT**

<INSTRUCTIONS> You MUST follow this exact three-step structure inside <THINKING> tags:
STEP 1 - SUMMARIZE: In 2-3 sentences, summarize what the model's solution attempts to do.
STEP 2 - CHECK KEY STEPS: List the 2-4 most critical steps/calculations. For each, note if it is correct (✓) or incorrect (×).
STEP 3 - DECIDE SCORES: Based on your analysis, determine each rubric score with one-line justification.
Then provide your final scores in <json> tags.
Example: <THINKING> STEP 1 - SUMMARIZE: The model attempts to solve the integral using substitution, converting the expression and evaluating bounds.
STEP 2 - CHECK KEY STEPS: - Substitution $u = x^2 + 1$: ✓Valid transformation - Computing $du = 2xdx$: ✓Correct derivative - Final evaluation at bounds: ×Arithmetic error (should be 15, got 12)
STEP 3 - DECIDE SCORES: - Logical coherence: 1 (steps follow logically) - Faithfulness: 1 (addresses the problem) - Methodological alignment: 1 (standard approach) - Intermediate correctness: 0 (arithmetic error) - Error awareness: 0 (error not noticed)

</THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 0, "score_error_awareness": 0, "score_total": 3 } </json> </INSTRUCTIONS>

**Strategy: Full CoT**

<INSTRUCTIONS> You MUST output your response in this EXACT format: 1. FIRST, write your step-by-step reasoning inside <THINKING>...</THINKING> tags 2. THEN, output the scores inside <json>...</json> tags
For instance, a valid grading response to a correct answer could look like this: "<THINKING> Logical Coherence (1): The argument proceeds in a clear sequence, with each step following from the last; no leaps or contradictions detected. Faithfulness to Task (1): The response stays focused on the stated goal and uses only information relevant to answering the problem. The response never strays away from the given task. Methodological Alignment (1): The chosen method is standard for this type of problem and is applied appropriately. Intermediate Correctness (1): Intermediate claims match known identities/definitions and are used consistently. Error Awareness / Self-Correction (1): No mistakes observed; scope conditions are implicitly respected and edge cases would not change the conclusion. </THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 1, "score_error_awareness": 1, "score_total": 5 } </json>"
A valid grading response to an incorrect answer could look like this: "<THINKING> Logical Coherence (0): The reasoning is inconsistent: it mixes unrelated steps, reverses implications, and reaches conclusions that don't follow from prior claims. Faithfulness to Task (1): The response attempts to answer the stated question and references the required quantities, but stays narrowly focused while still arriving at the wrong result. Methodological Alignment (0): The chosen approach conflicts with standard methods for this problem and misapplies key definitions/identities. Intermediate Correctness (0): Several intermediate statements are false (e.g., misuse of a determinant identity and an invalid simplification), so later conclusions are unsupported. Error Awareness / Self-Correction (0): No checks, sanity tests, or alternative cases are considered; contradictions and edge conditions go unaddressed. <json> { "score_logical_coherence": 0, "score_faithfulness_to_task": 1, "score_methodological_alignment": 0, "score_intermediate_correctness": 0, "score_error_awareness": 1, "score_total": 1 } </json>" </INSTRUCTIONS>

**Strategy: Self-Critique**

<INSTRUCTIONS> Follow this two-phase evaluation inside <THINKING> tags:
PHASE 1 - INITIAL ASSESSMENT: Quickly evaluate the solution and propose initial scores.
PHASE 2 - SELF-CRITIQUE: Challenge your initial assessment. Ask: "Am I being too harsh or too lenient?" "Did I miss anything?" Adjust scores if needed.
Then provide your final scores in <json> tags.
Example: <THINKING> PHASE 1 - INITIAL ASSESSMENT: The solution uses the correct method but has a sign error. Initial scores: coherence=1, faithfulness=1, method=1, intermediate=0, awareness=0.
PHASE 2 - SELF-CRITIQUE: Wait - I should check if the sign error propagates. Looking again... yes, it affects the final answer but the method is still valid. The model did attempt to verify the answer at the end, showing some error awareness even if unsuccessful. Adjusting error_awareness to 0.5. </THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 0, "score_error_awareness": 0.5, "score_total": 3.5 } </json> </INSTRUCTIONS>

**Strategy: Bullet Point**

<INSTRUCTIONS> Use bullet points inside <THINKING> tags to systematically evaluate each criterion:
• Logical Coherence: [observation] → [score] • Faithfulness to Task: [observation] → [score] • Methodological Alignment: [observation] → [score] • Intermediate Correctness: [observation] → [score] • Error Awareness: [observation] → [score] • Total: [sum]
Then provide scores in <json> tags.
Example: <THINKING> • Logical Coherence: Steps follow clear sequence with proper transitions → 1 • Faithfulness to Task: Directly addresses the integral as asked → 1 • Methodological Alignment: Uses substitution, same as ground truth → 1 • Intermediate Correctness: Calculation error in step 4 (8x3=24, not 26) → 0 • Error Awareness: No verification attempted → 0 • Total: 3 </THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 0, "score_error_awareness": 0, "score_total": 3 } </json> </INSTRUCTIONS>

**Strategy: Comparative**

<INSTRUCTIONS> Inside <THINKING> tags, perform a comparative analysis:
1. KEY APPROACH COMPARISON: - Ground Truth approach: [brief description] - Model approach: [brief description] - Match? [Yes/No/Partial]
2. CRITICAL DIFFERENCES: List any significant differences in reasoning or results.
3. ERROR ANALYSIS: Note any errors in the model output that differ from ground truth.
4. SCORING RATIONALE: Brief justification for each score based on the comparison.
Then provide scores in <json> tags.
Example: <THINKING> 1. KEY APPROACH COMPARISON: - Ground Truth: Uses integration by parts twice - Model: Uses integration by parts twice - Match? Yes
2. CRITICAL DIFFERENCES: - Model skips showing the second application explicitly - Final constant differs (model has +C, GT shows specific value)
3. ERROR ANALYSIS: - Sign error in step 2: -cos(x) written as cos(x) - This propagates to incorrect final answer
4. SCORING RATIONALE: - Coherence: 1 (logical flow maintained) - Faithfulness: 1 (addresses the integral) - Method: 1 (same technique) - Intermediate: 0 (sign error) - Awareness: 0 (not caught) </THINKING> <json> { "score_logical_coherence": 1, "score_faithfulness_to_task": 1, "score_methodological_alignment": 1, "score_intermediate_correctness": 0, "score_error_awareness": 0, "score_total": 3 } </json> </INSTRUCTIONS>

**Strategy: Quote Forcing**

<INSTRUCTIONS> You MUST output your response in this EXACT format: 1. FIRST, write your reasoning inside <THINKING>...</THINKING> tags, citing evidence from <MODEL_OUTPUT> for EVERY evaluative claim with short direct quotes like: "..." 2. THEN, output the scores inside <json>...</json> tags
Your response MUST start with <THINKING> and include quotes before the JSON block.
Your response MUST include a <json>...</json> block and NOTHING after it.
Example: <THINKING> The model claims to derive the formula, stating "$f(x) = x^2 + 2x$" [Quote from Line 4]. However, this contradicts the prompt which asked for "$f(x) = x^3$". The model later attempts to correct this: "Wait, I meant cubed" [Quote from Line 6], but the calculation remains quadratic. </THINKING> <json> { "score_logical_coherence": 0, "score_faithfulness_to_task": 1, "score_methodological_alignment": 0, "score_intermediate_correctness": 0, "score_error_awareness": 0, "score_total": 1 } </json> </INSTRUCTIONS>

MULTI-TURN STRATEGIES

**Strategy: Debate**

**Advocate Judge (Pro):** <INSTRUCTIONS> You are the "Advocate Judge". Your goal is to argue why this response deserves the HIGHEST POSSIBLE SCORE. Highlight every correct step, and every logical connection. Output your argument in <ARGUMENT_PRO> tags. </INSTRUCTIONS>

**Critic Judge (Con):** <INSTRUCTIONS> You are the "Critic Judge". Your goal is to argue why this response deserves the LOWEST POSSIBLE SCORE. Scrutinize every step, look for any logical gap, and interpret ambiguity as failure. Output your argument in <ARGUMENT_CON> tags. </INSTRUCTIONS>

**Arbiter:** <ARGUMENT_PRO> {pro_arg} </ARGUMENT_PRO>
<ARGUMENT_CON> {con_arg} </ARGUMENT_CON>
<INSTRUCTIONS> You are the Arbiter. Read the Pro and Con arguments. Decide which is more reasonable based on the Rubric. Usually the truth lies in the middle, but be decisive if one side is clearly hallucinating. Provide your neutral reasoning in <THINKING> and final scores in <json>. </INSTRUCTIONS>

**Strategy: G-Eval**

**Step 1 (Plan):** <INSTRUCTIONS> Your task is to convert the Rubric into atomic evaluation steps specific to this problem. 1. Break down the specific requirements of the problem and ground truth. 2. Create a checklist of 3-5 atomic steps/checks. 3. Mark each step as [CRITICAL] or [NON-CRITICAL].
Output inside <CHECKLIST_PLAN> tags.
Example: <CHECKLIST_PLAN> 1. [CRITICAL] Does the model correctly identify the integration bounds (0 to infinity)? 2. [CRITICAL] Is the substitution $u = x^2$ applied correctly? 3. [NON-CRITICAL] Is the final answer simplified to the simplest form? 4. [CRITICAL] Does the model check for convergence? </CHECKLIST_PLAN> </INSTRUCTIONS>

**Step 2 (Score):** <EVALUATION_PLAN> {step1_output} </EVALUATION_PLAN>
<INSTRUCTIONS> Evaluate the model output against the Checklist Plan above. For each criterion in the original rubric, calculate the score using this logic: - Score = 1 if ALL [CRITICAL] checks pass AND <= 1 [NON-CRITICAL] check fails. - Score = 0 if ANY [CRITICAL] check fails. - Score = 0.5 otherwise.
Show your work in <THINKING> tags, verifying each checklist item as PASS/FAIL. Output final scores in <json>. </INSTRUCTIONS>

**Strategy: CoVe**

**Step 2 (Plan Verification):** <DRAFT_EVALUATION> {draft_response} </DRAFT_EVALUATION>
<INSTRUCTIONS> Based on the draft evaluation above and the problem, generate 3-5 specific "Verification Questions" to fact-check the draft's claims. Focus on checking if the draft missed errors or hallucinations in the model output. Output inside <VERIFICATION_QUESTIONS>.
Example: <VERIFICATION_QUESTIONS> - Did the model actually calculate the derivative of log(x) correctly in step 2? - The draft claims the model followed the prompt constraints; does the model output contain the forbidden word "delve"? - Is the final constant matching the Ground Truth exactly? </VERIFICATION_QUESTIONS> </INSTRUCTIONS>

**Step 3 (Execute Verification):** <QUESTIONS> {questions} </QUESTIONS>
<INSTRUCTIONS> Answer the verification questions above objectively based *strictly* on the Model Output and Ground Truth. Do not grade yet, just answer the questions inside <VERIFICATION_ANSWERS>.

Example: <VERIFICATION_ANSWERS> 1. No, the model calculated $1/x^2$ instead of $1/x$. 2. Yes, the model used the word "delve". 3. No, Ground Truth is 15, Model is 14. </VERIFICATION_ANSWERS> </INSTRUCTIONS>

**Step 4 (Finalize):** <DRAFT_EVALUATION> {draft} </DRAFT_EVALUATION> <VERIFICATION_RESULTS> {qa_pairs} </VERIFICATION_RESULTS> <INSTRUCTIONS> Review the Draft Evaluation in light of the Verification Results. If the verification found errors in the model that the draft missed, downgrade the score. If the verification confirms the draft was correct, maintain the score. Provide final reasoning and <json>. </INSTRUCTIONS>

**Strategy: Multi-Judge (Base Judges)**

*Note: Three separate judges grade the problem in parallel, each with a distinct persona prepended to the standard Full CoT instructions.*
**Judge 1 (Pedantic):** "You are a strict, pedantic judge who focuses on details."
**Judge 2 (Holistic):** "You are a holistic judge who focuses on the big picture."
**Judge 3 (Teacherly):** "You are a helpful teacher grading a student."

*All three judges then receive the standard Full CoT prompt:* <INSTRUCTIONS> You MUST output your response in this EXACT format: 1. FIRST, write your step-by-step reasoning inside <THINKING>...</THINKING> tags 2. THEN, output the scores inside <json>...</json> tags
For instance, a valid grading response to a correct answer could look like this: "<THINKING> ... </THINKING> <json> { "score_logical_coherence": 1, ... } </json>" </INSTRUCTIONS>

**Strategy: Multi-Judge (Chair Judge)**

<BASE_JUDGMENTS> <JUDGE_1> ... </JUDGE_1> <JUDGE_2> ... </JUDGE_2> <JUDGE_3> ... </JUDGE_3> </BASE_JUDGMENTS>
<INSTRUCTIONS> You are the Chair Judge. You have received evaluations from 3 other judges. Synthesize their insights. If they disagree, look at the Ground Truth to break the tie. Provide a final, consolidated evaluation in <THINKING> and scores in <json>. </INSTRUCTIONS>

**Strategy: Overseer Audit**

<INSTRUCTIONS> Assess whether the JUDGE_OUTPUT contains any logical errors in its evaluation for EACH rubric component. If there is NO thinking chain, mark the entire evaluation as problematic.
For each component, answer if there is a logical error (true/false) and briefly explain why. Errors include: misapplication of the rubric, inconsistent or contradictory reasoning, unsupported claims, overlooking critical evidence from PROBLEM/GT/MODEL_OUTPUT, or internally inconsistent scoring rationale.
Return STRICT JSON inside <json> tags, with this exact schema and keys: { "logical_coherence": "...", "logical_coherence_verdict": <true|false>, "faithfulness_to_task": "...", "faithfulness_to_task_verdict": <true|false>, "methodological_alignment": "...", "methodological_alignment_verdict": <true|false>, "intermediate_correctness": "...", "intermediate_correctness_verdict": <true|false>, "error_awareness": "...", "error_awareness_verdict": <true|false>, "overall_error": <true|false>, "notes": "<optional notes>" } Only output the JSON within <json>...</json> tags. </INSTRUCTIONS>
Rubric the judge was supposed to apply: {RUBRIC_TEXT}
<PROBLEM> {problem["problem_statement"]} </PROBLEM>
<GROUND_TRUTH_SOLUTION> {problem["ground_truth_solution"]} </GROUND_TRUTH_SOLUTION>
<MODEL_OUTPUT> {problem["model_output"]} </MODEL_OUTPUT>

<JUDGE_OUTPUT> {judge_text} </JUDGE_OUTPUT>