CALIBRATING VERBALIZED CONFIDENCE AT ANY OUTPUT TOKEN POSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models often report confidence that does not match their accuracy, and existing methods typically calibrate it at the before-answer (BA) or after-answer (AA) position. This leaves open what confidence denotes at an arbitrary output position and how it should be calibrated. We define the target at token t as the conditional probability of eventual answer correctness given the current prefix; these values form a Doob martingale whose endpoints are BA and AA confidence. We then show empirically that coupling a BA report with answer generation can alter subsequent accuracy, so such reports need not estimate the solver's success probability under a plain reasoning prompt. Position-Agnostic Confidence Elicitation (PACE) therefore decouples the two: it samples and grades solutions once, then trains a confidence readout at randomly selected prefixes with a Brier reward, which we prove is uniquely maximized by the conditional correctness probability for a fixed solver and position-selection rule. We further analyze how group normalization in group relative policy optimization (GRPO) alters this objective. Mean subtraction preserves the expected reward-gradient direction, whereas standardization yields a comparison semi-gradient that, with two samples, induces threshold-type best responses; we characterize larger groups through structural identities and ablations. Across five backbones and seven test sets, PACE matches or outperforms strong baselines designed for one or two positions. Finally, confidence-based routing demonstrates the value of position flexibility: deferring at BA saves more cost than routing on AA confidence.

1 INTRODUCTION

Large language models (LLMs) are used to answer questions ranging from simple factual queries to Olympiad-level mathematics. To assess the reliability of an answer, users often ask the model to report a confidence score. Such verbalized confidence is useful only if calibrated, yet post-training can make LLMs overconfident (Epstein et al., 2025; Leng et al., 2025). Moreover, the position of the score matters. Depending on the prompt, it may be generated before, during, or after the solution. For example, when a response contains `answer`, `rationale`, and `confidence` fields, their generation order determines what information the confidence can reflect: only the question if generated first, or also the reasoning and answer if generated last.

Recent work calibrates verbalized confidence through reinforcement learning (RL) (Damani et al., 2026; Zhang et al., 2026) or supervised fine-tuning (SFT) (Li et al., 2025), but mostly targets after-answer (AA) confidence or treats its position as a formatting choice. CoCA (Li et al., 2026) distinguishes before-answer (BA) confidence, which predicts the probability of answering correctly, from AA confidence, which assesses a specific response. CALIBER (Finlay et al., 2026) makes this dependence on the information state explicit, defining separate supervision targets for the two positions. Both, however, address only the endpoints, leaving open how to define and calibrate confidence at intermediate positions, such as partway through a reasoning trace.

We ask: **can we calibrate verbalized confidence at any output token position?** For a fixed solver, we define the target at token index t as $V_t = \Pr[Z = 1 \mid x, y_{\le t}]$, the conditional probability of eventual answer correctness given the question x and the prefix $y_{\le t}$. These probabilities form a Doob martingale whose endpoints $t = 0$ and $t = T$ are BA prediction and AA assessment. Learning them faces an obstacle: a controlled experiment shows that generating confidence and answer together lets

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.

a BA report alter subsequent correctness, so the report may reflect the outcomes it induces rather than the solver's success probability under a plain reasoning prompt. *Position-Agnostic Confidence Elicitation (PACE)* therefore decouples the two: we sample several solutions per question under a reasoning prompt, grade them, and fix the outcomes, then cut the traces at random positions, append a probe prompt, and train the reports with a Brier reward that we prove is uniquely maximized by the conditional correctness probability for a fixed solver and position-selection rule.

We further analyze how group normalization in group relative policy optimization (GRPO) affects this objective. Subtracting the group mean, without dividing by the group standard deviation, preserves the expected reward-gradient direction; standardization instead yields a comparison semi-gradient whose best responses are of threshold type in a two-sample setting, and structural identities and ablations characterize larger groups.

To summarize, our contributions are: (1) Definition of a calibration target for verbalized confidence at any output token position, with BA and AA confidence as its endpoints. (2) Evidence of a self-fulfilling effect, in which a coupled BA report alters the graded outcome. (3) PACE, a decoupled training method with a provably proper Brier reward, and an analysis of how GRPO's reward transformations shape the calibration incentive. (4) Experiments across backbones, positions, and domains, where PACE matches or outperforms strong baselines, and a routing application showing the cost benefit of BA confidence.

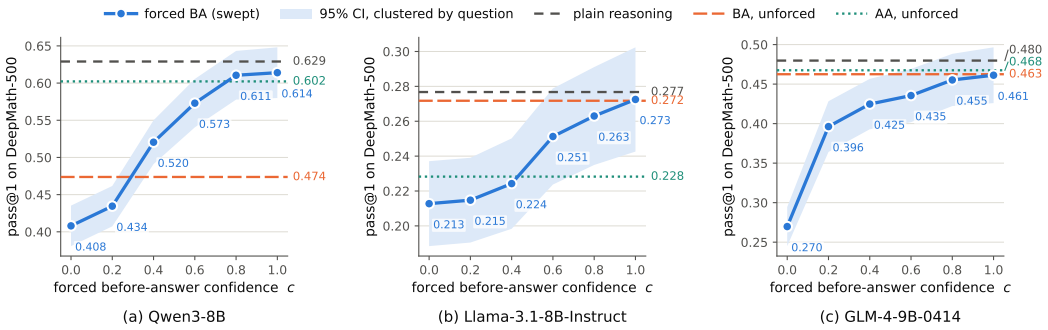
2 Related work

Uncertainty and confidence calibration. Reliable self-assessment is essential for deployment, yet post-trained LLMs often remain poorly calibrated (Leng et al., 2025; Prato et al., 2024; Kapoor et al., 2024). Confidence estimation methods fall into four categories. (1) *Logit-based methods* aggregate token probabilities, which need not reflect correctness and may be unavailable through APIs. (2) *Introspection-based methods* probe hidden states (Azaria & Mitchell, 2023; Radharapu et al., 2025), requiring auxiliary components and access to model internals. (3) *Sampling-based methods* measure consistency across generations (Xiong et al., 2024; Lyu et al., 2025), incurring extra inference cost for long reasoning traces. (4) *Verbalized confidence methods* elicit numerical reports via prompting (Yang et al., 2024; Lin et al., 2022) or train them with SFT (Li et al., 2025) or RL (Stangel et al., 2025; Damani et al., 2026). Most verbalized methods target AA reports or treat position as a formatting choice, although different positions reflect different information states. CoCA (Li et al., 2026) distinguishes BA prediction of success from AA assessment of a specific solution, training BA confidence with segment-specific credit assignment. CALIBER (Finlay et al., 2026) supervises before- and after-reasoning confidence with targets matched to each information state. We extend this view to arbitrary token positions, defining a conditional correctness target with BA and AA confidence as its endpoints and training a decoupled readout across solution prefixes.

Policy-gradient RL algorithms. Policy-gradient methods are widely used in LLM post-training to improve reasoning and align outputs with human preferences. Proximal policy optimization (PPO) (Schulman et al., 2017) constrains updates with a clipped surrogate objective and typically estimates advantages with a learned value function. GRPO replaces this value baseline with within-group reward normalization (Shao et al., 2024; DeepSeek-AI, 2025). Mean subtraction alone yields a scaled leave-one-out REINFORCE estimator (Williams, 1992; Ahmadian et al., 2024), whereas standardization can bias updates (Liu et al., 2025). Bereket & Leskovec (2025) show that standardization induces miscalibration under proper-score rewards, including Brier, and that removing the division mitigates it. Building on this work, we characterize which reward transformations preserve the expected on-policy reward-gradient direction and derive an exact comparison semi-gradient under standardization with two samples. For larger groups, structural identities and ablations clarify the implications and limits of this mechanism for confidence calibration.

3 Target for verbalized confidence at a token position

Let x be a question with reference answer $y^\star$, drawn from a dataset $\mathcal{D}$. Let π_θ denote an LLM with parameters θ. We use the same policy for reasoning and confidence elicitation, with two fixed instructions. The *reasoning instruction* $\mathtt{R}$ asks for a worked solution ("Please reason step by step,



(a) Qwen3-8B (b) Llama-3.1-8B-Instruct (c) GLM-4-9B-0414

Figure 1: **A coupled BA report changes graded accuracy.** The same DeepMath-500 questions, eight samples per condition. Plain, unforced BA, and AA prompts are references.

and output the final answer within `\boxed{}`"), and the *probe instruction* `P` asks for a number ("Do not reason. Output only `\confidence{p}`, p (0.00 $\sim$ 1.00) = probability your final answer will be correct"). Sampling from π_θ under `R` gives a solution $y = (y_1, \ldots, y_T)$ of length T with prefix $y_{\le t}$, and correctness $Z = \mathbb{I}[\text{ans}(y) = y^\star] \in \{0, 1\}$. Let $\mathcal{F}_t = \sigma(x, \texttt{R}, y_{\le t})$ denote the information available at token index t. At a given checkpoint θ, the *target* for verbalized confidence at t is

$$V_t \;:=\; \mathbb{E}[Z \mid \mathcal{F}_t] = \Pr[Z{=}1 \mid \mathcal{F}_t]\,, \qquad p(x) := V_0 = \Pr[Z{=}1 \mid x, \texttt{R}], \qquad \bar{p} := \mathbb{E}_x[p(x)]. \quad (1)$$

We suppress dependence on θ and fixed decoding settings in V_t and $p(x)$ for readability. Thus V_t is the probability that the model completes the observed prefix correctly. Carrying completed traces forward, (V_t) forms a Doob martingale: $\mathbb{E}_{\pi_\theta}[V_{t+1} \mid \mathcal{F}_t] = V_t$ (Williams, 1991). Before answering, $V_0 = p(x)$ is the model's success probability on the question. After answering, $V_T = Z$ is the completed solution's correctness under a deterministic verifier. They are endpoints of the same conditional-probability target. To obtain a verbalized confidence, we make the following definition.

Definition 1 (Probe state and confidence announcement)**.** The *probe state* is $s_t = (x, \texttt{R}, y_{\le t}, \texttt{P})$, obtained by appending the probe instruction to the observed prefix. The *confidence announcement* is $c_t \sim \pi_\theta(\cdot \mid s_t)$, decoded from the emitted numeral.

Selecting a prefix. For training and evaluation, a fixed rule ν may select a position τ from a completed trace. Define $S_\nu = (x, \texttt{R}, y_{\le \tau}, \texttt{P})$ and $V^\nu(S_\nu) = \mathbb{E}_{\pi_\theta, \nu}[Z \mid S_\nu]$. For a deterministic index $\tau = t$, this reduces to V_t. For a relative position $\tau = \lfloor uT \rceil$, $u \in [0, 1]$, selection depends on the eventual length T, so its conditional target need not equal the deterministic-index target (Appendix A.6). During offline confidence training, these expectations refer to the checkpoint that generated the stored solutions before parameter updates.

4 A COUPLED PROMPT CAN CHANGE THE ANSWER IT PREDICTS

Prior work typically elicits confidence and answers jointly via coupled prompts, yet we find that confidence can affect answer correctness, especially for BA reports.

A controlled intervention reveals a self-fulfilling effect. We use a coupled prompt that requests confidence before the reasoning trace and final answer. We prefill the assistant turn with `\confidence{c}` and sample the remainder, sweeping $c \in \{0, 0.2, \ldots, 1\}$. Figure 1 shows that raising the forced confidence from 0 to 1 increases pass@1 by 0.206, 0.060, and 0.192 for Qwen3-8B, Llama-3.1-8B, and GLM-4-9B, respectively. AA prompts change accuracy by -0.027, -0.048, and -0.012 relative to plain reasoning for Qwen3-8B, Llama-3.1-8B, and GLM-4-9B, respectively. These AA changes arise through the overall prompt rather than the later announcement.

Implications for confidence training. These results show that BA confidence can influence the very outcome it is meant to predict. Consequently, training confidence against answers generated after the report need not recover the model's probability of success under a plain reasoning prompt. Even a report that is well calibrated on these coupled outcomes may therefore describe accuracy shaped by the announcement. Section 6.2 reports how the coupled baselines calibrate in practice.

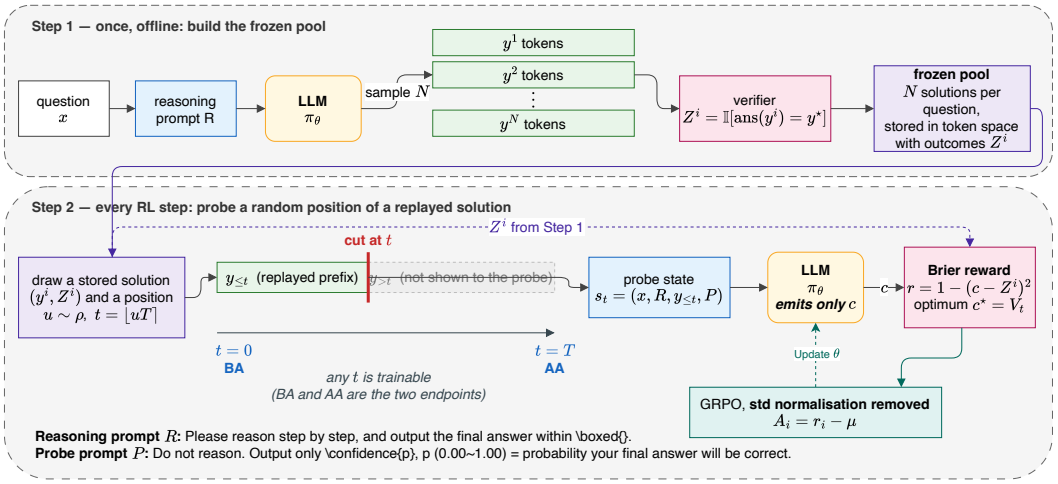


Figure 2: **PACE.** Offline solutions and their outcomes are frozen before confidence training. At each update, sampled prefixes elicit reports scored against stored outcomes.

5 Decoupled Calibration at an Arbitrary Position

The observation in Section 4 motivates decoupling answer generation from confidence training.

Position-Agnostic Confidence Elicitation (PACE). Figure 2 separates reasoning from elicitation. **Step 1**: sample N solutions per question under R, grade them, and freeze their token IDs and outcomes Z^i. **Step 2**: replay a solution, draw $u \sim \text{Uniform}(0, 1)$, set $t = \lfloor uT \rfloor$, and sample $c \sim \pi_\theta(\cdot \mid s_t)$. The Brier reward is

$$r(c, Z) = 1 - (c - Z)^2, \tag{2}$$

plus an auxiliary format term encouraging a parseable numeral in $[0, 1]$. We use GRPO without per-group standard-deviation normalization by default; §6.3.2 compares alternatives.

The stored outcome cannot be changed by the report and supervises every prefix of that solution. Let θ_0 denote the checkpoint that generated the solutions before confidence training. With this data-generating checkpoint and the position-selection rule fixed, the population Brier optimum is the conditional success probability at the selected prefix (Theorem 2).

5.1 The Brier Reward is Exactly Proper

A strictly proper score is uniquely maximized in expectation by the true outcome probability (Gneiting & Raftery, 2007). The following theorem applies this Brier identity to the selected prefix.

Theorem 2 (Exact properness)**.** *Fix the solution-generating checkpoint θ_0 and a θ-independent probe-selection rule ν. All conditional values below refer to solutions generated by π_{θ_0}. For the reward in Equation 2,*

$$\mathbb{E}_{\pi_{\theta_0}, \nu}[r(c, Z) \mid S_\nu] = 1 - \big(c - V^\nu(S_\nu)\big)^2 - V^\nu(S_\nu)\big(1 - V^\nu(S_\nu)\big).$$

Over $c \in [0, 1]$, the maximizer is unique and equals the selected state's conditional value, $c^\star(S_\nu) = V^\nu(S_\nu)$. If ν fixes a deterministic token t, this reduces to $c^\star(s_t) = V_t$ and the attainable ceiling is $R^\star(t) = 1 - V_t(1 - V_t)$.

The proof is in Appendix A.2. The offline target is fixed, but training updates shared parameters and may alter the solver's continuation probabilities; Section 6.2 shows that aggregate reasoning accuracy changes little.

5.2 What Group-Relative Updates Preserve

Consider a group of G announcements with rewards r_i, mean μ, and standard deviation σ. Standard GRPO uses $A_i = (r_i - \mu)/(\sigma + \epsilon)$; our default, *group-mean subtraction*, uses $A_i = r_i - \mu$,

which we show preserves the expected reward-gradient direction. For standardization, an exact characterization at $G = 2$ shows that only reward ordering is retained. These results concern the expected on-policy score-function update and do not cover PPO clipping, KL regularization, finite-sample effects, or auxiliary rewards.

Theorem 3 (Group-mean subtraction preserves the reward-gradient direction)**.** *Suppose the group states and outcomes are fixed before the announcements are sampled, the announcements are conditionally independent given those exogenous variables, and each reward $r_i = r(c_i, Z_i)$ depends on the actions only through its own announcement. Let $\bar{r}_{-i} = \frac{1}{G-1}\sum_{j\neq i} r_j$. Then $r_i - \mu = \frac{G-1}{G}(r_i - \bar{r}_{-i})$. If group members have exchangeable marginals, the unclipped on-policy score-function estimator satisfies*

$$\mathbb{E}\left[\sum_{i=1}^{G}(r_i - \mu)\nabla_\theta \log \pi_\theta(c_i \mid s_i)\right] = (G-1)\nabla_\theta \mathbb{E}[r(c, Z)].$$

Thus subtracting the group mean preserves the expected reward-gradient direction. Applied to the Brier component alone, it preserves its proper optimum.

Theorem 4 (Two-sample standardization retains only reward ordering)**.** *For $G = 2$, use population standard deviation, set $\epsilon = 0$, and assign zero advantage to tied rewards. Then*

$$\mu = \frac{r_1 + r_2}{2}, \qquad \sigma = \frac{|r_1 - r_2|}{2}, \qquad A_1 = \operatorname{sign}(r_1 - r_2), \quad A_2 = -A_1.$$

Thus the standardized advantage depends only on the reward ordering, not the reward gap.

By Theorem 4, standardization cancels the reward difference, so small and large improvements receive the same advantage. For Brier rewards, this removes the incentive to report the true probability.

Corollary 5 (Expected advantage favors extreme reports at one state)**.** *Use the two-sample normalization in Theorem 4 and the Brier reward. At one fixed probe state, let both outcomes have success probability V, and sample the two reports independently from π_θ, independently of the outcome pair. For a first report c, with the second report's distribution held fixed,*

$$\mathbb{E}[A_1 \mid c_1 = c] = (2V - 1)\big(\Pr[c_2 < c] - \Pr[c_2 > c]\big).$$

This expected advantage is nondecreasing in c when $V > 1/2$ and nonincreasing when $V < 1/2$. Consequently, $c = 1$ or $c = 0$, respectively, is a maximizer over $[0, 1]$. When $V = 1/2$, every report has zero expected advantage.

The corollary follows by comparing outcome pairs: the larger report wins if both answers are correct, the smaller wins if both are incorrect, and the two equiprobable mixed cases cancel. Proofs and the full characterization are in Appendices A.3–A.5.

Beyond $G = 2$ and GRPO. For $\epsilon = 0$ and nonzero group variance under the population convention, $\sum_i A_i = 0$ and $\sum_i A_i^2 = G$: standardization preserves the centered reward vector's direction but discards its length. For $G \geq 3$, this direction retains ratios of reward differences, so no comparison-only reduction generally follows. Appendix A.7 discusses this in detail. More generally, REINFORCE and action-independent baselines such as RLOO likewise preserve the expected reward-gradient direction under the stated assumptions, so applied to the Brier reward they retain its optimum; §6.3.2 compares optimizers.

5.3 Moment diagnostics across positions

The martingale formulation provides moment diagnostics. With the information $\mathcal{F}_t$ defined in Section 3, reports that equal their conditional correctness targets satisfy the following identities.

Proposition 6 (Position diagnostics)**.** *Suppose $c_t = \mathbb{E}[Z \mid \mathcal{F}_t]$ and $\sigma(x) \subseteq \mathcal{F}_t$ at every token position t. Assume that $\mathcal{F}_0 = \sigma(x)$ and that the completed solution determines Z, giving the endpoint reports $c_0 = p(x)$ and $c_T = Z$. Then (i) $\mathbb{E}[c_t] = \bar{p}$; (ii) $\operatorname{Var}(c_t) = \operatorname{Var}_x(p(x)) + \mathbb{E}_x[\operatorname{Var}(c_t \mid x)]$, whose between-question term is independent of t; (iii) for $t_1 < t_2$ with $\mathcal{F}_{t_1} \subseteq \mathcal{F}_{t_2}$, $\mathbb{E}_x[\operatorname{Var}(c_{t_2} \mid x)] \geq \mathbb{E}_x[\operatorname{Var}(c_{t_1} \mid x)]$; and (iv) the within-question variance term is zero at BA and equals $\mathbb{E}_x[p(x)(1 - p(x))]$ at AA, where the total variance is $\bar{p}(1 - \bar{p})$.*

Table 1: **Confidence calibration of different methods on Qwen3-8B.** OOD-Math (MATH-500, GSM8K, OlympiadBench) and OOD-NonMath (CommonsenseQA, GPQA-Diamond, LiveCodeBench) are macro averages; * marks our methods. ConfTuner's two blocks are separate fine-tuned models, each with its own pass@1. Per-dataset results are in Appendix C.

	DeepMath-500				OOD-Math				OOD-NonMath			
method	pass@1	ECE ↓	AUROC ↑	Brier ↓	pass@1	ECE ↓	AUROC ↑	Brier ↓	pass@1	ECE ↓	AUROC ↑	Brier ↓
$t = 0$												
Base-AvgProb	0.634	0.114	0.472	0.242	0.772	0.265	0.441	0.240	0.630	0.181	0.500	0.268
Base-ProbeMLP	0.634	0.076	**0.761**	0.194	0.772	0.065	0.784	0.117	0.630	0.154	0.569	0.237
Base-Verb	0.634	0.347	0.558	0.352	0.772	0.201	0.628	0.211	0.630	0.339	0.592	0.345
Base-ConfTuner	0.645	0.049	0.755	0.188	0.769	**0.051**	**0.796**	0.117	0.640	0.126	0.644	0.213
Base-PACE-$u^\star$	0.640	0.076	0.746	0.195	0.775	0.077	0.777	0.121	0.642	0.092	0.629	0.207
RLVR-AvgProb	0.801	0.218	0.468	0.213	0.826	0.290	0.434	0.231	0.670	0.179	0.495	0.262
RLVR-ProbeMLP	0.801	0.079	0.750	0.145	0.826	0.102	0.775	0.123	0.670	0.254	0.589	0.280
RLVR-Verb	0.801	0.198	0.604	0.200	0.826	0.160	0.600	0.166	0.670	0.305	0.588	0.311
RLVR-ConfTuner	0.797	**0.035**	0.751	0.138	0.825	0.052	0.785	0.102	0.674	0.153	0.663	0.215
Base-CoCA	0.784	0.211	0.550	0.209	0.823	0.168	0.534	0.165	0.650	0.286	0.658	0.288
Base-CoCA (std off)	0.795	0.041	0.684	0.153	0.830	0.097	0.704	0.118	0.672	0.160	0.632	0.222
Base-CALIBER	0.760	0.105	0.711	0.174	0.807	0.099	0.769	0.113	0.665	**0.073**	**0.688**	**0.184**
RLVR-PACE-$u^\star$	0.811	0.038	0.751	**0.132**	0.833	0.054	0.795	**0.099**	0.677	0.086	0.679	0.192
$t = T$												
Base-AvgProb	0.634	0.252	0.657	0.287	0.772	0.143	0.770	0.179	0.630	0.219	0.678	0.271
Base-ProbeMLP	0.634	0.213	0.785	0.232	0.772	0.139	0.775	0.142	0.630	0.257	0.623	0.276
Base-Verb	0.634	0.367	0.554	0.365	0.772	0.228	0.538	0.227	0.630	0.354	0.615	0.354
Base-ConfTuner	0.699	0.035	0.839	0.144	0.793	0.062	0.839	0.097	0.653	0.170	0.722	0.216
Base-PACE-$u^\star$	0.640	0.097	0.805	0.178	0.775	0.089	0.832	0.107	0.642	0.110	0.702	0.197
RLVR-AvgProb	0.801	0.165	0.683	0.184	0.826	0.149	0.788	0.159	0.670	0.264	0.719	0.285
RLVR-ProbeMLP	0.801	0.100	0.794	0.153	0.826	0.076	0.817	0.104	0.670	0.148	0.696	0.219
RLVR-Verb	0.801	0.197	0.541	0.197	0.826	0.174	0.528	0.174	0.670	0.315	0.622	0.315
RLVR-ConfTuner	0.802	**0.032**	0.845	0.113	0.835	**0.046**	0.856	**0.078**	0.677	0.096	**0.772**	0.174
Base-RLCR	0.797	0.179	0.693	0.182	0.824	0.140	0.687	0.153	0.665	0.277	0.630	0.291
Base-RLCC-L1	0.775	0.217	0.554	0.215	0.816	0.175	0.592	0.172	0.663	0.301	0.632	0.305
Base-CALIBER	0.760	0.076	0.783	0.153	0.807	0.107	0.773	0.113	0.665	0.108	0.735	0.191
RLVR-PACE-$u^\star$	0.811	0.055	**0.869**	**0.107**	0.833	0.057	**0.885**	0.079	0.677	**0.084**	0.770	**0.173**

6 EXPERIMENTS

6.1 SETUP

Models and data. We use Qwen3-8B (Yang et al., 2025) as the main backbone and additionally evaluate Qwen3-1.7B/32B, GLM-4-9B-0414 (Team GLM, 2024), and Llama-3.1-8B-Instruct (Team, 2024). Base-X denotes the released checkpoint, and RLVR-X the checkpoint after 100 steps of reinforcement learning with verifiable rewards (RLVR). We train on DeepMath-16K, a random 16K-question subset of the decontaminated mathematics dataset DeepMath-103K (He et al., 2025). We evaluate on DeepMath-500 (500 held-out DeepMath-103K questions disjoint from training), MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), OlympiadBench (He et al., 2024), CommonsenseQA (Talmor et al., 2019), GPQA-Diamond (Rein et al., 2023), and LiveCodeBench (Jain et al., 2025).

Training and metrics. PACE uses $N = 32$, $G = 8$, a batch size of 512, and GRPO with group-mean subtraction. Based on reward convergence, we evaluate Base-PACE at step 100 and RLVR-PACE at step 230. Training and evaluation temperatures are 1.0 and 0.6, respectively. We report ECE, AUROC, and Brier score, averaged over eight evaluation permutations with question-clustered bootstrap standard errors. Appendix B gives configurations and metric definitions.

Baselines. We compare against *AvgProb*, the mean token probability over the question tokens at $t = 0$ and the model's own answer tokens at $t = T$; *ProbeMLP*, an MLP on a frozen hidden state fitted separately per position; *Verb*, our probe prompt on an untrained checkpoint at both endpoints; and *ConfTuner* (Li et al., 2025), a tokenized Brier score at a single slot — answer-first as published, hence $t = T$, with the $t = 0$ rows a variant we retrain on a BA prompt. *Coupled RL* methods emit confidence inside the answer rollout, so their format fixes the position: *RLCR* (Damani et al., 2026) and *RLCC-L1* (Zhang et al., 2026) are AA ($t = T$), *CoCA* (Li et al., 2026) is confidence-first ($t = 0$), reported as published and a variant with the standard-deviation division removed from its confidence segment, and *CALIBER* (Finlay et al., 2026) reports both endpoints through two dedicated slots. Each is evaluated at the endpoints its format supports; configurations are in Appendix B.

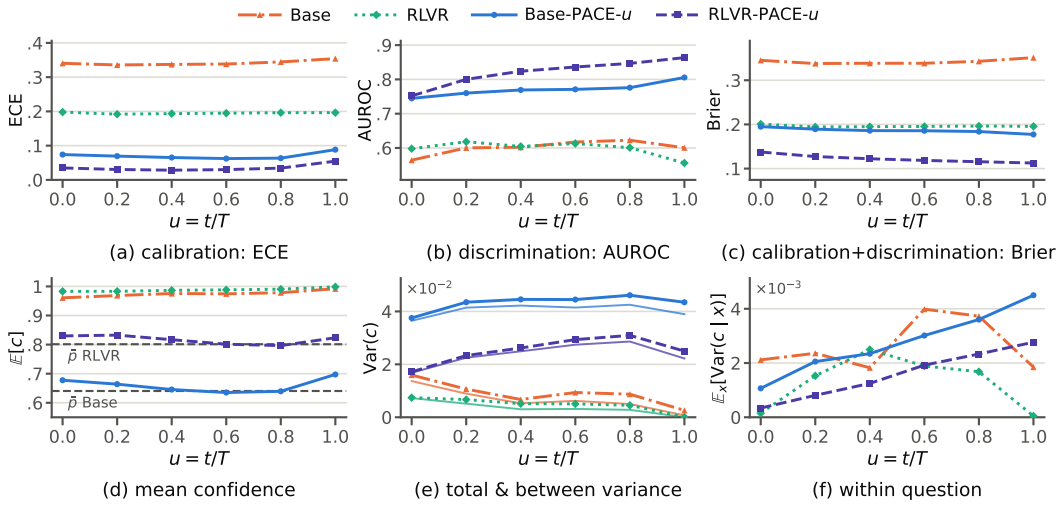


Figure 3: **Calibration performance at different token positions on DeepMath-500.**

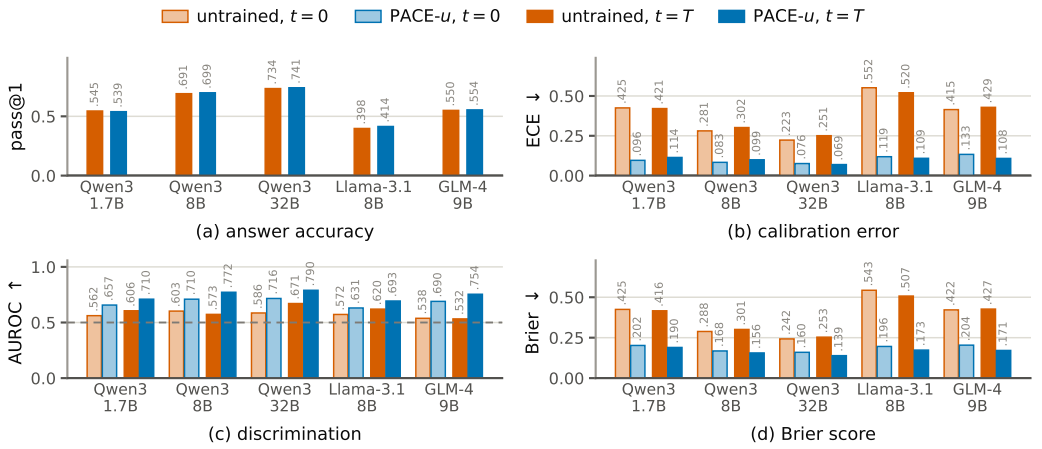


Figure 4: **PACE across backbones.** Seven-dataset macro averages, per-dataset in Appendix E.

6.2 Main results

PACE achieves strong calibration at both endpoints in a single run. Since some baselines also improve reasoning, we include an accuracy-matched RLVR initialization. In Table 1, PACE improves over the untrained verbalizer for both initializations, at both endpoints, and in all dataset groups. Across methods, PACE matches or outperforms the strongest baselines: RLVR-PACE holds seven of the 18 best confidence-metric results, as many as both ConfTuner arms combined, and unlike ConfTuner, which needs a separate training run per position, calibrates both endpoints in one run. CoCA's two rows show that both its coupled BA prompt and standard-deviation division harm calibration: even without the division, CoCA trails decoupled PACE, consistent with Section 4.

Confidence calibration only slightly changes reasoning. PACE improves pass@1 by at most 0.012 in all dataset groups and for both initializations, and per-question pass rates correlate with the initialization at macro $r \approx 0.93$, at or within 2% of the ceiling (Appendix Table 11).

With PACE, calibration holds across positions and discrimination grows with position. In Figure 3(a)–(c), PACE's ECE stays low throughout (at most 0.088), AUROC rises with position (0.752 → 0.863 for RLVR-PACE) and Brier falls, consistent with increasingly informative prefixes. The untrained arms are also flat in ECE but at five to six times the error, announcing a near-constant 0.96–0.99. Their AUROC likewise fails to grow: it stays within 0.56–0.63 and shifts by at most 0.04 between the endpoints, so a near-constant report gains nothing from a longer prefix.

Table 2: **Optimizer and group-size ablations.** Macro averages over seven test sets. *Extremal %* is the share of reports with $c \leq 0.05$ or $c \geq 0.95$. Per-dataset results are in Appendices F and G.

			t=0			t=T			extremal %	
objective	G	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	t=0	t=T
RLOO	8	0.694	**0.082**	**0.716**	**0.168**	**0.080**	**0.788**	**0.149**	27.2	35.2
REINFORCE++	8	0.697	0.112	0.715	0.175	0.110	0.777	0.158	24.8	26.0
PPO	8	0.699	0.102	0.714	0.173	0.097	0.770	0.158	11.6	26.5
GRPO, std on	2	0.695	0.275	0.598	0.278	0.200	0.671	0.208	97.5	93.9
GRPO, std on	8	0.704	0.176	0.642	0.205	0.154	0.711	0.176	62.2	69.0
GRPO, std off*	8	0.699	0.083	0.710	**0.168**	0.099	0.772	0.156	31.7	40.3
GRPO, std off	2	0.694	**0.076**	**0.716**	**0.167**	0.081	**0.782**	**0.154**	23.8	30.0
GRPO, std off	4	0.698	0.078	0.699	0.173	**0.079**	0.769	0.156	30.3	31.3
GRPO, std off	16	0.698	0.080	0.699	0.174	0.111	0.754	0.160	31.9	45.1

The moments partly agree with Proposition 6, but under-dispersion remains. For clause (i), panel (d) shows that both PACE arms track their pool's base rate $\bar{p}$ (within 0.057), whereas the untrained arms lie 0.2–0.35 above it. For clause (ii), panel (e) shows that nearly all variance lies between questions, though only approximately position-independently; clause (ii) holds exactly only for the true conditional expectation, so we read the residual as the gap from that ideal. Panel (f) shows that within-question variance grows with position for PACE but shrinks for the untrained arms. Still, the readout is under-dispersed: at $t = T$, within-question variance reaches only about 5% of its benchmark, between-question variance about 30%, and total variance about 20% of $\bar{p}(1 - \bar{p})$. Nevertheless, the rising AUROC in panel (b) indicates useful ranking despite incomplete recovery of the conditional value.

6.3 Ablation studies

6.3.1 PACE generalizes across scales and model families

On Qwen3-1.7B, Qwen3-32B, Llama-3.1-8B, and GLM-4-9B, Figure 4 shows PACE improving macro ECE, AUROC, and Brier over the untrained counterpart at both endpoints for every model. Within the Qwen family, BA ECE falls and AA AUROC rises with scale ($0.710 \to 0.772 \to 0.790$ from 1.7B to 8B to 32B). For Llama-3.1-8B and GLM-4-9B, PACE reduces BA ECE by 0.433 and 0.282, respectively.

6.3.2 Standardization harms calibration

Table 2 compares RL algorithms for PACE. **Baseline-subtracting objectives perform similarly.** GRPO with group-mean subtraction, RLOO, REINFORCE++, and PPO achieve similar macro AUROC and Brier, with RLOO slightly ahead. In the ideal analysis, RLOO's update is proportional to group-mean subtraction, and PPO's critic acts as an action-independent baseline (clipping and critic error aside); REINFORCE++'s batch scaling falls outside Theorem 3 (Appendix A.8). **Standardization harms calibration.** At $G = 8$, it raises ECE (e.g., $0.083 \to 0.176$ at $t = 0$), lowers AUROC by about 0.06–0.07, and nearly doubles the share of extremal reports, the largest at that group size. At $G = 2$, where Theorem 4 is exact, reports collapse to the extremes and degradation is larger.

6.3.3 Group size has a small effect on the macro average

Varying G from 2 to 16 changes macro Brier by at most 0.007 and AUROC by 0.028 (Table 2). On point estimates G=2 attains the best AUROC at both endpoints, and doubling G from 8 to 16 costs $1.72\times$ per step without improving either metric; we keep G=8 for a lower-variance group baseline.

6.3.4 Early-position training transfers to later positions, but not vice versa

Table 3 compares token-position sampling rules: single positions $u = 0, \frac{1}{2}, 1$ and discrete grids grid_2, grid_3, and grid_9, each trained separately and evaluated at $t = 0, \frac{T}{2}, T$. Transfer across positions is asymmetric: training at $u = 0$ or $\frac{1}{2}$ carries over to later positions, whereas training at $u = 1$ markedly degrades earlier ones. Continuous uniform sampling is competitive with the discrete rules while covering all positions, so we adopt it as our default.

Table 3: **Token position sampling rules.** grid_N samples u uniformly from $\{\frac{i}{N-1}\}_{i=0}^{N-1}$. Cells are macro averages over seven test sets; per-dataset results are in Appendix H.

		t=0			$t \approx T/2$			t=T		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
u=0	0.691	0.083	0.712	0.171	0.094	0.714	0.172	0.155	0.734	0.187
u=$\frac{1}{2}$	0.696	0.096	**0.725**	0.170	0.099	**0.763**	0.162	0.094	0.789	0.152
u=1	0.706	0.211	0.700	0.216	0.253	0.694	0.238	0.096	**0.806**	**0.146**
grid_2	0.700	0.128	0.724	0.180	0.159	0.740	0.186	0.105	0.791	0.152
grid_3	0.698	0.086	0.717	0.167	0.088	0.739	0.160	0.084	0.767	0.152
grid_9	0.697	0.084	0.720	**0.165**	**0.080**	0.744	**0.158**	**0.075**	0.786	0.148
$U[0,1]^\star$	0.699	**0.083**	0.710	0.168	0.096	0.748	0.165	0.099	0.772	0.156

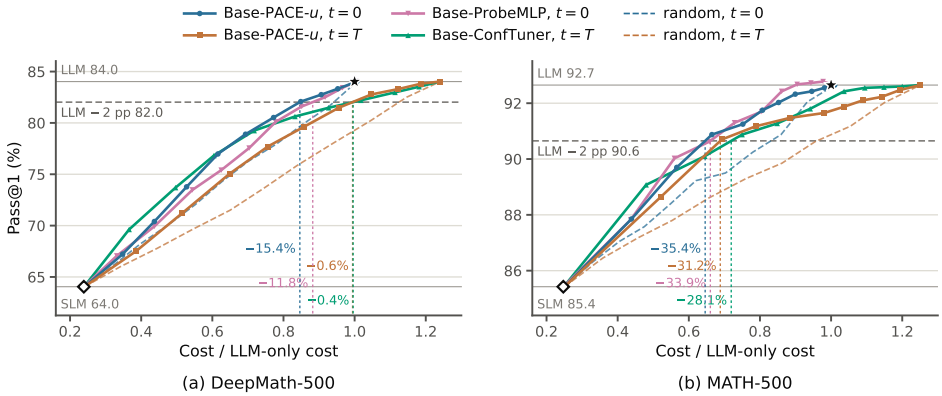


Figure 5: **Routing accuracy versus normalized cost on DeepMath-500 and MATH-500.**

6.4 Routing application

We use calibrated Qwen3-8B as the small language model (SLM) and RLVR-trained Qwen3-32B as the LLM, deferring a question to the LLM whenever the SLM's confidence satisfies $c < \alpha$. For prompt length n_{in} and answer lengths n_s, n_l, the weighted-token costs are $S = n_{\text{in}} + 4n_s$ for an SLM answer and $L = 4n_{\text{in}} + 16n_l$ for an LLM answer, reflecting a $4\times$ model-size ratio and a $4\times$ output-to-input token price ratio. With KV-cache reuse and a readout cost of $P = \delta + 32$, where δ is the number of instruction tokens, the cascade costs are

$$C_0(\alpha) \;=\; n_{\text{in}} + P \;+\; \mathbb{I}[c \geq \alpha]\, 4n_s \;+\; \mathbb{I}[c < \alpha]\, L, \qquad C_T(\alpha) \;=\; S + P \;+\; \mathbb{I}[c < \alpha]\, L. \quad (3)$$

Figure 5 plots accuracy against LLM-normalized cost as α sweeps from 0 to 1.1, comparing PACE-u with random routing and with the strongest-AUROC baselines at each position (Table 1): Base-ProbeMLP at $t = 0$ and Base-ConfTuner at $t = T$. Confidence-based routing generally outperforms random routing. At $t = T$, ConfTuner is competitive with PACE, but routing at $t = 0$ with PACE-u also avoids the SLM's reasoning cost, yielding larger savings when accuracy must stay close to the LLM's. At a pass@1 two points below the LLM's, Base-PACE-u at $t = 0$ reduces cost by 15.4% on DeepMath-500, compared with 0.4% for Base-ConfTuner at $t = T$. On the easier MATH-500, where the SLM confidently solves more questions, the savings are 35.4% and 28.1%, respectively.

7 Conclusion

We introduced PACE, which calibrates verbalized confidence at any output token position. To avoid the self-fulfilling effect of coupled BA prompts, PACE decouples confidence training from answer generation and trains with a Brier reward that is proper for a fixed solver and position-selection rule. Our analysis of GRPO shows that group-mean subtraction preserves the expected reward-gradient direction, whereas standardization with two samples reduces the incentive to a pairwise comparison. Across five backbones and seven test sets, a single PACE run improves calibration across positions and matches or outperforms the strongest baselines, and confidence-based routing shows that BA confidence saves more cost than AA confidence, highlighting the value of position flexibility.

AI USE STATEMENT

We used generative AI tools to assist with proof writing, design or provide feedback on research methodology and experiments, implement methods, clean and reformat datasets, support qualitative and thematic data analysis, and interpret results. We did not use these tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for mathematical proofs, or propose or refine hypotheses. Synthetic data generation and translation were not applicable to this work. We also used generative AI tools to create or modify scientific figures and images, create or edit software code, summarize or analyze existing literature, improve manuscript readability, identify relevant literature, and format references. We reviewed all AI-assisted work. All AI-assisted proofs were checked by three authors, and all LLM-generated code was verified and tested by at least two authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

The authors acknowledge and adhere to the ICLR Code of Ethics. This work uses only publicly released models and benchmark datasets in accordance with their licenses. It involves neither human subjects nor personally identifiable data.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed experimental settings, including hyperparameters, prompts, pool construction, parsing, grading, and checkpoint selection, in Appendix B, along with code for all algorithms in the supplementary material. Appendices C–I also report detailed per-dataset results to facilitate comparison.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Amos Azaria and Tom M. Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 967–976. Association for Computational Linguistics, 2023.

Michael Bereket and Jure Leskovec. Uncalibrated reasoning: Grpo induces overconfidence for stochastic outcomes, 2025. URL https://arxiv.org/abs/2508.11800.

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

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Elliot L. Epstein, John Winnicki, Thanawat Sornwanee, and Rajat Vadiraj Dwaraknath. Llms are overconfident: Evaluating confidence interval calibration with fermieval. *CoRR*, abs/2510.26995, 2025. doi: 10.48550/ARXIV.2510.26995. URL https://doi.org/10.48550/arXiv.2510.26995.

Conor Finlay, Joshua Kurien, Saurabh Dash, Marzieh Fadaee, and Beyza Ermis. Caliber: Calibrating confidence before and after reasoning in language models, 2026. URL https://arxiv.org/abs/2606.24281.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *CoRR*, abs/2504.11456, 2025. doi: 10.48550/ARXIV.2504.11456. URL https://doi.org/10.48550/arXiv.2504.11456.

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 Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Hugging Face. Math-verify. https://github.com/huggingface/Math-Verify, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katie Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in llms: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=l0tg0jzsdL.

Changcheng Li, Jiancan Wu, Hengheng Zhang, Zhengsu Chen, Guo An, Junxiang Qiu, Xiang Wang, and Qi Tian. Confidence before answering: A paradigm shift for efficient llm uncertainty estimation, 2026. URL https://arxiv.org/abs/2603.05881.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. Conftuner: Training large language models to express their confidence verbally. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VZQ04Ojhu5.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022, 2022.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. Calibrating large language models with sample consistency. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 19260–19268. AAAI Press, 2025. doi: 10.1609/AAAI.V39I18.34120. URL https://doi.org/10.1609/aaai.v39i18.34120.

Gabriele Prato, Jerry Huang, Prasanna Parthasarathi, Shagun Sodhani, and Sarath Chandar. Do large language models know how much they know? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 6054–6070. Association for Computational Linguistics, 2024.

Bhaktipriya Radharapu, Eshika Saxena, Kenneth Li, Chenxi Whitehouse, Adina Williams, and Nicola Cancedda. Calibrating LLM judges: Linear probes for fast and reliable uncertainty estimation. *CoRR*, abs/2512.22245, 2025. doi: 10.48550/ARXIV.2512.22245. URL https://doi.org/10.48550/arXiv.2512.22245.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023. doi: 10.48550/ARXIV.2311.12022. URL https://doi.org/10.48550/arXiv.2311.12022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models, 2025. URL https://arxiv.org/abs/2503.02623.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Team GLM. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

David Williams. *Probability with Martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, 1991.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for llms. *CoRR*, abs/2412.14737, 2024. doi: 10.48550/ARXIV.2412.14737. URL https://doi.org/10.48550/arXiv.2412.14737.

Chuang Zhang, Zizhen Zhu, Yihao Wei, Bing Tian, Junyi Liu, Henan Wang, Wang Xavier, and Yaxiao Liu. Confidence-calibrated small-large language model collaboration for cost-efficient reasoning. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2026 - Volume 1: Long Papers, Rabat, Morocco, March 24-29, 2026*, pp. 4480–4501. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.eacl-long.208/.

A PROOFS

A.1 PROPER SCORING RULES

Proper scoring rules are reward functions that evaluate the quality of probabilistic predictions by comparing predicted probabilities to observed outcomes. A scoring rule $r(c, Z)$ maps a reported probability c and the realized outcome $Z \in \{0, 1\}$ to a numerical score.

Definition 7 (Proper scoring rule)**.** A scoring rule $r(c, Z)$ is *proper* if the expected score is maximized when the reported probability equals the true probability, i.e., for all $c, V \in [0, 1]$:

$$V \cdot r(V, 1) + (1 - V) \cdot r(V, 0) \geq V \cdot r(c, 1) + (1 - V) \cdot r(c, 0), \tag{4}$$

where V denotes the true probability. It is *strictly proper* if equality holds only when $c = V$.

Properness guarantees that a rational agent has no incentive to misreport its belief. Classical strictly proper rules include the Brier reward $r(c, Z) = 1 - (c - Z)^2$ of Equation 2 and the logarithmic score $r(c, Z) = Z \log c + (1 - Z) \log(1 - c)$ (Gneiting & Raftery, 2007). In the context of LLM confidence calibration, proper scoring rules provide a principled foundation for designing reward functions: if the reward is proper with respect to the model's confidence output and its correctness, optimizing this reward encourages the model to output confidence scores that faithfully reflect its true probability of being correct.

A.2 PROOF OF THEOREM 2

Because $Z \in \{0, 1\}$ we have $Z^2 = Z$, so under the solution distribution generated by π_{θ_0} and the fixed probe-selection rule, conditionally on S_ν

$$\begin{aligned}
\mathbb{E}\big[(c - Z)^2 \mid S_\nu\big] &= c^2 - 2c\,\mathbb{E}[Z \mid S_\nu] + \mathbb{E}[Z^2 \mid S_\nu] \\
&= c^2 - 2cV^\nu(S_\nu) + V^\nu(S_\nu) \\
&= \big(c - V^\nu(S_\nu)\big)^2 + V^\nu(S_\nu)\big(1 - V^\nu(S_\nu)\big),
\end{aligned} \tag{5}$$

Substituting into Equation 2 gives the identity. The only c-dependent term is $-\big(c - V^\nu(S_\nu)\big)^2$, strictly concave and maximized uniquely at $c = V^\nu(S_\nu)$. For deterministic ν this is V_t and gives the stated ceiling. No group statistic appears in this calculation. □

Remark 8 (Where decoupling enters). The conditional value $V^\nu(S_\nu)$ is fixed during confidence training: the solutions predate the parameter updates, ν is independent of the trainable θ, the probe instruction adds no outcome information beyond the selected state, and sampling c cannot alter the stored Z. A rule ν that depends on the completed trace, as relative position does, changes the estimand to the induced conditional value of §3. Under coupling the outcome distribution depends on c, because the announcement is part of the context from which the answer is generated, so properness no longer guarantees recovery of the plain-prompt success probability (§4).

A.3 PROOF OF THEOREM 3

Split the group mean into the own term and the rest: $r_i - \mu = r_i - \frac{1}{G} r_i - \frac{1}{G} \sum_{j \neq i} r_j = \frac{G-1}{G}(r_i - \bar{r}_{-i})$. Conditional on the exogenous group states, outcomes and sibling actions, $\bar{r}_{-i}$ does not depend on c_i. The conditional score-function identity gives

$$\mathbb{E}[\bar{r}_{-i} \nabla_\theta \log \pi_\theta(c_i \mid s_i) \mid s_i, \bar{r}_{-i}] = 0,$$

so the surviving term is plain REINFORCE:

$$\mathbb{E}\left[\sum_i (r_i - \mu) \nabla_\theta \log \pi_\theta(c_i \mid s_i)\right] = \frac{G - 1}{G} \sum_i \mathbb{E}[r_i \nabla_\theta \log \pi_\theta(c_i \mid s_i)]\,.$$

Exchangeable marginals make the G expectations equal, yielding $(G-1)\nabla_\theta \mathbb{E}[r]$. This is an equality for the on-policy score-function term, not for clipping, KL penalties, or other components of the full optimizer. □

A.4 Proof of Theorem 4

For two rewards, $\mu = (r_1 + r_2)/2$, so $r_1 - \mu = (r_1 - r_2)/2$ and $r_2 - \mu = -(r_1 - r_2)/2$. The population variance is

$$\sigma^2 = \frac{(r_1 - \mu)^2 + (r_2 - \mu)^2}{2} = \frac{(r_1 - r_2)^2}{4}.$$

For unequal rewards, division therefore gives $A_1 = (r_1 - r_2)/|r_1 - r_2| = \operatorname{sign}(r_1 - r_2)$ and $A_2 = -A_1$. For equal rewards, the zero-advantage convention agrees with $\operatorname{sign}(0) = 0$. □

Other normalization conventions. With unbiased sample standard deviation, $\sigma = |r_1 - r_2|/\sqrt{2}$, so both advantages are multiplied by $1/\sqrt{2}$ and still depend only on reward ordering. With population standard deviation and $\epsilon > 0$, the exact expression instead is $A_1 = (r_1 - r_2)/(|r_1 - r_2| + 2\epsilon)$: the comparison-only identity assumes $\epsilon = 0$. Invariance to a common positive affine rescaling of the rewards holds at every group size, and is stated as part (iii) of Theorem 11.

The full expected-gradient characterization. The reward-ordering identity also yields a comparison semi-gradient for the expected update, now allowing the two probe states to differ. Let $(s_i, Z_i)_{i=1}^2$ be exchangeable state–outcome pairs with a θ-independent law, sample c_i independently from $\pi_\theta(\cdot \mid s_i)$ given those pairs, keep $\epsilon = 0$ and the tie convention, and write $\kappa = 1$ for population and $\kappa = 1/\sqrt{2}$ for unbiased sample standard deviation. The expected score-function update satisfies

$$g_{\mathrm{std}}(\theta) := \mathbb{E}\left[\sum_{i=1}^{2} A_i \nabla_\theta \log \pi_\theta(c_i \mid s_i)\right] = 2\kappa\, \mathbb{E}[\operatorname{sign}(r_1 - r_2) \nabla_\theta \log \pi_\theta(c_1 \mid s_1)],$$

by $A_2 = -A_1$, exchangeability, and relabeling the two samples. To express the update as a derivative, hold the second report's policy $\bar{\pi}$ fixed and define

$$W_{\bar{\pi}}(s, Z, c) := \mathbb{E}[\operatorname{sign}(r(c, Z) - r(c', Z')) \mid s, Z, c],$$

with (s', Z') following the second pair's conditional law given (s, Z) and $c' \sim \bar{\pi}(\cdot \mid s')$. Conditioning on the first pair and applying the score-function identity gives

$$g_{\mathrm{std}}(\theta) = 2\kappa\, \nabla_\theta \mathbb{E}_{(s,Z),\, c \sim \pi_\theta(\cdot|s)}[W_{\bar{\pi}}(s, Z, c)]\big|_{\bar{\pi} = \mathrm{sg}(\pi_\theta)}. \tag{6}$$

Only the first report's policy is differentiated; differentiating both occurrences in the antisymmetric payoff would give cancellation, not the GRPO update. The threshold behavior in Corollary 5 instead requires a common state.

A.5 Proof of Corollary 5

Work conditionally on the common probe state and fix the two reports at $c_1 = c$ and $c_2 = c'$. By Theorem 4, the first advantage is the sign of the reward difference. For the Brier reward,

$$\operatorname{sign}\big(r(c, Z_1) - r(c', Z_2)\big) = \begin{cases} \operatorname{sign}(c - c'), & (Z_1, Z_2) = (1, 1), \\ -\operatorname{sign}(c - c'), & (Z_1, Z_2) = (0, 0), \\ \operatorname{sign}(c + c' - 1), & (Z_1, Z_2) = (1, 0), \\ -\operatorname{sign}(c + c' - 1), & (Z_1, Z_2) = (0, 1). \end{cases}$$

Let $p_{ij} = \Pr[Z_1 = i, Z_2 = j]$. Equal marginal success probabilities imply $p_{10} = p_{01}$ and $p_{11} - p_{00} = 2V - 1$, without requiring independent outcomes. Since the reports are independent of the outcome pair, averaging the four cases cancels the mixed-outcome terms:

$$\mathbb{E}[A_1 \mid c_1 = c, c_2 = c'] = (2V - 1) \operatorname{sign}(c - c').$$

Averaging over the second report then gives

$$\mathbb{E}[A_1 \mid c_1 = c] = (2V - 1)\mathbb{E}[\operatorname{sign}(c - c_2)] = (2V - 1)\big(\Pr[c_2 < c] - \Pr[c_2 > c]\big),$$

where the expression in parentheses is nondecreasing in c and ties contribute zero, so multiplying by $2V - 1$ gives the three cases in the corollary. The endpoint maximizers need not be unique, so the result does not assert that every report is driven to an endpoint. □

Writing $D_{\bar{\pi}}(c) = \Pr_{c' \sim \bar{\pi}}[c' < c] - \Pr_{c' \sim \bar{\pi}}[c' > c]$ for a fixed second-report distribution $\bar{\pi}$, the identity is $(2V - 1)D_{\bar{\pi}}(c)$; this is the notation used in Appendix A.7, and holding $\bar{\pi}$ fixed describes the incentive facing one report at the current update rather than differentiation through both.

A.6 Proof of Proposition 6

Let $c_t = \mathbb{E}[Z \mid \mathcal{F}_t]$ with $\sigma(x) \subseteq \mathcal{F}_t$. The tower property gives $\mathbb{E}[c_t] = \mathbb{E}[Z] = \bar{p}$, proving (i), and also $\mathbb{E}[c_t \mid x] = \mathbb{E}[Z \mid x] = p(x)$. The law of total variance then gives

$$\mathrm{Var}(c_t) = \mathrm{Var}_x(p(x)) + \mathbb{E}_x[\mathrm{Var}(c_t \mid x)],$$

proving (ii).

For (iii), consider token positions $t_1 < t_2$ with $\mathcal{F}_{t_1} \subseteq \mathcal{F}_{t_2}$. The tower property gives $c_{t_1} = \mathbb{E}[c_{t_2} \mid \mathcal{F}_{t_1}]$. Conditional variance decomposition yields

$$\mathrm{Var}(c_{t_2} \mid x) = \mathrm{Var}(c_{t_1} \mid x) + \mathbb{E}[\mathrm{Var}(c_{t_2} \mid \mathcal{F}_{t_1}) \mid x] \geq \mathrm{Var}(c_{t_1} \mid x).$$

Taking $\mathbb{E}_x$ proves monotonicity for a nested family. At the BA endpoint $c_0 = p(x)$, so the within-question variance is zero. At the complete-answer endpoint $c_T = Z$, so it equals $\mathbb{E}_x[p(x)(1-p(x))]$ and total variance is $\mathrm{Var}(Z) = \bar{p}(1-\bar{p})$, proving (iv). □

Finite-pool benchmarks and relative selection. With $M = 8$ conditionally independent evaluation solutions per question and $\hat{p} = M^{-1}\sum_i Z_i$, finite-pool noise gives

$$\mathrm{Var}(\hat{p}) = \mathrm{Var}_x(p(x)) + \frac{\mathbb{E}_x[p(x)(1-p(x))]}{M}, \qquad \mathbb{E}[\hat{p}(1-\hat{p})] = \frac{M-1}{M}\mathbb{E}_x[p(x)(1-p(x))].$$

The plug-in between-question benchmark is therefore inflated and the within-question endpoint benchmark reduced. For fixed relative u, the selected-state conditional expectation still satisfies the mean, between-question variance, and endpoint identities. However, the information states need not be nested across u, so variance monotonicity is not guaranteed. In the training mixture, nearest-integer rounding selects $t = 0$ with probability $1/(2T)$:

$$\Pr(Z = 1 \mid x, \tau = 0) = \frac{\mathbb{E}[Z/T \mid x]}{\mathbb{E}[1/T \mid x]}.$$

This length-weighted target can differ from forcing BA on every evaluation trace. Sampling BA independently of T would remove that weighting. Calibration under relative selection alone does not establish calibration at a fixed online token index.

A.7 The standardized update at general G

The exact threshold reduction is special to $G = 2$. For $G \geq 3$, standardization retains the direction of the centered reward vector, which contains ratios of differences as well as their signs. We therefore present two algebraic facts and a frozen-opponent characterization, making no universal claim about the finite-G fixed point. The implementation groups probes from the same question but generally uses different frozen solutions, prefixes, and relative positions, so their states can have different conditional values: decoupling means only that each outcome is fixed before its announcement is sampled, not that announcements and outcomes are marginally independent across heterogeneous states. The following lemmas require only a realized reward vector and set the numerical stabilizer to zero. Write

$$A_i = (r_i - \bar{r})/s_G, \qquad \bar{r} = \frac{1}{G}\sum_j r_j, \qquad s_G^2 = \frac{1}{G}\sum_j (r_j - \bar{r})^2.$$

Lemma 9 (The advantage vector lies on a fixed sphere)**.** *Whenever $s_G > 0$,*

$$\sum_{i=1}^{G} A_i = 0, \qquad \sum_{i=1}^{G} A_i^2 = G.$$

Proof. $\sum_i (r_i - \bar{r}) = 0$ gives the first identity. For the second, $\sum_i (r_i - \bar{r})^2 = G\, s_G^2$ by the definition of s_G, so dividing by $s_G^2 > 0$ gives exactly G. (With the unbiased convention $s_G^2 = \frac{1}{G-1}\sum_j (r_j - \bar{r})^2$ the constant is $G - 1$; nothing below depends on which is used.) □

This is the precise sense in which standardization discards the length of the centered reward vector: the advantage vector lives on the intersection of $\{\mathbf{1}^\top A = 0\}$ with the sphere of radius $\sqrt{G}$, a $(G-2)$-sphere, so the rewards enter only through that vector's *direction*. At $G = 2$ this set is S^0, a pair of antipodal points, which is why only $\operatorname{sign}(r_1 - r_2)$ survives in Theorem 4; for $G \geq 3$ the direction carries more than an ordering, so no analogous reduction follows. The weight vector also has fixed norm even when the reward spread is small, whereas group-mean subtraction gives norm $\sqrt{G}\, s_G$ — hence standardization need not attenuate sampled weights near equal rewards. Neither fact determines the norm or fixed point of the parameter update, which also depends on score gradients, clipping, and regularization.

Lemma 10 (Monotone in one's own reward)**.** *Fix $G \geq 2$ and $r_2, \ldots, r_G$, and write $a = \sum_{j\geq 2} r_j$, $b = \sum_{j\geq 2} r_j^2$ and*

$$Q(r_1) = (G-1)r_1^2 - 2ar_1 + Gb - a^2.$$

Then $G^2 s_G^2 = Q$, so $s_G > 0 \iff Q > 0$, and on $\{Q > 0\}$

$$A_1 = \frac{(G-1)r_1 - a}{\sqrt{Q}}, \qquad \frac{\partial A_1}{\partial r_1} = \frac{G\big[(G-1)b - a^2\big]}{Q^{3/2}} \;\geq\; 0,$$

with equality if and only if $r_2 = \cdots = r_G$. Hence A_1 is nondecreasing in its own reward, and strictly increasing unless every sibling earns the same reward.

Proof. $G^2 s_G^2 = G\sum_j r_j^2 - \big(\sum_j r_j\big)^2 = G(r_1^2+b) - (r_1+a)^2 = Q$, and $r_1 - \bar{r} = \frac{1}{G}\big[(G-1)r_1 - a\big]$, so the two factors of G cancel in $A_1 = (r_1 - \bar{r})/s_G = N/\sqrt{Q}$ with $N := (G-1)r_1 - a$. Differentiating,

$$\frac{\partial}{\partial r_1}\Big(\frac{N}{\sqrt{Q}}\Big) = Q^{-3/2}\Big(N'Q - \tfrac{1}{2}NQ'\Big), \qquad N' = G-1, \quad Q' = 2\big[(G-1)r_1 - a\big] = 2N,$$

so the bracket is $(G-1)Q - N^2$. Expanding,

$$(G-1)Q - N^2 = (G-1)\big[(G-1)r_1^2 - 2ar_1 + Gb - a^2\big] - \big[(G-1)r_1 - a\big]^2 = G\big[(G-1)b - a^2\big],$$

the r_1^2 and r_1 terms canceling identically. By Cauchy–Schwarz, $a^2 = \big(\sum_{j\geq 2} r_j\big)^2 \leq (G-1)\sum_{j\geq 2} r_j^2 = (G-1)b$, with equality iff $r_2 = \cdots = r_G$. □

Theorem 11 (Frozen-opponent best response at any group size)**.** *Fix a homogeneous probe state with value $V \in (0,1)$, the Brier reward $r(c,z) = 1 - (c-z)^2$, and a frozen opponent $\bar{\pi}$ on $[0,1]$. Let $G \geq 2$ and let $Z_1, \ldots, Z_G$ have a θ-independent exchangeable law with marginal success probability V. Conditional on the state, announcements are mutually independent and independent of the outcome vector. Use population standard deviation, set $\epsilon = 0$, and assign $A_i = 0$ when $s_G = 0$. For a fixed focal announcement $c_1 = c$ and siblings $c_2, \ldots, c_G \overset{\text{iid}}{\sim} \bar{\pi}$, write*

$$\psi_z(c) = \mathbb{E}\big[A_1 \,\big|\, Z_1 = z;\; c_1 = c\big], \qquad \Psi_{\bar{\pi},G}(c) = V\psi_1(c) + (1-V)\psi_0(c).$$

The expectations include zero-variance groups, and gradient statements additionally assume that score-function differentiation is valid with $\mathbb{E}_{c\sim\pi_\theta}\|\nabla_\theta \log \pi_\theta(c)\| < \infty$. Then:

(i) the expected on-policy score-function update is the frozen-opponent semi-gradient

$$g_{\mathrm{std},G}(\theta) := \mathbb{E}_{c_1,\ldots,c_G \overset{\text{iid}}{\sim}\pi_\theta}\left[\sum_{i=1}^{G} A_i \nabla_\theta \log \pi_\theta(c_i)\right] = G\,\nabla_\theta \mathbb{E}_{c\sim\pi_\theta}[\Psi_{\bar{\pi},G}(c)]\big|_{\bar{\pi}=\mathrm{sg}(\pi_\theta)},$$

with the expectation also over outcomes and the opponent held fixed during differentiation;

(ii) ψ_1 is nondecreasing and ψ_0 nonincreasing. For each z, the corresponding monotonicity is strict if the siblings' rewards are not all equal with positive probability conditional on $Z_1 = z$;

(iii) $\Psi_{\bar{\pi},G}$ and its best responses are unchanged by a common transformation $r \mapsto \alpha r + \beta$ for every $\alpha > 0$;

(iv) at $G = 2$, $\Psi_{\bar{\pi},2}(c) = (2V - 1)D_{\bar{\pi}}(c)$, recovering Corollary 5;

(v) for $G \geq 3$, any interior maximizer $c^\star$ at which both ψ_z are differentiable satisfies $V\psi_1'(c^\star) + (1-V)\psi_0'(c^\star) = 0$. The assumptions do not rule out boundary best responses;

(vi) additionally suppose $Z_i \overset{\text{iid}}{\sim} \text{Bernoulli}(V)$ as G grows. Let μ and $\sigma^2 > 0$ be the mean and variance of $r(c', Z)$ for independent $c' \sim \bar{\pi}$ and $Z \sim \text{Bernoulli}(V)$. Then

$$\Psi_{\bar{\pi},G}(c) \longrightarrow \frac{\mathbb{E}_Z[r(c,Z)] - \mu}{\sigma} \quad \textit{uniformly in } c \in [0,1].$$

For a fixed opponent, the focal objective's gradient converges to $\sigma^{-1}\nabla_\theta \mathbb{E}_{c\sim\pi_\theta, Z}[r(c,Z)]$; equivalently, the group-average update $g_{\text{std},G}/G$ has this limit when $\bar{\pi} = \text{sg}(\pi_\theta)$ and $\sigma > 0$. The limiting frozen-opponent objective is uniquely maximized at $c = V$.

Proof. (i) When all announcements follow π_θ, the estimator's G terms are exchangeable. Conditioning the first term on $c_1 = c$ replaces A_1 by $\Psi_{\bar{\pi},G}(c)$ with $\bar{\pi} = \text{sg}(\pi_\theta)$. For fixed $\bar{\pi}$, the score identity gives $\nabla_\theta \mathbb{E}_{c\sim\pi_\theta}[\Psi_{\bar{\pi},G}(c)] = \mathbb{E}_{c\sim\pi_\theta}[\Psi_{\bar{\pi},G}(c)\nabla_\theta \log \pi_\theta(c)]$. This differentiates only the focal policy, not the opponent's policy occurrences.

(ii) Condition on $Z_1 = 1$. Then $r_1 = 1 - (1-c)^2$ is increasing in c while the siblings' rewards do not depend on c, so Lemma 10 makes A_1 nondecreasing in c, strictly so on $\{(G-1)b > a^2\}$. If all siblings earn the same reward r', then $A_1 = \sqrt{G-1}\,\text{sign}(r_1 - r')$, including zero at a tie, which is also nondecreasing in r_1. Taking conditional expectations proves monotonicity and the stated sufficient condition for strictness. For $Z_1 = 0$, $r_1 = 1 - c^2$ is decreasing and the same argument applies.

(iii) Under $r \mapsto \alpha r + \beta$ we have $\bar{r} \mapsto \alpha\bar{r} + \beta$ and $s_G \mapsto \alpha s_G$. Both the standardized advantages and the zero-variance convention are unchanged, so every conditional mean and best response is unchanged.

(iv) With population standard deviation and the tie convention, $A_1 = \text{sign}(r_1 - r_2)$ at $G = 2$. The exchangeability argument in Corollary 5 evaluates its unconditional mean as $(2V-1)D_{\bar{\pi}}(c)$: the two mixed-outcome cases cancel because their joint probabilities are equal.

(v) At a differentiable interior maximizer, $\Psi'_{\bar{\pi},G} = 0$ gives the stated condition. Boundary best responses can persist: if all outcomes share one $Z \sim \text{Bernoulli}(V)$ and $\bar{\pi} = \delta_a$ for $a \in (0,1)$, then the equal-sibling formula in (ii) gives $\Psi_{\bar{\pi},G}(c) = \sqrt{G-1}(2V-1)\,\text{sign}(c-a)$ for every G. Thus $c = 1$ remains a best response when $V > 1/2$, and $c = 0$ when $V < 1/2$.

(vi) Under the additional independence assumption, the $m = G-1$ sibling rewards are iid, bounded and independent of the focal outcome; let $\hat{\mu}_m$ and $\hat{v}_m$ be their empirical mean and population variance, so that $\hat{\mu}_m \to \mu$ and $\hat{v}_m \to \sigma^2$ almost surely by the strong law. For any focal reward $r_1 \in [0,1]$,

$$s_G^2 = \frac{m}{G}\hat{v}_m + \frac{m}{G^2}(r_1 - \hat{\mu}_m)^2, \qquad A_1 = \frac{(m/G)(r_1 - \hat{\mu}_m)}{s_G} \quad (s_G > 0).$$

Hence $A_1 \to (r_1 - \mu)/\sigma$ almost surely, uniformly in $r_1 \in [0,1]$. To pass to expectations, note that $|A_1| \leq 2/\sigma$ on $\{\hat{v}_m \geq \sigma^2/2\}$, while Chebyshev's inequality applied to the bounded empirical first two moments gives $\Pr(\hat{v}_m < \sigma^2/2) = O(m^{-1})$ and, on that exceptional event, Lemma 9 with the tie convention gives $|A_1| \leq \sqrt{G}$, so its contribution vanishes. Bounded convergence on the complementary event proves the uniform limit for $\Psi_{\bar{\pi},G}$; the integrable score assumption permits the same limit inside the focal gradient, and the constant $-\mu/\sigma$ contributes zero because $\mathbb{E}_{c\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(c)] = 0$. Finally, $\mathbb{E}_Z[r(c,Z)] = 1 - V(1-V) - (c-V)^2$, so the limiting objective uniquely elicits V. Exchangeability alone does not justify this limit; the shared-outcome example in (v), with $a \neq 1/2$ so that $\sigma > 0$, illustrates the failure. □

What the lemmas do and do not establish. Lemmas 9–10 together identify the scale-loss mechanism, and Theorem 11 assembles them into a best-response statement at any group size for a *homogeneous* state against a *frozen* opponent. What remains unestablished is the behavior when states, outcomes, and the opponent distribution all vary together, as they do in a training run, and — beyond the first-order condition of part (v) — where the finite-G maximizer actually sits.

A.8 WHICH UPDATES PRESERVE PROPERNESS

Action-independent baselines. Subtracting a baseline that does not depend on the sampled announcement leaves the expected on-policy score-function gradient unchanged, and multiplying by a deterministic positive constant changes only the step size. GRPO with group-mean subtraction and RLOO (Ahmadian et al., 2024) have this property under the assumptions of Theorem 3. A scale estimated from the same random batch is different: it can correlate with the score estimator and reweight batches, even without creating a separate comparison game inside each prompt group. Batch-standardized REINFORCE++ is therefore outside the exact identity, and its similarity to the group-mean subtraction and RLOO arms in Table 2 is an empirical observation.

Per-group scale normalization. Dividing by a statistic that depends on sibling rewards is not an action-independent baseline; it makes the weights invariant to a common positive affine transformation of each group's rewards. At $G = 2$ only the comparison sign remains, and Corollary 5 is an exact counterexample to probability elicitation. At $G \geq 3$, affine invariance alone does *not* prove a particular biased optimum: standardized weights retain ratios of differences, and the fixed point can depend on the policy, state mixture, clipping, and regularization. Our general claim is therefore the scale-loss mechanism, supported at the practical $G = 8$ by the ablation rather than by a universal optimum theorem.

B IMPLEMENTATION DETAILS

Shared setup. Unless an alternative optimizer is specified, the RL arms use GRPO (Shao et al., 2024) with std off on 8×H100-80GB under one codebase, with learning rate $1{\times}10^{-6}$, $G = 8$ samples per prompt, PPO mini-batch 128, a KL-to-reference penalty of 1×10^{-3} (`low_var_kl`), sampling temperature 1.0 during training and 0.6 at evaluation, and `enable_thinking=False` for hybrid-reasoning models so that a closed reasoning block is never stripped from a replayed prefix. Rollout generation uses vLLM with tensor-parallel 2. Arm-specific deviations are listed below.

B.1 OUR ALGORITHM: PACE-u

Two stages. *(1) Frozen pool.* For each of the 16K training questions we sample $N{=}32$ solutions once from the initialization, at temperature 1.0, grade them, and store the token IDs together with the per-solution outcomes. Nothing in stage 2 regenerates a solution or assigns reward to reasoning tokens. Group sampling is without replacement, and pool depth is checked to be at least G.

(2) Position-agnostic confidence training. Each step draws a batch of 512 questions, $G{=}8$ pool solutions per question, and one position per solution: $u \sim U[0, 1]$ with the probe placed at $t = \lfloor uT \rceil$. The prompt is the probe state $s_t = (x, \mathtt{R}, y_{\leq t}, \mathtt{P})$ and the response is the confidence alone, capped at 64 tokens; the reward is $1 - (c - Z)^2$ against *that solution's* own outcome, weight 1.0, plus a format term of weight 0.1.

Chat-template serialization. At $t = 0$, one user turn contains x, `R` and `P`. At $0 < t \leq T$, the turns are user $(x, \mathtt{R})$, assistant $y_{\leq t}$, then user `P`; the assistant turn is closed even at an interior position. We retain `R` to match pool generation. On 500 *unforced greedy* probes of the $t = 0$-trained ablation checkpoint, including it changes parse rate $99.2\% \rightarrow 89.0\%$, ECE $0.150 \rightarrow 0.212$, and AUROC $0.679 \rightarrow 0.700$. That parse cost is specific to unforced decoding: the endpoint probes reported in Table 1 are forced by pre-filling `\confidence{`, so both PACE arms parse at 100% there and no reported metric is computed on a subset. The nested-prefix argument in Section 3 concerns the available information, independently of this template choice. This evaluates a separate probe call, not uninterrupted confidence emission within a JSON response.

Format reward and training budget. The format term is a sum of discrete penalties on a confidence-only response, shifted into $[0, 1]$: starting from 1, subtract 0.35 if the reply does not contain exactly one `\confidence{}`; 0.35 for a `\boxed{}` holding anything other than the confidence itself; 0.35 if more than 100 characters of other text surround the confidence; and 0.30 if the reply is mostly non-Latin script. The result is clipped to $[0, 1]$.

An unparseable confidence receives Brier reward $r = 0$, the minimum in Equation 2, and remains in the group. Its reward also enters the group-mean baseline. The main Base-PACE run uses 100 steps (51,200 question presentations); RLVR-PACE uses 230 steps (117,760 presentations). Standard-deviation division is disabled (`norm_adv_by_std_in_grpo=False`); Theorem 3 motivates this choice and §6.3.2 tests it empirically. These budgets are not compute-matched to the baselines; PACE also pays for the offline pool and prefix processing, which the short probe-completion cap does not include.

GLM prompt compatibility. Under the default `P`, the untrained GLM-4-9B verbalizer often copies the placeholder `\confidence{p}` literally, leaving a parse rate of 80.1% on DeepMath-500. For a fair comparison we therefore report the untrained baseline under a reworded instruction, `Do not reason. Output only the probability that your final answer will be correct, as a number from 0.00 to 1.00.`, which parses at 100%. GLM PACE-u uses the default `P`, so a prompt difference remains between the two rows.

B.2 Baselines

Untrained and lightly trained readouts. *AvgProb* needs no training: it is the mean teacher-forced token probability over the question at $t = 0$ and the stored answer at $t = T$. *ProbeMLP* is a two-layer MLP, hidden width 256 with ReLU and dropout 0.1, on the hidden state at the readout position, trained with logistic loss on held-out correctness labels. At $t = 0$ the feature is the final-layer state of the *last prompt token*, and at $t = T$ it is the *middle-layer* state (index $\lfloor L/2 \rfloor$ of the $L+1$ returned hidden states, counting the embedding layer) at the last solution token. A separate probe is fitted per position. Optimization is AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 256, at most 40 epochs, early-stopped on the AUROC of a *question-disjoint* 10% validation split. The probe is re-fitted for each initialization on its own states and rollouts. *Verb* is the same prompt as our probe applied to an untrained checkpoint. *ConfTuner* (Li et al., 2025) is supervised, not on-policy: it minimizes a tokenized Brier score over the ten digit tokens at the single confidence position, learning rate 1×10^{-5}, 10 epochs. We evaluate **epoch 9**, selected on held-out AUROC among the saved epochs. The Base prefix denotes initialization: ConfTuner's BA and AA variants are separate fine-tuned solvers.

Coupled RL baselines. Each baseline is trained with its own prompt and read at the position defined by that format.

- *RLCR* (Damani et al., 2026): after-answer confidence; reward is correctness plus format plus Brier against the rollout's own Z; batch 256, 100 steps, std division **off**. Completions average about 1450 tokens, versus PACE's 64-token cap.
- *RLCC-L1* (Zhang et al., 2026): after-answer confidence; reward is correctness plus format plus $1 - |c - \hat{p}|$, with group pass rate $\hat{p}$; batch 256, 100 steps, std division **on**. Format weight is 1.0, matching the unit coefficients in the original paper's Equation (1).
- *CoCA* (Li et al., 2026): confidence first, with per-segment standardized advantages applied to their respective token spans; batch 256, 100 steps, and std division **on** as published. We also include a variant with std division **off** for the confidence reward to isolate the effect of the coupled prompt on confidence calibration. CoCA supplies only $t = 0$ confidence.
- *CALIBER* (Finlay et al., 2026): one rollout emits `\preconfidence{}` before reasoning and `\postconfidence{}` after the answer, targeting group success rate and own correctness, respectively. All terms are gated on format reward; 100 steps, std division **off**, with the paper's two-stage warmup reproduced.

B.3 Evaluation, grading, and prompts

Metric definitions. Write $c_i \in [0, 1]$ for the i-th reported confidence and $z_i \in \{0, 1\}$ for the graded correctness of the solution it describes, over n scored reports.

Expected calibration error uses $B = 10$ equal-width bins on $[0, 1]$, with $\mathcal{B}_b$ the reports whose confidence falls in bin b:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{n} \left| \, \bar{c}_{\mathcal{B}_b} - \bar{z}_{\mathcal{B}_b} \right|, \qquad \bar{c}_{\mathcal{B}_b} = \frac{1}{|\mathcal{B}_b|} \sum_{i \in \mathcal{B}_b} c_i.$$

Empty bins contribute nothing. Equal-width binning is reported because it is the common convention, but it is fragile for a verbalizer that concentrates its mass: if every report falls in one bin, ECE reduces to $|\bar{c} - \bar{z}|$ and measures only the aggregate gap. Reported ECE values for such arms should therefore be read alongside their confidence distributions.

AUROC is the probability that a correct solution receives a higher confidence than an incorrect one, computed as the Mann–Whitney statistic with mid-ranks for ties. With n_1 correct and n_0 incorrect reports and R_i the rank of c_i in the pooled sample,

$$\mathrm{AUROC} = \frac{\sum_{i:z_i=1} R_i - \frac{1}{2} n_1 (n_1 + 1)}{n_1 n_0}.$$

Tie handling is load-bearing rather than cosmetic: verbalized confidence is heavily tied, and scoring ties as wins would report discrimination the readout does not have. A cell with no correct or no incorrect report is undefined and is omitted.

Brier score is the mean squared error of the report against the outcome,

$$\mathrm{Brier} = \frac{1}{n} \sum_{i=1}^{n} (c_i - z_i)^2,$$

which admits Murphy's decomposition into reliability minus resolution plus the base-rate term $\bar{p}(1 - \bar{p})$. Conditional on $\mathcal{F}_t$ the irreducible squared-error risk is instead $\mathbb{E}[V_t(1 - V_t)]$, the complement of the ceiling in Theorem 2, and it vanishes at $t = T$ under a deterministic verifier. Because the base-rate term depends on accuracy, Brier is comparable across arms only when they share a pool: an arm that answers more accurately has a smaller base-rate term and so a lower attainable Brier. ECE and AUROC carry no such dependence.

The position sweep and the reported cells. Figure 3 and Table 1 aggregate the same kind of draws differently. The sweep scores all $4{,}000$ forced probes at a position together (500 questions $\times$ 8 stored answers, one draw each), whereas a reported cell averages eight permutations that each score one answer per question. ECE and AUROC are nonlinear in the sample, so the two aggregations differ slightly where the sweep's endpoints meet the table's $t{=}0$ and $t{=}T$ columns; Brier, a plain mean, agrees whenever every unit parses.

Grading. Each dataset is graded by the rule its answer format requires, and the grader is identical across arms. Mathematics (DeepMath-500, MATH-500, OlympiadBench) uses `math_verify` (Hugging Face, 2024) on the content of the last `\boxed{}`; GSM8K uses the final-number rule; GPQA-Diamond and CommonsenseQA use multiple-choice extraction; LiveCodeBench executes the emitted program against stored test cases, with a per-case timeout inside a container. A response with no extractable answer scores 0.

LiveCodeBench results cover 400 questions with at most eight stored tests each. The custom grader has not been validated against the official harness, so the reported scores apply to this subset and test protocol. Two of 3,200 rollout labels (0.0625%) were load-dependent under the timeout; this is a repeatability check, not a measure of agreement with the official harness.

Checkpoint selection. RL arms use their own validation-selected checkpoint among those saved, rather than a common step; the supervised readouts use the selection rules stated above. The selection set is the held-out 500-question split of the frozen DeepMath pool — disjoint from the 16K training questions and from every evaluation benchmark — scored at the position the arm was trained for, and the criterion is total reward convergence. The main PACE-u rows use step 100 and step 230 for the base and RLVR initializations, respectively.

Prompts and parsing. The reasoning instruction `R` is "Please reason step by step, and output the final answer within `\boxed{}`". The probe instruction `P` is "Do not reason. Output only `\confidence{p}`, p (0.00 $\sim$ 1.00) = probability your final answer will be correct." A confidence is parsed as the content of the first `\confidence{}` (or, for CALIBER, of the matching pre/post slot), accepted only if it reads as a number in $[0, 1]$; there is no clipping and no rescaling. Parse rates are reported per cell in §C, where teacher-forced readouts show `n/a`. For decoupled arms at $t = 0$ and $t = T$ the probe position is forced by pre-filling `\confidence{`. Parse rates must still be checked per cell: prefilling does not guarantee a valid numeric completion.

Coupled-baseline instructions. Each coupled baseline appends its own instruction to the question:

- *RLCR* and *RLCC-L1* (after-answer): `Please reason step by step, and put your final answer within \boxed{}, then output the confidence (0.0~1.0) that your answer is correct within \confidence{}.`
- *CoCA* (confidence-first): `Please first output the confidence (0.0~1.0) that you can solve the problem correctly within \confidence{}, then reason step by step, and put your final answer within \boxed{}.`
- *CALIBER* (pre- and post-confidence): `Please first output the confidence (0.0~1.0) that you can solve the problem correctly within \preconfidence{}, then reason step by step, and put your final answer within \boxed{}, then output the confidence (0.0~1.0) that your answer is correct within \postconfidence{}.`

Table 4: **Confidence calibration of different methods on DeepMath-500** (500 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.634	n/a	0.114 ±0.016	0.472 ±0.023	0.242 ±0.004	0.634	n/a	0.252 ±0.016	0.657 ±0.018	0.287 ±0.013
Base-ProbeMLP	0.634	n/a	0.076 ±0.011	0.761 ±0.017	0.194 ±0.008	0.634	n/a	0.213 ±0.011	0.785 ±0.013	0.232 ±0.011
Base-Verb	0.634	97.2%	0.347 ±0.017	0.558 ±0.017	0.352 ±0.017	0.634	97.2%	0.367 ±0.017	0.554 ±0.008	0.365 ±0.017
Base-ConfTuner	0.645	n/a	0.049 ±0.009	0.755 ±0.017	0.188 ±0.006	0.699	n/a	0.035 ±0.005	0.839 ±0.013	0.144 ±0.007
Base-PACE-$u^\star$	0.640	100.0%	0.076 ±0.012	0.746 ±0.017	0.195 ±0.007	100.0%	0.097 ±0.010	0.805 ±0.013	0.178 ±0.006	
RLVR-AvgProb	0.801	n/a	0.218 ±0.015	0.468 ±0.028	0.213 ±0.004	0.801	n/a	0.165 ±0.015	0.683 ±0.021	0.184 ±0.014
RLVR-ProbeMLP	0.801	n/a	0.079 ±0.012	0.750 ±0.021	0.145 ±0.009	0.801	n/a	0.100 ±0.009	0.794 ±0.015	0.153 ±0.008
RLVR-ConfTuner	0.797	n/a	0.035 ±0.009	0.751 ±0.021	0.138 ±0.008	0.802	n/a	0.032 ±0.005	0.845 ±0.016	0.113 ±0.007
RLVR-Verb	0.801	99.0%	0.198 ±0.015	0.604 ±0.021	0.200 ±0.015	0.801	98.6%	0.197 ±0.015	0.541 ±0.011	0.197 ±0.015
Base-RLCR	–	–	–	–	–	0.797	97.8%	0.179 ±0.015	0.693 ±0.020	0.182 ±0.014
Base-RLCC-L1	–	–	–	–	–	0.775	98.7%	0.217 ±0.015	0.554 ±0.011	0.215 ±0.015
Base-CoCA	0.784	100.0%	0.211 ±0.015	0.550 ±0.014	0.209 ±0.015	–	–	–	–	–
Base-CoCA (std off)	0.795	100.0%	0.041 ±0.011	0.684 ±0.022	0.153 ±0.009	–	–	–	–	–
Base-CALIBER	0.760	99.6%	0.105 ±0.014	0.711 ±0.021	0.174 ±0.006	0.760	99.6%	0.076 ±0.011	0.783 ±0.017	0.153 ±0.009
RLVR-PACE-$u^\star$	0.811	100.0%	0.038 ±0.008	0.751 ±0.023	0.132 ±0.008	100.0%	0.055 ±0.008	0.869 ±0.014	0.107 ±0.006	

Table 5: **Confidence calibration of different methods on MATH-500** (OOD math, 500 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.851	n/a	0.279 ±0.013	0.307 ±0.025	0.214 ±0.004	0.851	n/a	0.068 ±0.011	0.775 ±0.021	0.123 ±0.010
Base-ProbeMLP	0.851	n/a	0.056 ±0.008	0.844 ±0.016	0.099 ±0.007	0.851	n/a	0.104 ±0.009	0.853 ±0.014	0.109 ±0.009
Base-Verb	0.851	98.8%	0.126 ±0.012	0.693 ±0.015	0.139 ±0.012	0.851	98.3%	0.149 ±0.013	0.549 ±0.011	0.148 ±0.013
Base-ConfTuner	0.844	n/a	0.067 ±0.009	0.868 ±0.014	0.101 ±0.006	0.872	n/a	0.041 ±0.005	0.887 ±0.015	0.070 ±0.005
Base-PACE-$u^\star$	0.854	100.0%	0.061 ±0.008	0.845 ±0.017	0.099 ±0.006	100.0%	0.050 ±0.006	0.879 ±0.016	0.078 ±0.006	
RLVR-AvgProb	0.899	n/a	0.311 ±0.012	0.277 ±0.030	0.201 ±0.004	0.899	n/a	0.079 ±0.011	0.801 ±0.024	0.095 ±0.011
RLVR-ProbeMLP	0.899	n/a	0.066 ±0.009	0.844 ±0.020	0.083 ±0.009	0.899	n/a	0.051 ±0.007	0.888 ±0.018	0.073 ±0.007
RLVR-ConfTuner	0.901	n/a	0.023 ±0.005	0.863 ±0.017	0.070 ±0.006	0.913	n/a	0.025 ±0.004	0.912 ±0.015	0.050 ±0.005
RLVR-Verb	0.899	99.4%	0.090 ±0.011	0.658 ±0.028	0.097 ±0.011	0.899	98.5%	0.103 ±0.011	0.536 ±0.013	0.103 ±0.011
Base-RLCR	–	–	–	–	–	0.897	98.2%	0.072 ±0.010	0.737 ±0.022	0.085 ±0.010
Base-RLCC-L1	–	–	–	–	–	0.891	98.6%	0.099 ±0.011	0.605 ±0.021	0.097 ±0.011
Base-CoCA	0.894	100.0%	0.106 ±0.012	0.510 ±0.009	0.106 ±0.012	–	–	–	–	–
Base-CoCA (std off)	0.903	100.0%	0.066 ±0.007	0.794 ±0.025	0.078 ±0.007	–	–	–	–	–
Base-CALIBER	0.881	99.6%	0.113 ±0.008	0.839 ±0.018	0.097 ±0.005	0.881	99.6%	0.085 ±0.008	0.831 ±0.023	0.078 ±0.007
RLVR-PACE-$u^\star$	0.908	100.0%	0.039 ±0.006	0.849 ±0.019	0.070 ±0.006	100.0%	0.030 ±0.005	0.930 ±0.011	0.051 ±0.005	

C MAIN RESULTS, PER DATASET

Table 1 macro-averages each group; Tables 4–10 give the same columns per dataset. Cells are **mean ± standard error**. Each of eight evaluation permutations scores one stored rollout per question, and the eight together use every rollout exactly once; the mean is their average on the data as collected. The standard error is the spread of that same eight-permutation average across 1000 bootstrap replicates, each resampling questions with replacement to the dataset's size. The bars therefore measure which questions were evaluated; rollout selection is averaged over rather than resampled, so it is quantified separately below, and neither quantity reflects independent training runs.

Evaluation-seed spread is narrower than the reported bars, capturing only probe and rollout-selection noise on a fixed question set; question-bootstrap errors are $1.1\times$–$2.0\times$ larger. AvgProb, ProbeMLP, and ConfTuner have no probe noise: each is a deterministic readout from one teacher-forced forward pass over a stored solution, and at $t = 0$ the first two condition on the prompt alone, returning one value per question. Their uncertainty is thus entirely across questions, which the bootstrap captures, and they cannot fail to parse. Per-cell errors do not give paired intervals for between-method differences.

In the parse column, `n/a` marks a teacher-forced readout: it reads a number out of logits or hidden features and never parses text, so the share of units emitting a parseable confidence is undefined for it rather than 100%. "–" marks a position the method does not have at all.

Table 6: **Confidence calibration of different methods on GSM8K** (OOD math, 1319 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.916	n/a	0.404 ±0.007	0.611 ±0.024	0.240 ±0.001	0.916	n/a	0.020 ±0.004	0.758 ±0.019	0.073 ±0.006
Base-ProbeMLP	0.916	n/a	0.029 ±0.006	0.674 ±0.024	0.075 ±0.006	0.916	n/a	0.081 ±0.007	0.656 ±0.023	0.082 ±0.006
Base-Verb	0.916	100.0%	0.066 ±0.007	0.623 ±0.019	0.080 ±0.006	0.916	99.9%	0.084 ±0.007	0.509 ±0.003	0.084 ±0.007
Base-ConfTuner	0.918	n/a	0.025 ±0.004	0.723 ±0.019	0.071 ±0.005	0.921	n/a	0.044 ±0.006	0.760 ±0.019	0.065 ±0.005
Base-PACE-$u^\star$	0.917	100.0%	0.031 ±0.006	0.671 ±0.019	0.074 ±0.005	100.0%	0.049 ±0.006	0.740 ±0.021	0.067 ±0.005	
RLVR-AvgProb	0.944	n/a	0.430 ±0.006	0.632 ±0.032	0.238 ±0.001	0.944	n/a	0.039 ±0.006	0.759 ±0.022	0.054 ±0.005
RLVR-ProbeMLP	0.944	n/a	0.055 ±0.006	0.734 ±0.025	0.056 ±0.006	0.944	n/a	0.040 ±0.006	0.742 ±0.025	0.052 ±0.005
RLVR-ConfTuner	0.943	n/a	0.028 ±0.005	0.690 ±0.026	0.051 ±0.005	0.947	n/a	0.026 ±0.003	0.779 ±0.024	0.046 ±0.004
RLVR-Verb	0.944	100.0%	0.049 ±0.006	0.588 ±0.022	0.055 ±0.006	0.944	100.0%	0.056 ±0.006	0.502 ±0.002	0.056 ±0.006
Base-RLCR	–	–	–	–	–	0.940	100.0%	0.029 ±0.006	0.638 ±0.018	0.056 ±0.005
Base-RLCC-L1	–	–	–	–	–	0.941	100.0%	0.058 ±0.006	0.556 ±0.014	0.058 ±0.006
Base-CoCA	0.941	100.0%	0.059 ±0.006	0.501 ±0.002	0.059 ±0.006	–	–	–	–	–
Base-CoCA (std off)	0.945	100.0%	0.048 ±0.006	0.525 ±0.011	0.054 ±0.005	–	–	–	–	–
Base-CALIBER	0.934	100.0%	0.067 ±0.006	0.653 ±0.026	0.064 ±0.005	0.934	100.0%	0.055 ±0.005	0.636 ±0.019	0.058 ±0.005
RLVR-PACE-$u^\star$	0.946	100.0%	0.023 ±0.005	0.718 ±0.026	0.049 ±0.004	100.0%	0.018 ±0.005	0.825 ±0.020	0.045 ±0.005	

Table 7: **Confidence calibration of different methods on OlympiadBench** (OOD math, 674 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.548	n/a	0.114 ±0.014	0.405 ±0.018	0.267 ±0.004	0.548	n/a	0.343 ±0.015	0.777 ±0.014	0.342 ±0.012
Base-ProbeMLP	0.548	n/a	0.108 ±0.011	0.833 ±0.013	0.176 ±0.007	0.548	n/a	0.230 ±0.010	0.817 ±0.011	0.235 ±0.009
Base-Verb	0.548	98.8%	0.411 ±0.016	0.570 ±0.011	0.414 ±0.015	0.548	98.6%	0.451 ±0.016	0.555 ±0.006	0.449 ±0.016
Base-ConfTuner	0.544	n/a	0.062 ±0.010	0.797 ±0.014	0.180 ±0.006	0.587	n/a	0.102 ±0.010	0.869 ±0.010	0.155 ±0.006
Base-PACE-$u^\star$	0.554	100.0%	0.139 ±0.012	0.817 ±0.012	0.189 ±0.005	100.0%	0.169 ±0.011	0.876 ±0.010	0.176 ±0.006	
RLVR-AvgProb	0.637	n/a	0.129 ±0.016	0.394 ±0.020	0.253 ±0.004	0.637	n/a	0.330 ±0.016	0.803 ±0.015	0.330 ±0.015
RLVR-ProbeMLP	0.637	n/a	0.183 ±0.014	0.748 ±0.017	0.230 ±0.010	0.637	n/a	0.137 ±0.009	0.820 ±0.012	0.188 ±0.007
RLVR-ConfTuner	0.631	n/a	0.104 ±0.012	0.801 ±0.015	0.185 ±0.007	0.646	n/a	0.086 ±0.009	0.876 ±0.011	0.140 ±0.007
RLVR-Verb	0.637	99.2%	0.340 ±0.016	0.555 ±0.016	0.345 ±0.015	0.637	99.0%	0.364 ±0.017	0.546 ±0.006	0.362 ±0.017
Base-RLCR	–	–	–	–	–	0.634	94.8%	0.320 ±0.015	0.685 ±0.014	0.319 ±0.014
Base-RLCC-L1	–	–	–	–	–	0.615	96.7%	0.367 ±0.016	0.613 ±0.010	0.360 ±0.016
Base-CoCA	0.635	100.0%	0.340 ±0.016	0.590 ±0.012	0.332 ±0.015	–	–	–	–	–
Base-CoCA (std off)	0.640	100.0%	0.178 ±0.014	0.794 ±0.014	0.221 ±0.009	–	–	–	–	–
Base-CALIBER	0.607	98.8%	0.119 ±0.012	0.816 ±0.013	0.178 ±0.004	0.607	98.8%	0.182 ±0.012	0.853 ±0.011	0.204 ±0.008
RLVR-PACE-$u^\star$	0.645	100.0%	0.100 ±0.011	0.817 ±0.014	0.179 ±0.006	100.0%	0.123 ±0.009	0.901 ±0.011	0.142 ±0.006	

Table 8: **Confidence calibration of different methods on CommonsenseQA** (OOD non-math, 1221 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.803	n/a	0.362 ±0.010	0.502 ±0.018	0.290 ±0.001	0.803	n/a	0.045 ±0.006	0.678 ±0.013	0.153 ±0.006
Base-ProbeMLP	0.803	n/a	0.152 ±0.010	0.497 ±0.017	0.196 ±0.008	0.803	n/a	0.170 ±0.009	0.571 ±0.012	0.186 ±0.009
Base-Verb	0.803	99.7%	0.162 ±0.009	0.530 ±0.012	0.184 ±0.009	0.803	99.9%	0.184 ±0.009	0.598 ±0.012	0.189 ±0.009
Base-ConfTuner	0.799	n/a	0.113 ±0.009	0.607 ±0.016	0.182 ±0.005	0.814	n/a	0.247 ±0.009	0.724 ±0.014	0.207 ±0.004
Base-PACE-$u^\star$	0.806	100.0%	0.164 ±0.009	0.565 ±0.014	0.184 ±0.003	100.0%	0.185 ±0.009	0.666 ±0.011	0.182 ±0.002	
RLVR-AvgProb	0.816	n/a	0.380 ±0.010	0.495 ±0.019	0.295 ±0.001	0.816	n/a	0.113 ±0.009	0.694 ±0.013	0.160 ±0.008
RLVR-ProbeMLP	0.816	n/a	0.149 ±0.009	0.531 ±0.018	0.172 ±0.009	0.816	n/a	0.063 ±0.009	0.630 ±0.013	0.146 ±0.007
RLVR-ConfTuner	0.817	n/a	0.048 ±0.009	0.557 ±0.015	0.151 ±0.007	0.817	n/a	0.080 ±0.009	0.758 ±0.013	0.137 ±0.004
RLVR-Verb	0.816	99.7%	0.156 ±0.010	0.529 ±0.015	0.174 ±0.009	0.816	100.0%	0.173 ±0.010	0.602 ±0.012	0.177 ±0.009
Base-RLCR	–	–	–	–	–	0.814	100.0%	0.128 ±0.010	0.598 ±0.010	0.164 ±0.008
Base-RLCC-L1	–	–	–	–	–	0.814	100.0%	0.149 ±0.010	0.636 ±0.012	0.169 ±0.009
Base-CoCA	0.811	100.0%	0.151 ±0.010	0.600 ±0.016	0.173 ±0.009	–	–	–	–	–
Base-CoCA (std off)	0.819	100.0%	0.071 ±0.010	0.532 ±0.010	0.153 ±0.008	–	–	–	–	–
Base-CALIBER	0.808	100.0%	0.045 ±0.009	0.585 ±0.016	0.154 ±0.005	0.808	100.0%	0.028 ±0.006	0.690 ±0.013	0.141 ±0.006
RLVR-PACE-$u^\star$	0.817	100.0%	0.030 ±0.008	0.598 ±0.015	0.147 ±0.006	100.0%	0.051 ±0.006	0.754 ±0.012	0.132 ±0.005	

Table 9: **Confidence calibration of different methods on GPQA-Diamond** (OOD non-math, 198 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.472	n/a	0.117 ±0.021	0.477 ±0.032	0.271 ±0.006	0.472	n/a	0.359 ±0.026	0.609 ±0.029	0.370 ±0.018
Base-ProbeMLP	0.472	n/a	0.209 ±0.025	0.520 ±0.031	0.294 ±0.014	0.472	n/a	0.357 ±0.021	0.593 ±0.024	0.371 ±0.018
Base-Verb	0.472	99.5%	0.498 ±0.027	0.582 ±0.021	0.494 ±0.025	0.472	99.5%	0.516 ±0.027	0.542 ±0.019	0.514 ±0.027
Base-ConfTuner	0.497	n/a	0.186 ±0.025	0.511 ±0.025	0.287 ±0.010	0.510	n/a	0.208 ±0.023	0.622 ±0.022	0.276 ±0.011
Base-PACE-$u^\star$	0.497	100.0%	0.039 ±0.014	0.543 ±0.017	0.247 ±0.002	100.0%	0.070 ±0.016	0.614 ±0.021	0.239 ±0.003	
RLVR-AvgProb	0.545	n/a	0.108 ±0.021	0.472 ±0.032	0.262 ±0.006	0.545	n/a	0.385 ±0.027	0.675 ±0.027	0.388 ±0.023
RLVR-ProbeMLP	0.545	n/a	0.265 ±0.025	0.497 ±0.033	0.321 ±0.015	0.545	n/a	0.099 ±0.012	0.710 ±0.021	0.223 ±0.010
RLVR-ConfTuner	0.551	n/a	0.179 ±0.025	0.654 ±0.027	0.258 ±0.012	0.554	n/a	0.065 ±0.012	0.718 ±0.025	0.213 ±0.009
RLVR-Verb	0.545	99.9%	0.429 ±0.027	0.592 ±0.023	0.428 ±0.025	0.545	99.7%	0.442 ±0.027	0.555 ±0.018	0.441 ±0.026
Base-RLCR	–	–	–	–	–	0.536	97.7%	0.395 ±0.027	0.594 ±0.017	0.397 ±0.024
Base-RLCC-L1	–	–	–	–	–	0.538	97.1%	0.414 ±0.026	0.596 ±0.024	0.412 ±0.025
Base-CoCA	0.536	100.0%	0.416 ±0.027	0.671 ±0.025	0.408 ±0.024	–	–	–	–	–
Base-CoCA (std off)	0.544	100.0%	0.278 ±0.027	0.621 ±0.026	0.315 ±0.018	–	–	–	–	–
Base-CALIBER	0.532	98.9%	0.091 ±0.019	0.653 ±0.027	0.235 ±0.007	0.532	98.9%	0.202 ±0.025	0.687 ±0.023	0.266 ±0.014
RLVR-PACE-$u^\star$	0.559	100.0%	0.113 ±0.023	0.633 ±0.028	0.246 ±0.010	100.0%	0.100 ±0.018	0.711 ±0.026	0.222 ±0.010	

Table 10: **Confidence calibration of different methods on LiveCodeBench** (OOD non-math, 400 questions).

	t=0					t=T				
method	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓
Base-AvgProb	0.614	n/a	0.065 ±0.018	0.522 ±0.024	0.242 ±0.007	0.614	n/a	0.252 ±0.020	0.748 ±0.017	0.289 ±0.015
Base-ProbeMLP	0.614	n/a	0.100 ±0.015	0.689 ±0.023	0.222 ±0.009	0.614	n/a	0.244 ±0.016	0.704 ±0.016	0.271 ±0.013
Base-Verb	0.614	99.8%	0.356 ±0.020	0.662 ±0.017	0.356 ±0.019	0.614	99.4%	0.362 ±0.020	0.705 ±0.017	0.358 ±0.019
Base-ConfTuner	0.625	n/a	0.080 ±0.012	0.814 ±0.016	0.171 ±0.008	0.636	n/a	0.055 ±0.007	0.820 ±0.015	0.163 ±0.008
Base-PACE-$u^\star$	0.623	100.0%	0.073 ±0.012	0.780 ±0.016	0.188 ±0.007	100.0%	0.076 ±0.010	0.824 ±0.015	0.170 ±0.007	
RLVR-AvgProb	0.648	n/a	0.049 ±0.014	0.517 ±0.026	0.230 ±0.008	0.648	n/a	0.296 ±0.020	0.789 ±0.017	0.307 ±0.018
RLVR-ProbeMLP	0.648	n/a	0.347 ±0.020	0.738 ±0.019	0.347 ±0.020	0.648	n/a	0.282 ±0.018	0.749 ±0.017	0.287 ±0.017
RLVR-ConfTuner	0.653	n/a	0.233 ±0.016	0.779 ±0.018	0.235 ±0.013	0.661	n/a	0.142 ±0.014	0.839 ±0.015	0.172 ±0.010
RLVR-Verb	0.648	100.0%	0.329 ±0.020	0.643 ±0.018	0.331 ±0.019	0.648	99.8%	0.329 ±0.020	0.707 ±0.017	0.327 ±0.019
Base-RLCR	–	–	–	–	–	0.644	94.9%	0.309 ±0.019	0.697 ±0.016	0.311 ±0.018
Base-RLCC-L1	–	–	–	–	–	0.638	97.2%	0.339 ±0.019	0.663 ±0.016	0.333 ±0.019
Base-CoCA	0.604	63.0%	0.289 ±0.018	0.704 ±0.020	0.282 ±0.016	–	–	–	–	–
Base-CoCA (std off)	0.654	69.7%	0.131 ±0.017	0.743 ±0.020	0.200 ±0.012	–	–	–	–	–
Base-CALIBER	0.655	98.8%	0.083 ±0.010	0.826 ±0.015	0.163 ±0.006	0.655	98.8%	0.095 ±0.011	0.827 ±0.015	0.166 ±0.007
RLVR-PACE-$u^\star$	0.656	100.0%	0.115 ±0.014	0.806 ±0.017	0.183 ±0.009	100.0%	0.102 ±0.012	0.846 ±0.014	0.165 ±0.008	

D CONFIDENCE CALIBRATION DOES NOT HURT REASONING

Table 11 asks whether training a confidence readout disturbs the solver it shares parameters with. For each PACE arm we recompute the per-question pass rate on that arm's own rollouts and compare it with the initialization's, so the two columns describe the same questions under the same grader. The paired shift is small and positive throughout, $+0.008$ on the macro average for both initializations and at most $+0.025$ on any single set (GPQA-Diamond, the smallest), and the per-question rates stay strongly aligned, macro $r = 0.926$ and 0.925. Because a pass rate averages eight Bernoulli draws, r cannot reach 1 even for an unchanged solver: the split-half reference puts its attainable value at 0.926 and 0.943, so the base pair sits *at* that reference and the RLVR pair within 2% of it. Aggregate reasoning is therefore preserved, though this is a distributional check and does not establish that prefix-conditional continuation probabilities are unchanged.

Table 11: **PACE preserves aggregate reasoning accuracy.** Per-question pass rate for each PACE arm against the initialization it was trained from, on each arm's own rollouts. Δ is the paired mean shift. *ceiling* is an estimated finite-sample reliability reference, from a Spearman–Brown-corrected split-half correlation of each arm's own draws, so r should be read against it rather than against 1.

dataset	n	pass@1 before	pass@1 after	Δ	r	ceiling
Qwen3-8B → Base-PACE-u						
deepmath	500	0.634	0.640	+0.006	0.867	0.923
math500	500	0.851	0.854	+0.003	0.941	0.916
gsm8k	1319	0.916	0.917	+0.001	0.960	0.948
olympiadbench	674	0.548	0.554	+0.006	0.950	0.945
commonsenseqa	1221	0.803	0.806	+0.003	0.947	0.939
gpqa-diamond	198	0.472	0.497	+0.025	0.890	0.890
livecodebench	400	0.614	0.623	+0.009	0.925	0.922
macro	–	0.691	0.699	+0.008	**0.926**	0.926
RLVR → RLVR-PACE-u						
deepmath	500	0.801	0.811	+0.010	0.917	0.944
math500	500	0.899	0.908	+0.009	0.912	0.926
gsm8k	1319	0.944	0.946	+0.002	0.928	0.962
olympiadbench	674	0.637	0.645	+0.008	0.949	0.960
commonsenseqa	1221	0.816	0.817	+0.001	0.953	0.948
gpqa-diamond	198	0.545	0.559	+0.014	0.900	0.925
livecodebench	400	0.648	0.656	+0.008	0.918	0.933
macro	–	0.756	0.763	+0.008	**0.925**	0.943

E ACROSS SCALE AND MODEL FAMILIES, PER DATASET

Tables 12–18 give the per-dataset metrics behind Figure 4. Conventions, once for all of them: columns are as in that figure, with parse rates at both positions; cells are mean ± one question-level bootstrap standard error; and GLM-4-9B's untrained row uses the reworded probe instruction described in Appendix B.1, while every other row, GLM's PACE-u included, uses the original.

The additional untrained Qwen3-1.7B, Qwen3-32B, and Llama-3.1-8B rows use the same question sets and task instructions as their trained counterparts, with each model generating its own eight solutions per question. Their metrics use the same forced endpoints and seed estimator. These metrics are conditional on parseable confidence; missing questions are not assigned zero scores.

Table 12: **Confidence calibration of different models on DeepMath-500**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.457	94.6%	0.469 ±0.017	0.570 ±0.016	0.476 ±0.016	99.8%	0.498 ±0.017	0.622 ±0.015	0.491 ±0.016
Qwen3-1.7B	PACE-u	0.452	100.0%	0.071 ±0.014	0.670 ±0.014	0.221 ±0.004	100.0%	0.070 ±0.010	0.748 ±0.013	0.198 ±0.004
Qwen3-8B†	Untrained	0.634	97.2%	0.347 ±0.017	0.558 ±0.017	0.352 ±0.017	97.2%	0.367 ±0.017	0.554 ±0.008	0.365 ±0.017
Qwen3-8B†	PACE-u	0.640	100.0%	0.076 ±0.012	0.746 ±0.017	0.195 ±0.007	100.0%	0.097 ±0.010	0.805 ±0.013	0.178 ±0.006
Qwen3-32B	Untrained	0.676	100.0%	0.283 ±0.017	0.555 ±0.012	0.297 ±0.015	99.9%	0.311 ±0.016	0.600 ±0.015	0.312 ±0.016
Qwen3-32B	PACE-u	0.704	100.0%	0.060 ±0.010	0.727 ±0.018	0.183 ±0.006	100.0%	0.067 ±0.009	0.791 ±0.016	0.164 ±0.006
Llama-3.1-8B	Untrained	0.270	100.0%	0.653 ±0.015	0.549 ±0.018	0.632 ±0.013	97.5%	0.603 ±0.014	0.635 ±0.014	0.577 ±0.013
Llama-3.1-8B	PACE-u	0.284	100.0%	0.025 ±0.007	0.626 ±0.015	0.183 ±0.007	100.0%	0.054 ±0.009	0.739 ±0.016	0.162 ±0.006
GLM-4-9B	Untrained	0.481	100.0%	0.503 ±0.017	0.504 ±0.010	0.502 ±0.016	100.0%	0.505 ±0.016	0.556 ±0.007	0.498 ±0.016
GLM-4-9B	PACE-u	0.468	100.0%	0.074 ±0.012	0.702 ±0.015	0.220 ±0.005	100.0%	0.065 ±0.010	0.779 ±0.013	0.189 ±0.005

Table 13: **Confidence calibration of different models on MATH-500**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.733	92.8%	0.254 ±0.016	0.616 ±0.019	0.256 ±0.016	99.4%	0.253 ±0.015	0.693 ±0.016	0.251 ±0.015
Qwen3-1.7B	PACE-u	0.735	100.0%	0.076 ±0.011	0.765 ±0.017	0.159 ±0.007	100.0%	0.097 ±0.009	0.844 ±0.014	0.134 ±0.007
Qwen3-8B†	Untrained	0.851	98.8%	0.126 ±0.012	0.693 ±0.015	0.139 ±0.012	98.3%	0.149 ±0.013	0.549 ±0.011	0.148 ±0.013
Qwen3-8B†	PACE-u	0.854	100.0%	0.061 ±0.008	0.845 ±0.017	0.099 ±0.006	100.0%	0.050 ±0.006	0.879 ±0.016	0.078 ±0.006
Qwen3-32B	Untrained	0.868	99.9%	0.097 ±0.011	0.641 ±0.011	0.122 ±0.010	99.9%	0.126 ±0.011	0.787 ±0.013	0.128 ±0.011
Qwen3-32B	PACE-u	0.872	100.0%	0.044 ±0.008	0.851 ±0.015	0.090 ±0.006	100.0%	0.033 ±0.004	0.914 ±0.013	0.067 ±0.006
Llama-3.1-8B	Untrained	0.458	99.8%	0.514 ±0.016	0.612 ±0.014	0.507 ±0.015	97.8%	0.492 ±0.015	0.653 ±0.010	0.481 ±0.015
Llama-3.1-8B	PACE-u	0.472	100.0%	0.063 ±0.012	0.728 ±0.014	0.203 ±0.007	100.0%	0.060 ±0.009	0.789 ±0.012	0.180 ±0.006
GLM-4-9B	Untrained	0.644	100.0%	0.347 ±0.017	0.542 ±0.009	0.349 ±0.017	100.0%	0.353 ±0.017	0.511 ±0.003	0.352 ±0.017
GLM-4-9B	PACE-u	0.661	100.0%	0.105 ±0.011	0.809 ±0.013	0.174 ±0.005	100.0%	0.077 ±0.006	0.883 ±0.009	0.131 ±0.005

Table 14: **Confidence calibration of different models on GSM8K**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.809	100.0%	0.183 ±0.008	0.553 ±0.011	0.185 ±0.008	100.0%	0.184 ±0.008	0.593 ±0.011	0.185 ±0.008
Qwen3-1.7B	PACE-u	0.808	100.0%	0.107 ±0.008	0.680 ±0.014	0.156 ±0.007	100.0%	0.095 ±0.007	0.715 ±0.013	0.138 ±0.006
Qwen3-8B†	Untrained	0.916	100.0%	0.066 ±0.007	0.623 ±0.019	0.080 ±0.006	99.9%	0.084 ±0.007	0.509 ±0.003	0.084 ±0.007
Qwen3-8B†	PACE-u	0.917	100.0%	0.031 ±0.006	0.671 ±0.019	0.074 ±0.005	100.0%	0.049 ±0.006	0.740 ±0.021	0.067 ±0.005
Qwen3-32B	Untrained	0.923	100.0%	0.041 ±0.006	0.604 ±0.010	0.072 ±0.006	100.0%	0.074 ±0.006	0.655 ±0.018	0.076 ±0.006
Qwen3-32B	PACE-u	0.924	100.0%	0.024 ±0.005	0.677 ±0.021	0.068 ±0.005	100.0%	0.052 ±0.006	0.726 ±0.019	0.067 ±0.006
Llama-3.1-8B	Untrained	0.756	99.9%	0.227 ±0.008	0.585 ±0.010	0.234 ±0.008	99.5%	0.237 ±0.008	0.545 ±0.005	0.236 ±0.008
Llama-3.1-8B	PACE-u	0.817	100.0%	0.240 ±0.008	0.610 ±0.013	0.228 ±0.005	100.0%	0.091 ±0.007	0.578 ±0.008	0.148 ±0.003
GLM-4-9B	Untrained	0.858	100.0%	0.133 ±0.007	0.556 ±0.008	0.138 ±0.007	100.0%	0.140 ±0.007	0.516 ±0.003	0.140 ±0.007
GLM-4-9B	PACE-u	0.864	100.0%	0.119 ±0.007	0.668 ±0.015	0.129 ±0.004	100.0%	0.037 ±0.005	0.724 ±0.014	0.103 ±0.004

Table 15: **Confidence calibration of different models on OlympiadBench**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.409	94.6%	0.528 ±0.015	0.598 ±0.014	0.521 ±0.015	99.5%	0.546 ±0.015	0.658 ±0.012	0.537 ±0.015
Qwen3-1.7B	PACE-u	0.404	100.0%	0.103 ±0.013	0.716 ±0.013	0.209 ±0.004	100.0%	0.107 ±0.007	0.829 ±0.010	0.173 ±0.005
Qwen3-8B†	Untrained	0.548	98.8%	0.411 ±0.016	0.570 ±0.011	0.414 ±0.015	98.6%	0.451 ±0.016	0.555 ±0.006	0.449 ±0.016
Qwen3-8B†	PACE-u	0.554	100.0%	0.139 ±0.012	0.817 ±0.012	0.189 ±0.005	100.0%	0.169 ±0.011	0.876 ±0.010	0.176 ±0.006
Qwen3-32B	Untrained	0.574	100.0%	0.380 ±0.016	0.562 ±0.008	0.386 ±0.015	99.9%	0.411 ±0.016	0.719 ±0.010	0.406 ±0.016
Qwen3-32B	PACE-u	0.580	100.0%	0.092 ±0.011	0.791 ±0.014	0.192 ±0.006	100.0%	0.106 ±0.009	0.872 ±0.011	0.157 ±0.006
Llama-3.1-8B	Untrained	0.161	99.9%	0.781 ±0.010	0.627 ±0.015	0.749 ±0.010	97.1%	0.706 ±0.010	0.709 ±0.009	0.674 ±0.010
Llama-3.1-8B	PACE-u	0.162	100.0%	0.087 ±0.009	0.590 ±0.014	0.136 ±0.005	100.0%	0.114 ±0.007	0.765 ±0.014	0.125 ±0.004
GLM-4-9B	Untrained	0.318	100.0%	0.670 ±0.014	0.526 ±0.007	0.666 ±0.013	100.0%	0.674 ±0.014	0.520 ±0.002	0.670 ±0.014
GLM-4-9B	PACE-u	0.335	100.0%	0.081 ±0.010	0.770 ±0.013	0.173 ±0.005	100.0%	0.101 ±0.006	0.866 ±0.009	0.137 ±0.004

Table 16: **Confidence calibration of different models on CommonsenseQA**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.700	95.9%	0.295 ±0.011	0.548 ±0.011	0.298 ±0.011	99.8%	0.266 ±0.011	0.588 ±0.012	0.277 ±0.010
Qwen3-1.7B	PACE-u	0.697	100.0%	0.203 ±0.011	0.547 ±0.013	0.258 ±0.004	100.0%	0.302 ±0.011	0.600 ±0.011	0.295 ±0.003
Qwen3-8B†	Untrained	0.803	99.7%	0.162 ±0.009	0.530 ±0.012	0.184 ±0.009	99.9%	0.184 ±0.009	0.598 ±0.012	0.189 ±0.009
Qwen3-8B†	PACE-u	0.806	100.0%	0.164 ±0.009	0.565 ±0.014	0.184 ±0.003	100.0%	0.185 ±0.009	0.666 ±0.011	0.182 ±0.002
Qwen3-32B	Untrained	0.842	100.0%	0.099 ±0.008	0.580 ±0.013	0.142 ±0.007	100.0%	0.132 ±0.008	0.693 ±0.012	0.147 ±0.008
Qwen3-32B	PACE-u	0.840	100.0%	0.142 ±0.008	0.616 ±0.015	0.158 ±0.004	100.0%	0.044 ±0.007	0.748 ±0.013	0.123 ±0.005
Llama-3.1-8B	Untrained	0.710	99.9%	0.220 ±0.010	0.562 ±0.011	0.251 ±0.009	98.6%	0.230 ±0.010	0.629 ±0.011	0.245 ±0.009
Llama-3.1-8B	PACE-u	0.709	100.0%	0.226 ±0.009	0.598 ±0.011	0.271 ±0.004	100.0%	0.215 ±0.009	0.659 ±0.010	0.244 ±0.004
GLM-4-9B	Untrained	0.795	100.0%	0.115 ±0.009	0.558 ±0.012	0.183 ±0.007	100.0%	0.176 ±0.009	0.591 ±0.010	0.188 ±0.008
GLM-4-9B	PACE-u	0.805	100.0%	0.379 ±0.010	0.586 ±0.015	0.310 ±0.004	100.0%	0.287 ±0.009	0.634 ±0.012	0.233 ±0.003

Table 17: **Confidence calibration of different models on GPQA-Diamond**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.305	91.5%	0.659 ±0.024	0.450 ±0.026	0.655 ±0.023	99.6%	0.620 ±0.021	0.511 ±0.021	0.600 ±0.019
Qwen3-1.7B	PACE-u	0.284	100.0%	0.055 ±0.012	0.498 ±0.019	0.210 ±0.008	100.0%	0.035 ±0.010	0.482 ±0.010	0.206 ±0.008
Qwen3-8B†	Untrained	0.472	99.5%	0.498 ±0.027	0.582 ±0.021	0.494 ±0.025	99.5%	0.516 ±0.027	0.542 ±0.019	0.514 ±0.027
Qwen3-8B†	PACE-u	0.497	100.0%	0.039 ±0.014	0.543 ±0.017	0.247 ±0.002	100.0%	0.070 ±0.016	0.614 ±0.021	0.239 ±0.003
Qwen3-32B	Untrained	0.565	100.0%	0.384 ±0.027	0.540 ±0.011	0.391 ±0.024	100.0%	0.411 ±0.027	0.630 ±0.019	0.410 ±0.025
Qwen3-32B	PACE-u	0.582	100.0%	0.103 ±0.020	0.579 ±0.028	0.249 ±0.005	100.0%	0.092 ±0.016	0.660 ±0.026	0.231 ±0.006
Llama-3.1-8B	Untrained	0.242	98.6%	0.657 ±0.018	0.524 ±0.024	0.622 ±0.016	98.8%	0.567 ±0.019	0.600 ±0.022	0.536 ±0.017
Llama-3.1-8B	PACE-u	0.254	100.0%	0.015 ±0.008	0.505 ±0.005	0.189 ±0.009	100.0%	0.022 ±0.009	0.528 ±0.015	0.188 ±0.009
GLM-4-9B	Untrained	0.374	100.0%	0.538 ±0.023	0.504 ±0.018	0.527 ±0.019	100.0%	0.536 ±0.023	0.526 ±0.020	0.524 ±0.020
GLM-4-9B	PACE-u	0.367	100.0%	0.123 ±0.021	0.514 ±0.018	0.245 ±0.010	100.0%	0.094 ±0.019	0.590 ±0.019	0.231 ±0.009

Table 18: **Confidence calibration of different models on LiveCodeBench**.

			$t=0$				$t=T$			
Backbone	Readout	pass@1	parse	ECE ↓	AUROC ↑	Brier ↓	parse	ECE ↓	AUROC ↑	Brier ↓
Qwen3-1.7B	Untrained	0.402	99.5%	0.588 ±0.021	0.596 ±0.013	0.583 ±0.020	99.9%	0.580 ±0.022	0.574 ±0.020	0.573 ±0.021
Qwen3-1.7B	PACE-u	0.395	100.0%	0.056 ±0.013	0.722 ±0.020	0.201 ±0.008	100.0%	0.092 ±0.016	0.750 ±0.019	0.184 ±0.008
Qwen3-8B†	Untrained	0.614	99.8%	0.356 ±0.020	0.662 ±0.017	0.356 ±0.019	99.4%	0.362 ±0.020	0.705 ±0.017	0.358 ±0.019
Qwen3-8B†	PACE-u	0.623	100.0%	0.073 ±0.012	0.780 ±0.016	0.188 ±0.007	100.0%	0.076 ±0.010	0.824 ±0.015	0.170 ±0.007
Qwen3-32B	Untrained	0.692	100.0%	0.277 ±0.019	0.618 ±0.017	0.287 ±0.018	100.0%	0.290 ±0.019	0.611 ±0.014	0.294 ±0.018
Qwen3-32B	PACE-u	0.686	100.0%	0.065 ±0.010	0.772 ±0.018	0.177 ±0.008	100.0%	0.092 ±0.010	0.818 ±0.015	0.164 ±0.006
Llama-3.1-8B	Untrained	0.187	100.0%	0.810 ±0.015	0.548 ±0.008	0.807 ±0.015	98.7%	0.807 ±0.015	0.572 ±0.011	0.803 ±0.015
Llama-3.1-8B	PACE-u	0.203	100.0%	0.175 ±0.013	0.758 ±0.021	0.161 ±0.007	100.0%	0.205 ±0.012	0.796 ±0.018	0.167 ±0.007
GLM-4-9B	Untrained	0.381	100.0%	0.596 ±0.021	0.575 ±0.013	0.588 ±0.020	100.0%	0.617 ±0.021	0.507 ±0.002	0.616 ±0.021
GLM-4-9B	PACE-u	0.381	100.0%	0.052 ±0.011	0.781 ±0.018	0.176 ±0.007	100.0%	0.095 ±0.013	0.803 ±0.017	0.174 ±0.006

Table 19: **Different RL algorithms on DeepMath-500.**

		t=0			t=T			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	t=0	t=T
RLOO	0.632	**0.050** ±.009	**0.746** ±.016	**0.192** ±.006	**0.055** ±.010	0.795 ±.013	**0.175** ±.006	17.2	22.3
REINFORCE++	0.629	0.063 ±.009	0.726 ±.017	0.199 ±.007	0.063 ±.010	0.789 ±.014	0.178 ±.005	23.1	16.9
PPO	0.640	0.067 ±.013	0.736 ±.015	0.196 ±.005	0.100 ±.012	0.785 ±.013	0.183 ±.005	5.5	12.0
GRPO, std on (G=2)	0.647	0.317 ±.015	0.608 ±.014	0.317 ±.015	0.251 ±.014	0.671 ±.014	0.251 ±.014	100.0	99.7
GRPO, std on	0.658	0.217 ±.014	0.642 ±.016	0.256 ±.013	0.216 ±.013	0.739 ±.015	0.218 ±.012	74.6	75.6
GRPO, std off*	0.641	0.076 ±.012	0.746 ±.017	0.195 ±.007	0.097 ±.010	**0.805** ±.013	0.178 ±.006	25.9	33.7

Table 20: **Different RL algorithms on MATH-500.**

		t=0			t=T			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	t=0	t=T
RLOO	0.854	0.080 ±.008	0.839 ±.018	0.101 ±.005	**0.038** ±.005	0.891 ±.012	**0.077** ±.005	48.2	67.7
REINFORCE++	0.857	0.075 ±.008	0.832 ±.020	**0.091** ±.006	0.071 ±.006	**0.901** ±.012	0.079 ±.005	50.8	53.4
PPO	0.849	0.101 ±.008	**0.846** ±.015	0.107 ±.005	0.072 ±.006	0.900 ±.012	0.082 ±.005	24.9	54.6
GRPO, std on (G=2)	0.849	0.139 ±.011	0.583 ±.017	0.139 ±.011	0.112 ±.010	0.647 ±.018	0.112 ±.010	99.3	100.0
GRPO, std on	0.855	0.094 ±.009	0.704 ±.024	0.110 ±.009	0.096 ±.009	0.730 ±.020	0.092 ±.008	86.5	92.1
GRPO, std off*	0.854	**0.061** ±.008	0.845 ±.017	0.099 ±.006	0.050 ±.006	0.879 ±.016	0.078 ±.006	57.6	77.4

F DIFFERENT RL ALGORITHMS, PER DATASET

Table 2 macro-averages the seven test sets in both blocks; Tables 19–32 report each dataset separately. Conventions follow the main text: same forced positions and seed estimator, with each point estimate followed by its question-clustered bootstrap standard error in smaller type. Rows, row order, and columns follow the corresponding block of Table 2; bold marks the best arm per column within each sweep *on that dataset*. * denotes our method. Pass@1 (per row) and *Extremal %* (the share of reports with $c \leq 0.05$ or $c \geq 0.95$, pooled over all draws for that arm, dataset, and position) carry no standard error. Extremal % is a degeneracy diagnostic, not a quantity to minimize: a readout collapsed onto the endpoints can still score a low ECE when the base rate sits near them, and because $V_T = Z$ is two-valued, a correct readout is expected to rise somewhat from $t = 0$ to $t = T$.

The macro row hides two things. First, the objective ranking varies by dataset: the standardized arms have the worst ECE on all seven sets at $t = 0$ and on six of seven at $t = T$, yet the $G = 8$ standardized arm has the best AUROC of the six on CommonsenseQA at both positions. Its penalty thus falls on probability level rather than ranking, with domain-dependent effects. Second, PPO is near-constant on GPQA-Diamond (1.75/2.00 distinct values, mean confidence $0.501/0.507$, AUROC $0.506/0.541$ at $t = 0/t = T$) while remaining competitive elsewhere. An arm that looks ordinary in the macro row therefore has one cell where the readout has essentially stopped varying. For this reason, extremal share must be read alongside the distinct-value count. PPO has the *lowest* extremal share of any objective at $t = 0$ (11.6; 0.0 on GPQA-Diamond) and the second lowest at $t = T$ (26.5 vs. REINFORCE++'s 26.0). Extremal % detects endpoint degeneracy, which standardization produces, but not interior degeneracy, which a learned critic produced here.

Table 21: **Different RL algorithms on GSM8K.**

		$t{=}0$			$t{=}T$			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
RLOO	0.915	**0.015** ±.004	0.687 ±.020	**0.073** ±.005	**0.015** ±.003	0.736 ±.018	0.065 ±.005	80.0	88.2
REINFORCE++	0.916	0.055 ±.005	0.713 ±.018	0.074 ±.005	0.046 ±.005	0.747 ±.019	0.067 ±.005	70.2	79.0
PPO	0.918	0.060 ±.006	**0.721** ±.016	0.075 ±.004	0.025 ±.004	**0.777** ±.019	**0.062** ±.005	35.0	78.0
GRPO, std on ($G{=}2$)	0.919	0.085 ±.007	0.538 ±.011	0.085 ±.007	0.077 ±.006	0.538 ±.008	0.077 ±.006	99.9	100.0
GRPO, std on	0.918	0.081 ±.007	0.513 ±.007	0.082 ±.007	0.078 ±.007	0.537 ±.008	0.078 ±.007	99.2	99.2
GRPO, std off*	0.917	0.031 ±.006	0.671 ±.019	0.074 ±.005	0.049 ±.006	0.740 ±.021	0.067 ±.005	87.4	94.8

Table 22: **Different RL algorithms on OlympiadBench.**

		$t{=}0$			$t{=}T$			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
RLOO	0.557	**0.106** ±.011	0.803 ±.014	0.188 ±.005	**0.128** ±.009	0.870 ±.010	**0.163** ±.006	12.2	30.7
REINFORCE++	0.558	0.119 ±.011	0.811 ±.014	**0.185** ±.005	0.140 ±.010	0.860 ±.010	0.166 ±.005	11.0	19.5
PPO	0.554	0.150 ±.012	0.790 ±.013	0.201 ±.004	0.149 ±.009	0.863 ±.010	0.181 ±.005	2.8	19.7
GRPO, std on ($G{=}2$)	0.542	0.355 ±.014	0.622 ±.013	0.354 ±.014	0.268 ±.012	0.711 ±.010	0.268 ±.012	99.1	99.8
GRPO, std on	0.557	0.203 ±.013	0.734 ±.014	0.241 ±.011	0.267 ±.012	0.783 ±.011	0.241 ±.011	60.6	70.1
GRPO, std off*	0.554	0.139 ±.012	**0.817** ±.012	0.189 ±.005	0.169 ±.011	**0.876** ±.010	0.176 ±.006	19.4	40.1

Table 23: **Different RL algorithms on CommonsenseQA.**

		$t{=}0$			$t{=}T$			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
RLOO	0.802	0.175 ±.009	0.600 ±.014	0.189 ±.003	0.149 ±.009	0.735 ±.012	0.163 ±.003	4.2	5.5
REINFORCE++	0.804	0.288 ±.009	0.580 ±.012	0.240 ±.002	0.256 ±.009	0.679 ±.010	0.211 ±.002	0.1	0.0
PPO	0.804	0.216 ±.009	0.602 ±.014	0.202 ±.002	0.181 ±.009	0.692 ±.011	0.179 ±.003	0.3	0.5
GRPO, std on ($G{=}2$)	0.794	0.365 ±.012	0.579 ±.014	0.373 ±.012	0.173 ±.008	0.726 ±.012	0.197 ±.008	95.8	90.2
GRPO, std on	0.806	**0.160** ±.009	**0.632** ±.016	0.187 ±.007	**0.096** ±.008	**0.749** ±.013	**0.144** ±.007	50.0	65.7
GRPO, std off*	0.806	0.164 ±.009	0.565 ±.014	**0.184** ±.003	0.185 ±.009	0.666 ±.011	0.182 ±.002	5.7	4.7

Table 24: **Different RL algorithms on GPQA-Diamond.**

		$t{=}0$			$t{=}T$			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
RLOO	0.475	0.062 ±.012	0.543 ±.025	0.249 ±.003	0.068 ±.012	0.651 ±.020	**0.230** ±.004	0.0	0.0
REINFORCE++	0.490	0.102 ±.022	0.539 ±.021	0.258 ±.006	0.065 ±.019	0.612 ±.020	0.241 ±.004	0.0	0.0
PPO	0.501	**0.022** ±.011	0.506 ±.003	0.250 ±.000	**0.042** ±.013	0.541 ±.014	0.247 ±.001	0.0	0.0
GRPO, std on ($G{=}2$)	0.491	0.408 ±.024	**0.559** ±.020	0.419 ±.024	0.297 ±.021	**0.668** ±.024	0.321 ±.019	92.9	71.1
GRPO, std on	0.512	0.290 ±.024	0.516 ±.030	0.342 ±.017	0.155 ±.017	0.659 ±.024	0.258 ±.013	6.1	18.6
GRPO, std off*	0.497	0.039 ±.014	0.543 ±.017	**0.247** ±.002	0.070 ±.016	0.614 ±.021	0.239 ±.003	0.0	0.0

Table 25: **Different RL algorithms on LiveCodeBench.**

		$t{=}0$			$t{=}T$			extremal %	
objective	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
RLOO	0.620	0.089 ±.013	0.796 ±.017	0.185 ±.008	0.104 ±.012	0.835 ±.014	0.170 ±.008	28.8	32.0
REINFORCE++	0.626	0.083 ±.012	**0.803** ±.015	**0.180** ±.007	0.131 ±.011	**0.853** ±.014	**0.163** ±.006	18.6	13.2
PPO	0.629	0.096 ±.012	0.799 ±.015	0.183 ±.006	0.107 ±.011	0.833 ±.014	0.174 ±.007	12.6	20.6
GRPO, std on ($G{=}2$)	0.620	0.258 ±.016	0.694 ±.016	0.262 ±.015	0.224 ±.014	0.734 ±.015	0.230 ±.014	95.6	96.4
GRPO, std on	0.619	0.185 ±.015	0.756 ±.017	0.217 ±.012	0.173 ±.014	0.778 ±.016	0.201 ±.012	58.2	61.5
GRPO, std off*	0.623	**0.073** ±.012	0.780 ±.016	0.188 ±.007	**0.076** ±.010	0.824 ±.015	0.170 ±.007	25.9	31.3

Table 26: **Group size on DeepMath-500.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.625	0.050 ± .009	0.733 ± .016	0.200 ± .005	0.087 ± .010	0.795 ± .014	0.187 ± .007	7.9	14.1
$G{=}4$	0.634	0.060 ± .013	0.724 ± .016	0.198 ± .007	**0.063** ± .010	0.779 ± .013	0.182 ± .006	22.1	21.8
$G{=}8^\star$	0.641	0.076 ± .012	**0.746** ± .017	**0.195** ± .007	0.097 ± .010	**0.805** ± .013	**0.178** ± .006	25.9	33.7
$G{=}16$	0.647	**0.032** ± .009	0.715 ± .016	0.196 ± .006	0.113 ± .013	0.783 ± .013	0.185 ± .008	22.4	40.1

Table 27: **Group size on MATH-500.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.853	0.071 ± .009	0.835 ± .017	0.100 ± .006	0.059 ± .006	**0.890** ± .012	0.082 ± .006	39.3	52.1
$G{=}4$	0.854	0.039 ± .007	0.806 ± .017	0.100 ± .006	0.053 ± .007	0.877 ± .013	0.083 ± .005	53.2	60.7
$G{=}8^\star$	0.854	0.061 ± .008	**0.845** ± .017	**0.099** ± .006	**0.050** ± .006	0.879 ± .016	**0.078** ± .006	57.6	77.4
$G{=}16$	0.848	**0.037** ± .006	0.827 ± .017	0.103 ± .006	0.082 ± .008	0.861 ± .016	0.085 ± .007	55.3	81.9

G GROUP SIZE, PER DATASET

Rows and their order follow the corresponding block of Table 2; columns match it; each cell is the point estimate followed by its question-clustered bootstrap standard error, set a size smaller; bold is the best arm in that column *on that dataset*; pass@1 is per row and carries no standard error. * denotes our proposed method.

Conventions as in §F. The group-size sweep has no consistent direction: on point estimates no test set moves monotonically in G in more than one of the six metric columns, and five of the seven move monotonically in none. Its macro flatness is partly an averaging effect, however. The $t = T$ AUROC spread across $G \in \{2, 4, 8, 16\}$ is 0.028 on the macro average but 0.104 on GSM8K and 0.098 on GPQA-Diamond, against 0.012 on LiveCodeBench, so G is close to immaterial on average while individual cells move several times as much. The tables are given so that the choice of $G = 8$ can be checked against its alternatives rather than taken on the macro average alone.

Table 28: **Group size on GSM8K.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.913	0.015 ± .003	0.691 ± .019	0.076 ± .005	**0.020** ± .004	0.721 ± .019	0.070 ± .005	67.2	82.2
$G{=}4$	0.916	**0.014** ± .004	0.689 ± .019	0.073 ± .005	0.021 ± .005	**0.785** ± .017	**0.064** ± .004	79.2	78.0
$G{=}8^\star$	0.917	0.031 ± .006	0.671 ± .019	0.074 ± .005	0.049 ± .006	0.740 ± .021	0.067 ± .005	87.4	94.8
$G{=}16$	0.917	0.030 ± .005	**0.711** ± .018	**0.072** ± .005	0.067 ± .006	0.681 ± .019	0.071 ± .006	89.2	95.5

Table 29: **Group size on OlympiadBench.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.552	0.134 ± .011	**0.818** ± .013	0.191 ± .005	0.161 ± .011	**0.876** ± .010	0.179 ± .006	6.0	15.5
$G{=}4$	0.552	**0.121** ± .011	0.791 ± .013	0.199 ± .006	**0.146** ± .011	0.845 ± .011	0.186 ± .006	16.3	24.7
$G{=}8^\star$	0.554	0.139 ± .012	0.817 ± .012	**0.189** ± .005	0.169 ± .011	0.876 ± .010	**0.176** ± .006	19.4	40.1
$G{=}16$	0.550	0.125 ± .013	0.759 ± .013	0.203 ± .005	0.225 ± .011	0.862 ± .010	0.197 ± .008	16.8	46.8

Table 30: **Group size on CommonsenseQA.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.802	**0.072** ± .008	**0.600** ± .014	**0.161** ± .006	**0.025** ± .006	0.720 ± .012	**0.141** ± .006	24.9	24.6
$G{=}4$	0.802	0.151 ± .009	0.593 ± .013	0.187 ± .004	0.123 ± .009	0.687 ± .012	0.159 ± .004	8.7	4.5
$G{=}8^\star$	0.806	0.164 ± .009	0.565 ± .014	0.184 ± .003	0.185 ± .009	0.666 ± .011	0.182 ± .002	5.7	4.7
$G{=}16$	0.807	0.216 ± .009	0.589 ± .013	0.206 ± .003	0.134 ± .009	**0.724** ± .012	0.160 ± .004	7.3	16.8

Table 31: **Group size on GPQA-Diamond.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.495	0.112 ± .024	0.538 ± .022	0.259 ± .006	0.112 ± .019	**0.641** ± .020	0.245 ± .008	0.1	0.9
$G{=}4$	0.496	0.036 ± .013	0.518 ± .016	0.253 ± .003	0.057 ± .014	0.588 ± .019	0.243 ± .005	0.0	0.1
$G{=}8^\star$	0.497	0.039 ± .014	**0.543** ± .017	**0.247** ± .002	0.070 ± .016	0.614 ± .021	**0.239** ± .003	0.0	0.0
$G{=}16$	0.491	**0.020** ± .012	0.504 ± .002	0.249 ± .000	**0.037** ± .016	0.543 ± .014	0.244 ± .003	0.0	0.1

Table 32: **Group size on LiveCodeBench.**

		$t{=}0$			$t{=}T$			extremal %	
group size	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	$t{=}0$	$t{=}T$
$G{=}2$	0.621	0.081 ± .013	**0.798** ± .016	**0.184** ± .007	0.101 ± .012	**0.834** ± .014	0.175 ± .007	21.0	20.7
$G{=}4$	0.630	0.127 ± .017	0.774 ± .016	0.199 ± .009	0.092 ± .012	0.822 ± .015	0.174 ± .008	32.8	29.2
$G{=}8^\star$	0.623	**0.073** ± .012	0.780 ± .016	0.188 ± .007	**0.076** ± .010	0.824 ± .015	**0.170** ± .007	25.9	31.3
$G{=}16$	0.629	0.103 ± .014	0.788 ± .017	0.189 ± .008	0.116 ± .012	0.827 ± .015	0.175 ± .008	32.5	34.2

Table 33: **Position sampling rules on DeepMath-500.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.622	**0.042** ±.009	0.715 ±.016	0.201 ±.006	**0.050** ±.011	0.703 ±.015	0.205 ±.006	0.205 ±.015	0.750 ±.015	0.229 ±.010
$u{=}\frac{1}{2}$	0.634	0.061 ±.011	0.743 ±.016	0.195 ±.006	0.064 ±.009	**0.788** ±.014	**0.179** ±.005	0.079 ±.010	0.819 ±.013	0.171 ±.005
$u{=}1$	0.687	0.254 ±.015	0.691 ±.018	0.259 ±.006	0.279 ±.015	0.699 ±.015	0.270 ±.006	0.093 ±.011	**0.828** ±.013	**0.158** ±.005
grid_2	0.652	0.065 ±.011	0.735 ±.017	0.195 ±.005	0.130 ±.014	0.736 ±.016	0.208 ±.005	**0.055** ±.008	0.800 ±.013	0.170 ±.006
grid_3	0.634	0.052 ±.010	0.744 ±.016	**0.194** ±.005	0.070 ±.009	0.779 ±.014	0.183 ±.006	0.073 ±.010	0.798 ±.014	0.177 ±.006
grid_9	0.635	0.061 ±.010	0.744 ±.016	0.195 ±.006	0.061 ±.010	0.763 ±.015	0.189 ±.007	0.080 ±.010	0.795 ±.014	0.180 ±.007
$U[0,1]^\star$	0.641	0.076 ±.012	**0.746** ±.017	0.195 ±.007	0.075 ±.010	0.767 ±.015	0.188 ±.006	0.097 ±.010	0.805 ±.013	0.178 ±.006

Table 34: **Position sampling rules on MATH-500.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.853	0.075 ±.007	0.854 ±.015	0.101 ±.006	0.070 ±.007	0.859 ±.014	0.095 ±.005	0.068 ±.009	0.799 ±.019	0.097 ±.008
$u{=}\frac{1}{2}$	0.849	0.088 ±.008	0.855 ±.014	0.103 ±.005	0.063 ±.006	**0.895** ±.012	0.085 ±.005	0.062 ±.006	0.908 ±.012	0.078 ±.005
$u{=}1$	0.858	0.304 ±.011	0.827 ±.016	0.200 ±.006	0.328 ±.011	0.839 ±.013	0.215 ±.006	0.044 ±.006	**0.912** ±.011	**0.073** ±.005
grid_2	0.847	0.109 ±.009	**0.859** ±.014	0.110 ±.005	0.155 ±.009	0.886 ±.012	0.120 ±.005	0.061 ±.006	0.896 ±.014	0.075 ±.005
grid_3	0.851	0.082 ±.008	0.842 ±.016	0.102 ±.005	0.055 ±.006	0.860 ±.017	0.088 ±.006	0.049 ±.006	0.878 ±.015	0.079 ±.006
grid_9	0.852	**0.057** ±.008	0.844 ±.017	**0.098** ±.006	**0.036** ±.006	0.876 ±.014	**0.084** ±.006	**0.043** ±.006	0.898 ±.013	0.078 ±.006
$U[0,1]^\star$	0.854	0.061 ±.008	0.845 ±.017	0.099 ±.006	0.057 ±.007	0.885 ±.014	0.087 ±.005	0.050 ±.006	0.879 ±.016	0.078 ±.006

H DIFFERENT TOKEN-POSITION SAMPLING RULES, PER DATASET

Table 3 macro-averages the seven test sets; Tables 33–39 give each dataset its own breakdown. The position rules, forced evaluation positions, and bootstrap estimator match Table 3. Each cell gives a point estimate and question-clustered standard error; bold marks the best rule within that dataset. Differences in the best-performing rule reflect domain variation as well as sampling uncertainty.

The per-dataset view also bounds how much the macro average hides. Worst-case ECE over the 21 dataset-by-position cells is 0.203 for grid_3 and 0.253 for continuous u, against 0.353 for the $u = 1$ specialist — so the ordering in the main text survives disaggregation, but every rule has at least one cell six to seventeen times its own best cell. Those cells are concentrated in the two hardest non-mathematical sets rather than spread evenly, which is the same pattern Table 1 shows for the elicitation families.

Table 35: **Position sampling rules on GSM8K.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.915	0.021 ±.004	0.693 ±.020	0.075 ±.005	0.027 ±.004	0.673 ±.019	0.076 ±.005	0.030 ±.006	0.569 ±.012	0.074 ±.006
$u{=}\frac{1}{2}$	0.917	0.019 ±.005	0.720 ±.020	**0.071** ±.005	0.030 ±.005	**0.767** ±.017	**0.069** ±.005	0.027 ±.005	**0.798** ±.018	0.064 ±.005
$u{=}1$	0.921	0.227 ±.007	0.736 ±.019	0.129 ±.003	0.303 ±.007	0.689 ±.016	0.171 ±.003	0.040 ±.005	0.789 ±.019	**0.063** ±.005
grid_2	0.918	0.083 ±.006	**0.755** ±.019	0.077 ±.004	0.101 ±.006	0.760 ±.017	0.083 ±.004	0.047 ±.005	0.719 ±.019	0.066 ±.005
grid_3	0.915	**0.012** ±.004	0.656 ±.022	0.073 ±.005	**0.018** ±.006	0.629 ±.016	0.072 ±.005	**0.020** ±.006	0.647 ±.017	0.068 ±.005
grid_9	0.917	0.016 ±.005	0.622 ±.020	0.072 ±.005	0.028 ±.006	0.613 ±.013	0.072 ±.006	0.039 ±.006	0.695 ±.016	0.069 ±.006
$U[0,1]^\star$	0.917	0.031 ±.006	0.671 ±.019	0.074 ±.005	0.040 ±.005	0.751 ±.017	0.071 ±.005	0.049 ±.006	0.740 ±.021	0.067 ±.005

Table 36: **Position sampling rules on OlympiadBench.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.543	0.123 ±.012	0.789 ±.013	0.194 ±.005	0.151 ±.011	0.796 ±.012	0.199 ±.005	0.289 ±.013	0.822 ±.011	0.260 ±.009
$u{=}\frac{1}{2}$	0.547	0.142 ±.011	**0.817** ±.013	0.188 ±.005	**0.107** ±.009	**0.862** ±.011	**0.159** ±.005	0.131 ±.009	0.880 ±.010	0.159 ±.005
$u{=}1$	0.558	0.182 ±.014	0.749 ±.014	0.242 ±.005	0.177 ±.014	0.750 ±.013	0.238 ±.005	**0.097** ±.007	**0.881** ±.010	**0.148** ±.005
grid_2	0.554	0.130 ±.012	0.810 ±.013	0.187 ±.004	0.146 ±.011	0.833 ±.012	0.186 ±.005	0.113 ±.009	0.878 ±.010	0.153 ±.005
grid_3	0.553	**0.107** ±.011	0.813 ±.013	**0.183** ±.005	0.131 ±.010	0.849 ±.011	0.168 ±.005	0.131 ±.009	0.867 ±.010	0.169 ±.006
grid_9	0.562	0.129 ±.012	0.817 ±.013	0.185 ±.005	0.122 ±.009	0.854 ±.011	0.172 ±.006	0.149 ±.010	0.878 ±.010	0.169 ±.006
$U[0,1]^\star$	0.554	0.139 ±.012	0.817 ±.012	0.189 ±.005	0.121 ±.010	0.853 ±.011	0.167 ±.005	0.169 ±.011	0.876 ±.010	0.176 ±.006

Table 37: **Position sampling rules on CommonsenseQA.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.805	0.176 ±.009	0.602 ±.014	0.192 ±.004	**0.161** ±.009	0.623 ±.013	0.186 ±.004	**0.052** ±.008	**0.741** ±.012	**0.141** ±.006
$u{=}\frac{1}{2}$	0.802	0.216 ±.010	0.556 ±.012	0.204 ±.002	0.259 ±.009	0.622 ±.013	0.220 ±.001	0.208 ±.009	0.621 ±.009	0.193 ±.002
$u{=}1$	0.808	0.272 ±.010	0.574 ±.015	0.233 ±.003	0.353 ±.010	0.566 ±.014	0.286 ±.003	0.181 ±.009	0.727 ±.013	0.172 ±.003
grid_2	0.802	0.303 ±.010	0.536 ±.008	0.249 ±.001	0.314 ±.010	0.568 ±.011	0.255 ±.001	0.211 ±.009	0.735 ±.012	0.186 ±.003
grid_3	0.805	0.198 ±.009	0.583 ±.014	0.194 ±.003	0.203 ±.009	0.630 ±.014	0.193 ±.002	0.157 ±.009	0.703 ±.013	0.170 ±.003
grid_9	0.804	0.187 ±.009	**0.609** ±.014	0.190 ±.003	0.174 ±.009	**0.631** ±.014	**0.186** ±.003	0.075 ±.007	0.740 ±.012	0.146 ±.004
$U[0,1]^\star$	0.806	**0.164** ±.009	0.565 ±.014	**0.184** ±.003	0.253 ±.009	0.617 ±.013	0.215 ±.001	0.185 ±.009	0.666 ±.011	0.182 ±.002

Table 38: **Position sampling rules on GPQA-Diamond.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.484	0.041 ±.014	0.525 ±.012	0.249 ±.002	0.076 ±.017	0.550 ±.018	0.252 ±.005	0.267 ±.024	0.642 ±.021	0.305 ±.015
$u{=}\frac{1}{2}$	0.492	0.054 ±.012	**0.587** ±.023	0.245 ±.004	0.090 ±.021	0.596 ±.025	0.250 ±.006	0.060 ±.013	0.660 ±.023	0.232 ±.004
$u{=}1$	0.496	0.140 ±.026	0.513 ±.025	0.268 ±.008	0.178 ±.025	0.522 ±.017	0.281 ±.010	0.120 ±.022	0.663 ±.023	0.243 ±.007
grid_2	0.504	0.087 ±.021	0.551 ±.026	0.256 ±.005	0.127 ±.023	0.585 ±.023	0.259 ±.006	0.103 ±.019	**0.666** ±.022	0.238 ±.004
grid_3	0.500	0.073 ±.017	0.575 ±.027	0.249 ±.004	0.066 ±.017	0.615 ±.024	0.243 ±.004	0.067 ±.014	0.646 ±.022	0.235 ±.004
grid_9	0.480	0.055 ±.015	0.587 ±.023	**0.245** ±.003	0.067 ±.015	**0.653** ±.023	**0.234** ±.004	**0.050** ±.012	0.656 ±.021	**0.231** ±.004
$U[0,1]^\star$	0.497	**0.039** ±.014	0.543 ±.017	0.247 ±.002	**0.054** ±.016	0.564 ±.021	0.248 ±.004	0.070 ±.016	0.614 ±.021	0.239 ±.003

Table 39: **Position sampling rules on LiveCodeBench.**

		$t{=}0$			$t \approx T/2$			$t{=}T$		
position	pass@1	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓	ECE ↓	AUROC ↑	Brier ↓
$u{=}0$	0.618	0.106 ±.014	0.804 ±.016	0.186 ±.008	0.122 ±.015	0.794 ±.015	0.192 ±.009	0.175 ±.016	0.816 ±.014	0.203 ±.010
$u{=}\frac{1}{2}$	0.626	0.093 ±.013	0.799 ±.017	0.184 ±.008	0.084 ±.011	0.812 ±.016	0.173 ±.006	0.091 ±.011	0.837 ±.015	0.165 ±.007
$u{=}1$	0.618	0.097 ±.012	0.807 ±.016	0.183 ±.006	0.154 ±.014	0.796 ±.015	0.206 ±.006	0.093 ±.011	0.839 ±.014	0.164 ±.006
grid_2	0.620	0.120 ±.011	**0.820** ±.015	0.186 ±.005	0.141 ±.013	0.813 ±.015	0.192 ±.005	0.145 ±.012	**0.841** ±.014	0.174 ±.005
grid_3	0.626	0.079 ±.012	0.808 ±.016	0.176 ±.006	0.071 ±.010	0.810 ±.016	0.174 ±.006	0.088 ±.011	0.833 ±.015	0.166 ±.007
grid_9	0.624	0.084 ±.012	0.820 ±.016	**0.172** ±.007	**0.070** ±.010	**0.820** ±.016	**0.169** ±.007	0.086 ±.011	0.840 ±.015	**0.162** ±.007
$U[0,1]^\star$	0.623	**0.073** ±.012	0.780 ±.016	0.188 ±.007	0.075 ±.011	0.799 ±.016	0.177 ±.006	**0.076** ±.010	0.824 ±.015	0.170 ±.007

I ROUTING: DETAILED RESULTS AND COST ASSUMPTIONS

Evaluation setup. The small arm is Qwen3-8B PACE-u (uniform sampling, step 100), and the large arm is Qwen3-32B RLVR-GRPO (DeepMath, step 100). The large arm is an answer-generation checkpoint, distinct from the 32B confidence-trained arm in Figure 4, and is chosen to reason better than the small arm. Each arm has eight answers per question. At $t = 0$, confidence draws and small-arm answer outcomes are permuted independently; at $t = T$, each confidence stays paired with its own answer. The large arm is permuted separately. Datasets are weighted equally, and pass@1 is evaluated on the selected arm's answer. Operating points are selected retrospectively on the evaluation sets rather than on a separate validation set. We therefore report descriptive point estimates. The question-clustered bootstrap intervals (1000 replicates) cover individual curves but not paired operating-point comparisons, so we claim no significance and do not dismiss negative results on small datasets as noise.

Cost model. We assume that inference cost scales with model size ($4\times$ from 8B to 32B) and that output tokens cost $4\times$ as much as input tokens, a ratio typical of commercial API pricing. With answer prompt length n_{in} and generated lengths n_s, n_l, the answer costs are $S = n_{\text{in}} + 4n_s$ and $L = 4n_{\text{in}} + 16n_l$ for the SLM and LLM, respectively. We further assume KV-cache reuse at both positions, so no prefix is prefilled twice. The readout is charged identically at both positions, with δ instruction tokens as input and eight digits as output, giving $P = \delta + 32$. By Equation 3, at $t = 0$ a kept question costs $S + P$ and a deferred one $n_{\text{in}} + P + L$, whereas at $t = T$ every question pays $S + P$ and deferral adds L. Without context reuse (e.g., when the late readout is a separate request on a different worker), the answer must be re-prefilled; this changes only the cost accounting, not the predictions. The proxy approximates, but is neither, measured latency nor an actual API price, and we do not claim that a serving system achieves the assumed reuse. Cascade costs are normalized by each dataset's $\bar{L}_j$, and the ratio $\bar{L}_j/\bar{S}_j$ averages 4.082 (Table 40).

Table 40: **Answer-generation endpoints for routing.** Direct execution without a confidence probe; costs use the weighted-token proxy. The last row gives total questions and macro averages.

Dataset	Questions	Small pass@1	Large pass@1	Large/small cost
MATH-500	500	0.854	0.926	4.046
GSM8K	1319	0.917	0.941	3.753
GPQA-Diamond	198	0.497	0.609	4.421
CommonsenseQA	1221	0.806	0.849	4.284
OlympiadBench	674	0.554	0.678	4.233
LiveCodeBench	400	0.623	0.710	3.648
DeepMath-500	500	0.640	0.840	4.189
Macro / total	4812	0.699	0.793	4.082

Comparators. *Random routing* defers a uniformly random subset of fixed fraction r and pays the same probe cost as the pipeline it is compared against. At equal r, it is *cheaper* than confidence routing, which preferentially defers hard questions. Hard questions draw longer answers: a question's pass rate correlates -0.49 and -0.73 with the large arm's answer length on DeepMath-500 and MATH-500, respectively. We therefore compare at equal cost rather than equal deferral rate. *Base-ProbeMLP* is evaluated on the PACE-u answer pool. ConfTuner's digit is read from teacher-forced logits and charged only the four input tokens of its `"Confidence: "` prefix.

Threshold sweeps. A deployed cascade fixes a threshold α and keeps the small answer whenever $c \geq \alpha$. Tables 41 and 42 sweep α on every test set at both positions for PACE. The grid ends serve as a sanity check. At $\alpha = 0$, no question is deferred, which recovers the SLM's pass@1. At $\alpha = 1.1$, which a readout bounded by 1 cannot reach, every question is deferred, which recovers the LLM's pass@1. The $t = T$ curves start at the same accuracy as the $t = 0$ curves but end above $1.2\times$ the LLM-only cost, since a late deferral discards a small answer that has already been paid for. All entries average eight seeds.

Table 41: **Cascade over the confidence threshold α: mathematics.** DeepMath-500 (in-domain) and three OOD mathematics sets. For each test set and position, rows give pass@1, cost relative to LLM-only, and deferral rate; the SLM answer is kept if $c \geq \alpha$. The $\alpha = 0$ and $\alpha = 1.1$ columns reproduce the endpoints in Table 40. Eight seeds; context is reused at $t = T$.

			confidence threshold α											
dataset	position	quantity	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1
DeepMath-500	t=0	pass@1	0.641	0.641	0.641	0.641	0.643	0.647	0.760	0.820	0.822	0.825	0.832	0.840
		cost	0.24	0.24	0.24	0.24	0.25	0.28	0.59	0.84	0.86	0.88	0.95	1.01
		defer	0.00	0.00	0.00	0.00	0.01	0.03	0.36	0.69	0.72	0.74	0.89	1.00
DeepMath-500	t=T	pass@1	0.641	0.641	0.641	0.641	0.646	0.648	0.756	0.824	0.826	0.826	0.836	0.840
		cost	0.24	0.24	0.24	0.24	0.27	0.29	0.67	1.00	1.02	1.02	1.15	1.24
		defer	0.00	0.00	0.00	0.00	0.01	0.02	0.32	0.64	0.66	0.66	0.83	1.00
MATH-500	t=0	pass@1	0.854	0.854	0.854	0.854	0.856	0.861	0.884	0.910	0.913	0.915	0.925	0.927
		cost	0.25	0.25	0.25	0.25	0.27	0.30	0.48	0.70	0.75	0.77	0.96	1.01
		defer	0.00	0.00	0.00	0.00	0.01	0.02	0.12	0.33	0.39	0.43	0.84	1.00
MATH-500	t=T	pass@1	0.854	0.854	0.854	0.854	0.860	0.865	0.885	0.907	0.908	0.908	0.917	0.927
		cost	0.25	0.25	0.25	0.25	0.31	0.34	0.50	0.69	0.72	0.72	0.98	1.25
		defer	0.00	0.00	0.00	0.00	0.02	0.03	0.09	0.20	0.22	0.23	0.50	1.00
GSM8K	t=0	pass@1	0.917	0.917	0.917	0.917	0.917	0.917	0.919	0.925	0.926	0.927	0.940	0.941
		cost	0.28	0.28	0.28	0.28	0.28	0.28	0.29	0.37	0.39	0.41	0.93	1.03
		defer	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.08	0.10	0.13	0.82	1.00
GSM8K	t=T	pass@1	0.917	0.917	0.917	0.917	0.917	0.917	0.919	0.928	0.929	0.929	0.939	0.941
		cost	0.28	0.28	0.28	0.28	0.28	0.28	0.29	0.35	0.36	0.36	0.81	1.28
		defer	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.05	0.05	0.05	0.44	1.00
OlympiadBench	t=0	pass@1	0.554	0.554	0.554	0.554	0.561	0.576	0.631	0.657	0.666	0.668	0.678	0.678
		cost	0.24	0.24	0.24	0.24	0.30	0.40	0.74	0.90	0.94	0.95	1.00	1.01
		defer	0.00	0.00	0.00	0.00	0.05	0.14	0.48	0.70	0.77	0.81	0.98	1.00
OlympiadBench	t=T	pass@1	0.554	0.554	0.554	0.554	0.566	0.570	0.624	0.654	0.657	0.657	0.673	0.678
		cost	0.24	0.24	0.24	0.24	0.33	0.35	0.78	0.99	1.04	1.04	1.17	1.24
		defer	0.00	0.00	0.00	0.00	0.05	0.07	0.35	0.54	0.59	0.60	0.82	1.00

Table 42: **Cascade over the confidence threshold α: non-mathematics.** CommonsenseQA, GPQA-Diamond, and LiveCodeBench (zero-shot transfer). Conventions as in Table 41.

			confidence threshold α											
dataset	position	quantity	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1
CommonsenseQA	t=0	pass@1	0.806	0.806	0.806	0.806	0.806	0.806	0.818	0.846	0.846	0.847	0.849	0.849
		cost	0.24	0.24	0.24	0.24	0.24	0.24	0.38	0.93	0.93	0.98	1.02	1.02
		defer	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.87	0.88	0.94	1.00	1.00
CommonsenseQA	t=T	pass@1	0.806	0.806	0.806	0.806	0.808	0.808	0.833	0.849	0.849	0.849	0.849	0.849
		cost	0.24	0.24	0.24	0.24	0.24	0.24	0.38	1.13	1.14	1.20	1.24	1.24
		defer	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.88	0.89	0.95	1.00	1.00
GPQA-Diamond	t=0	pass@1	0.497	0.497	0.497	0.497	0.496	0.489	0.605	0.609	0.609	0.609	0.609	0.609
		cost	0.23	0.23	0.23	0.23	0.24	0.27	0.96	1.01	1.01	1.01	1.01	1.01
		defer	0.00	0.00	0.00	0.00	0.01	0.04	0.89	1.00	1.00	1.00	1.00	1.00
GPQA-Diamond	t=T	pass@1	0.497	0.497	0.497	0.497	0.508	0.507	0.594	0.609	0.609	0.609	0.609	0.609
		cost	0.23	0.23	0.23	0.23	0.31	0.33	1.08	1.23	1.23	1.23	1.23	1.23
		defer	0.00	0.00	0.00	0.00	0.05	0.06	0.78	1.00	1.00	1.00	1.00	1.00
LiveCodeBench	t=0	pass@1	0.623	0.623	0.623	0.623	0.623	0.640	0.654	0.698	0.702	0.705	0.708	0.710
		cost	0.28	0.28	0.28	0.28	0.28	0.45	0.59	0.90	0.92	0.94	1.03	1.05
		defer	0.00	0.00	0.00	0.00	0.00	0.15	0.29	0.68	0.70	0.74	0.94	1.00
LiveCodeBench	t=T	pass@1	0.623	0.623	0.623	0.623	0.623	0.645	0.662	0.692	0.702	0.703	0.710	0.710
		cost	0.28	0.28	0.28	0.28	0.31	0.57	0.73	1.01	1.09	1.10	1.27	1.28
		defer	0.00	0.00	0.00	0.00	0.02	0.19	0.31	0.58	0.68	0.69	0.97	1.00