OPSRD: ON-POLICY SELF-ROLE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Role prompting elicits specialized behavior from large language models through an expert identity, offering a lightweight way to guide reasoning on demanding tasks. However, evaluating or distilling complete role-prompted answers can miss useful next-token preferences when the sampled solution remains incorrect. Transferring these preferences also requires an objective that reaches alternatives the student rarely predicts. We introduce OPSRD, which uses a fixed expert role as privileged teaching context for on-policy self-distillation without reference solutions. A role-free student generates a trajectory, and a frozen instance of the same base model supplies role-conditioned distributions on its exact prefixes, exposing alternatives beyond the sampled continuation. Teacher-weighted forward KL targets alternatives the student underestimates, with clipping to limit individual vocabulary contributions. Supervision is restricted to the highest-entropy half of student positions, concentrating learning where predictions are uncertain. Experiments on three competition-math benchmarks with Qwen3-1.7B, 4B, and 8B show improvements over the base models without role prompts at inference. Forward KL achieves the highest macro-averaged accuracy among the three evaluated divergences at every scale.

1 INTRODUCTION

Large language models serve as general-purpose assistants for reasoning, explanation, and problem solving. Their usefulness depends on how instructions elicit the capabilities needed for a particular task (Liu et al., 2023). Role prompting makes this dependence explicit by asking a model to adopt an expert identity (Xu et al., 2025a). A mathematical expert role can request careful derivations and verification, while a critic role can encourage checking intermediate conclusions. Such prompts offer a practical way to guide model behavior without changing its parameters.

However, existing uses of role prompting often assess expertise through complete generated responses. Reported reasoning gains and failures (Kong et al., 2024; Kim et al., 2024) therefore mix the role's local influence with the effects of sampling an entire solution. A useful next-token preference can remain unsampled or be followed by a later mistake. Distilling expert-prompted responses (Xu et al., 2025a) likewise transfers only the sampled continuations. On-policy self-distillation accesses richer token distributions, but solution-conditioned OPSD derives its teaching advantage from a reference solution for each problem (Zhao et al., 2026). The open question is whether an expert role can supervise the student's reasoning even when its complete answers offer no measured advantage.

Our starting observation is a negative result. On three competition-math benchmarks, adding an expert olympiad mathematician role to Qwen3-1.7B changes macro accuracy from 35.74 to 34.54, providing no measured improvement over the ordinary prompt. A neutral system instruction reaches the same score. The expert label alone is insufficient to improve sampled solutions in this setting. What remains unobserved is the role-conditioned distribution at each intermediate choice, before later decisions determine the final answer.

We hypothesize that a role-conditioned model can teach useful next-token preferences even when its complete solutions offer no accuracy gain. Under this hypothesis, the role's value lies in shifting probability toward helpful continuations at uncertain student prefixes. Holding these prefixes fixed lets the teacher evaluate the choices the student actually faces. On-policy distillation provides this alignment (Agarwal et al., 2024; Gu et al., 2024), allowing us to test a reusable role as privileged teaching context without per-problem reference solutions.

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

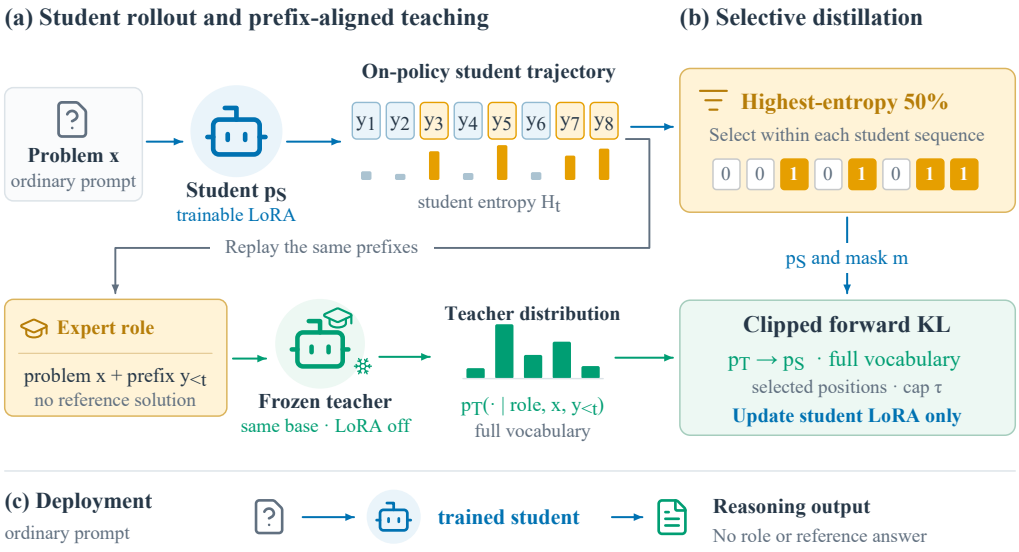

Figure 1: OPSRD overview. The student generates a role-free trajectory; the frozen expert-role teacher evaluates the same prefixes. The highest-entropy half of student positions receive clipped, full-vocabulary forward KL. Only student LoRA parameters are updated, and deployment uses the ordinary prompt at test time.

We develop OPSRD to test this hypothesis. The student generates trajectories under its ordinary prompt. A frozen instance of the same base model receives the expert role and scores the exact student prefixes, supplying a distribution over possible next tokens. We retain the highest-entropy half of student positions to focus supervision on uncertain choices. Only a low-rank student adapter is updated (Hu et al., 2022); the teacher stays fixed, no reference answer enters training, and the trained student uses its ordinary prompt at inference.

For this feedback to help, the objective must reach alternatives the student currently underestimates. A reverse-KL update weakens as the student's probability of an alternative approaches zero, even when the teacher supports it. Forward KL weights the target by teacher probability and directly corrects this underestimation before clipping. We therefore use forward KL, with a cap on individual vocabulary contributions to limit large discrepancies from an imperfect teacher. We compare forward KL, reverse KL, and Jensen–Shannon divergence under the same role and entropy mask.

Experiments cover Qwen3-1.7B, 4B, and 8B on three competition-math benchmarks. Forward KL achieves the highest macro accuracy among the evaluated divergences at all three scales (Figure 3). At 1.7B, OPSRD improves Base by 6.30 percentage points in the primary run, and independent training repeats retain the gain. Placement controls favor keeping the role in the teacher. The study contributes an empirical contrast between direct prompting and role-based supervision, an answer-free on-policy method, and an analysis of how divergence and role placement govern this transfer.

2 Related Work

2.1 On-Policy Distillation

On-policy distillation learns from student-generated trajectories, with the teacher evaluating the same prefixes (Agarwal et al., 2024; Gu et al., 2024). Sequence-level distillation instead trains on teacher-generated outputs (Kim & Rush, 2016). Using student prefixes addresses the train–test distribution mismatch central to imitation learning (Ross et al., 2011). The teacher supplies distributional supervision (Hinton et al., 2015), while the student determines where it applies. Generalized knowledge distillation permits different divergences, and MiniLLM uses reverse KL. Entropy-aware OPD adds forward KL when teacher entropy is high (Jin et al., 2026), linking the objective to teacher uncertainty.

The quality of feedback also depends on the prefixes the student visits. Speculative knowledge distillation lets the teacher replace poorly ranked student proposals, bringing the sampled trajectory closer to teacher-supported continuations (Xu et al., 2025b). Other methods adjust the learning signal at a fixed prefix. Asymmetric OPD combines positive reinforcement with local divergence minimization (Jia et al., 2026), while empirical analyses identify unreliable teacher guidance and imbalanced token supervision as failure modes (Fu et al., 2026). CROP selects positions using counterfactual task relevance (Li et al., 2026), and spurious-signal filtering uses input dependence together with divergence to remove misleading updates (Jiang et al., 2026). This work establishes that visiting student states alone does not make every teacher preference equally useful.

Self-distillation changes where the teacher's advantage comes from. Earlier approaches learn from model-generated reasoning, instructions, or filtered outputs (Zelikman et al., 2022; Wang et al., 2023; Gulcehre et al., 2023). OPSD uses one model under two contexts: its teacher sees a privileged solution and evaluates the question-only student's rollouts (Zhao et al., 2026). This supplies token-level feedback without a separately pretrained teacher. OPSRD builds on this contextual asymmetry, replacing the per-problem solution with a reusable expert role and asking how to transfer that weaker source of supervision.

2.2 Role-Conditioned Supervision

Prompts can expose behavior that a model does not reliably express under an ordinary instruction. ExpertPrompting constructs an expert identity for each instruction, uses it to generate responses, and trains ExpertLLaMA on the resulting data (Xu et al., 2025a). Role-play prompting also improves selected reasoning tasks (Kong et al., 2024), and RoleLLM studies how to elicit and train broader role-playing abilities (Wang et al., 2024). Controlled experiments find that persona prompts can impair reasoning as well as help it (Kim et al., 2024). The question for distillation is consequently how a role's influence should be measured and transferred when its complete sampled answers offer an uncertain advantage.

Context distillation provides a bridge from conditional behavior to learned behavior by internalizing gains elicited by instructions and scratchpads (Snell et al., 2022). In-context learning distillation transfers the ability to learn from demonstrations (Huang et al., 2022), and self-distillation can reduce distribution gaps during fine-tuning (Yang et al., 2024). These studies show several ways that a model's context can create a teaching signal. For a role teacher, the relevant signal may lie in its preferences among possible next tokens on an existing student prefix. Capturing these preferences gives access to guidance that a comparison of complete role-prompted solutions can miss.

In OPSRD, a single role supplies context for a teacher that shares the student's base model. The teacher scores role-free student prefixes, and entropy selection concentrates forward-KL supervision on uncertain positions. This makes the role's distributional feedback the object of distillation while keeping student generation under the ordinary prompt.

3 Method

3.1 Setting

Let x be a math problem and let c_S denote the role-free training context. The student p_θ contains low-rank trainable parameters on top of a pretrained model p_{θ_0}. For each problem, the student samples a reasoning trajectory

$$y_{1:T} \sim p_\theta(\cdot \mid c_S, x), \tag{1}$$

without a role or reference solution. Sampled tokens are detached. The frozen teacher uses the same base weights with the student adapter disabled and an expert-role context c_T. Training uses a non-thinking student template and a thinking teacher template; evaluation enables thinking. Training also includes a `Problem:` prefix that evaluation omits. Appendix A.2 specifies the complete prompts.

At position t, both models receive the exact student prefix $y_{<t}$. Let z_S^t and z_T^t be their logits. The distributions used in entropy selection and distillation apply temperature $\eta = 1.1$,

$$p_S^t(v) = \mathrm{softmax}(z_S^t/\eta)_v, \quad z_S^t = z_\theta(c_S, x, y_{<t}), \tag{2}$$

$$p_T^t(v) = \mathrm{softmax}(z_T^t/\eta)_v, \quad z_T^t = z_{\theta_0}(c_T, x, y_{<t}), \tag{3}$$

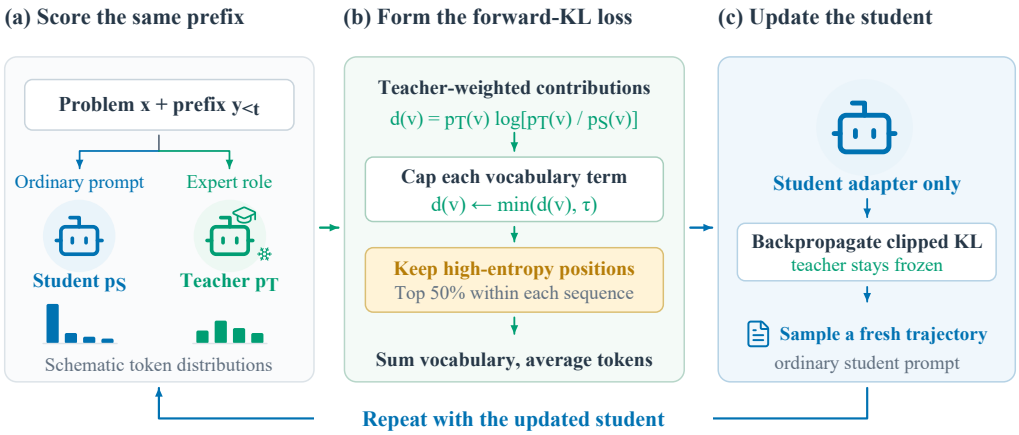

Figure 2: One OPSRD update. Student and teacher score the same prefix; clipped vocabulary contributions are summed and averaged over the selected high-entropy positions. Only the student adapter is updated before the next rollout. The small distributions are schematic.

for every vocabulary item v. The teacher evaluates exactly the tokens sampled by the student. Sampling additionally applies top-$p = 0.95$ and top-$k = 20$; the loss distributions retain the full vocabulary. Only student logits receive gradients; sampling and teacher probabilities remain detached.

3.2 Role-privileged teaching context

The teacher's reusable role requests rigorous olympiad reasoning, verification, and a checked final answer, without task-specific hints. Holding $(x, y_{<t})$ fixed aligns the compared positions. Their distributional difference includes the role, thinking template, and adapter state. The neutral-teacher control shares the template and chat placement but uses a generic instruction. Figure 1 shows the overall method; Figure 2 follows one training update.

3.3 Student-entropy selection

The role does not need equal authority everywhere. Many tokens are nearly fixed by syntax or an earlier decision, so we compute student entropy at each valid position,

$$H_t = -\sum_{v \in \mathcal{V}} p_S^t(v) \log p_S^t(v), \tag{4}$$

and retain the $k = \max(1, \lceil \rho T \rceil)$ largest values within each sequence, with default $\rho = 0.5$. This assigns every sequence its own selection quota $m_t \in \{0, 1\}$. Longer sequences still contribute more selected positions to the token-averaged objective. Selection uses detached student logits, so the ranking receives no gradient; Appendix B illustrates the selection rule.

3.4 Clipped forward-KL objective

For a selected position, the unmodified forward KL is

$$D_t = \sum_{v \in \mathcal{V}} p_T^t(v) \log \frac{p_T^t(v)}{p_S^t(v)}. \tag{5}$$

We retain the full vocabulary and cap each signed contribution at $\tau = 0.05$ for the primary 1.7B study. The 8B runs use $\tau = 0.06$. Writing $d_{t,v} = p_T^t(v) \log[p_T^t(v)/p_S^t(v)]$, the batch objective is

$$\mathcal{L}(\theta) = \frac{\sum_{i \in \mathcal{B}} \sum_{t=1}^{T_i} m_{i,t} \sum_{v \in \mathcal{V}} \min\{d_{i,t,v}, \tau\}}{\max(1, \sum_{i \in \mathcal{B}} \sum_{t=1}^{T_i} m_{i,t})}. \tag{6}$$

The clip limits a few large teacher preferences without truncating the vocabulary or changing the selected positions. The unmodified KL is nonnegative, but its vocabulary contributions have both

signs. Capping only positive contributions can make the optimized sum negative. We therefore log the unclipped divergence separately from the training loss. Only LoRA parameters receive gradients; the teacher remains fixed.

3.5 WHY FORWARD KL FITS A ROLE TEACHER

The teacher's contribution is a distribution over alternatives at an existing student prefix. Forward KL gives each alternative weight proportional to its teacher probability. To see the resulting update, consider an unclipped selected position. Its derivative with respect to student logit $z_{S,j}$ is

$$\frac{\partial D_{\mathrm{F}}}{\partial z_{S,j}} = \frac{p_S(j) - p_T(j)}{\eta}. \tag{7}$$

When the teacher assigns appreciable probability to a continuation the student rarely predicts, this gradient directly increases its logit. For reverse KL, $D_{\mathrm{R}} = \sum_v p_S(v) \log[p_S(v)/p_T(v)]$, the derivative instead becomes

$$\frac{\partial D_{\mathrm{R}}}{\partial z_{S,j}} = \frac{p_S(j)}{\eta} \left(\log \frac{p_S(j)}{p_T(j)} - D_{\mathrm{R}} \right). \tag{8}$$

For fixed positive teacher probabilities, this update vanishes as $p_S(j)$ approaches zero. The distinction motivates forward KL for transferring preferences that the role elicits but the student underuses. It also complements entropy selection, which directs learning toward prefixes with several plausible continuations. The contribution cap modifies these derivatives when active; Appendix B gives the exact clipped forward gradient. The following experiments evaluate this choice under the implemented loss and training procedure.

3.6 WHY THE ASYMMETRY MATTERS

Teacher-only conditioning keeps the student persona-free during both rollout and evaluation. Shared or student-only conditioning changes the prefixes on which the teacher supplies supervision. Appendix B contrasts the three placements. Their comparison tests whether role-conditioned student sampling helps the final model under either evaluation prompt. Thinking-mode and formatting choices remain fixed across the placement ablation, as specified in Appendix A.2.

4 EXPERIMENTS

Our experiments test whether an expert role can provide useful supervision when it offers little benefit as a direct prompt. We first compare role distillation with prompting and answer-conditioned distillation, then examine the choices that govern the transfer. Repeated training and evaluation assess the stability of the gains, and a problem-level analysis shows how those gains change the student's answers.

4.1 EXPERIMENTAL SETUP

Models and data. We use Qwen3-1.7B, 4B, and 8B (Yang et al., 2025), with a frozen instance of each base model serving as its own teacher. Training uses the 29,434-example mathematical split derived from OpenThoughts (Guha et al., 2025). All methods train a rank-64 LoRA adapter for 100 optimizer steps with learning rate 5×10^{-6} and effective batch size 36. The student samples one rollout per problem, capped at 1,024 tokens. We use 1.7B for the detailed ablations and independent training repeats; the larger models test whether the recipe remains useful as base accuracy increases. Appendix A gives the remaining hyperparameters and exact prompts.

Evaluation. We evaluate on AIME 2024, AIME 2025, and HMMT February 2025, each containing 30 problems. Every problem receives 12 samples at temperature 1.0 and top-p 0.95, with a 38,912-token generation cap. We report average sample accuracy for each task and its unweighted macro average, denoted $\mathrm{Avg@12}$. All comparisons use the final step-100 checkpoint and the ordinary evaluation prompt unless a prompt change is the intervention being tested. Reported 95% intervals use paired problem-level bootstrap resampling, with the training repeats additionally accounting for shared test problems. Appendix D defines these estimators.

Table 1: Primary Qwen3-1.7B comparison. Avg@12 in percent; gains in percentage points.

Method	AIME24	AIME25	HMMT25	Macro	Δ Base
Base	46.67	36.11	24.44	35.74	0.00
Answer OPSD	51.39	39.17	25.56	38.70	+2.96
Base with expert role	46.94	37.22	19.44	34.54	−1.20
Base with neutral prompt	45.83	35.83	21.94	34.54	−1.20
Role full	55.56	39.72	25.00	40.09	+4.35
OPSRD	**57.78**	**41.67**	**26.67**	**42.04**	+6.30
Neutral full	50.28	40.00	24.44	38.24	+2.50

Table 2: Macro Avg@12. The 4B summary averages four evaluation repeats.

Model	Evaluation	Base	Answer OPSD	OPSRD	Δ Base	Δ Answer
Qwen3-1.7B	Primary	35.74	38.70	42.04	+6.30	+3.33
Qwen3-4B	Mean of 4	60.86	61.50	62.52	+1.67	+1.02
Qwen3-8B	Primary	62.78	64.26	65.83	+3.06	+1.57

Baselines. Base is the untrained model. Direct expert and neutral prompting test whether an extra system message improves inference by itself. Answer OPSD supplies a reference solution to the teacher, following Zhao et al. (2026), and provides the main training baseline. Role full and Neutral full distill an expert or generic teacher instruction at every valid position. OPSRD combines the expert teacher with the highest-entropy half of student positions. This set of comparisons connects the practical benefit of role distillation to the teacher context, role placement, and training objective.

4.2 Main Results

From prompting to supervision. The central result is a change in how the same role becomes useful. Direct expert prompting gives 34.54 macro accuracy, close to the 35.74 Base score, and the paired difference has an interval of $[-4.35, 1.94]$. Using that role for the teacher raises accuracy to 40.09 with full distillation and 42.04 with OPSRD (Table 1). The latter improves Base by $6.30\,\mathrm{pp}$, with interval $[3.24, 9.54]$, and Answer OPSD by $3.33\,\mathrm{pp}$, with interval $[0.83, 6.02]$. The teacher receives the same short instruction for every training example, making the supervision reusable across problems without their reference solutions.

The benchmark results clarify this change. Direct prompting lowers HMMT accuracy while leaving the two AIME scores near Base. After distillation, all three task point estimates improve, with the largest gain on AIME 2024. The role's failure to raise aggregate prompting accuracy therefore does not preclude useful feedback on the student's prefixes. The placement and divergence ablations below examine how that feedback reaches the trained model.

Results across model sizes. Table 2 extends the comparison to stronger base models. OPSRD improves macro accuracy at all three scales, with gains of 6.30, 1.67, and 3.06 points over Base. The 4B entry averages four evaluations of each fixed checkpoint; the other scales use the primary evaluation. The gains are smaller as baseline accuracy rises, although their magnitude does not change monotonically with model size. Across these summaries, role distillation also exceeds the answer-conditioned comparator while requiring less privileged training context.

The per-task comparisons reveal different preferences for teacher context. Answer OPSD has the strongest 4B primary evaluation, and it slightly exceeds OPSRD on 8B AIME 2025. Averaging the repeated 4B evaluations changes the overall ordering. Role distillation remains useful across model sizes, while the strength of its advantage depends on the benchmark and evaluation sample. Appendix C provides the complete benchmark scores, and Section 4.4 assesses variation across all four repeated evaluations.

4.3 Ablation Studies

Teacher context. We first ask how much of the benefit requires an expert instruction. Neutral full improves Base by 2.50 points, showing that a generic teacher context already supplies useful

Table 3: Role placement and deployment prompt on Qwen3-1.7B.

Training role	Test role	Macro
Teacher only	No role	42.04
Teacher only	Expert role	39.44
Shared role	No role	38.98
Shared role	Expert role	36.94
Student only	No role	38.24
Student only	Expert role	36.30

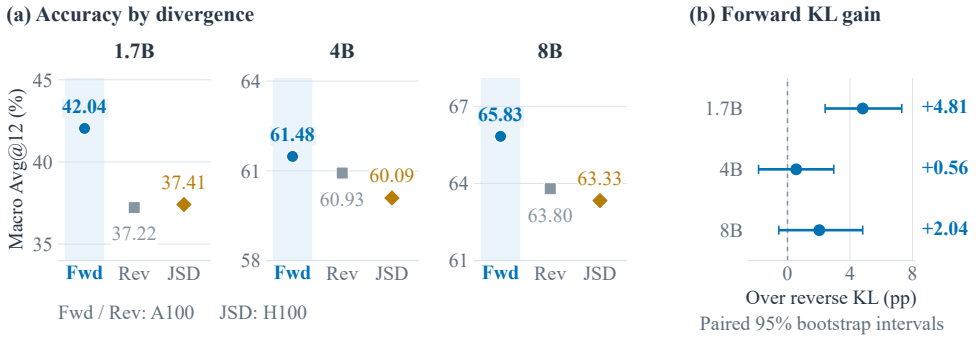

Figure 3: Divergence ablation. (a) Macro Avg@12 from the primary evaluation at each scale. (b) Forward-minus-reverse gains with paired 95% problem-bootstrap intervals. Each condition uses one training run; JSD uses H100 and the KL runs use A100.

self-distillation feedback. Role full adds 1.85 points over this control, with interval $[-0.28, 3.98]$. The combined OPSRD recipe is 3.80 points above Neutral full, with interval $[1.20, 6.39]$. These results motivate the expert context, while the missing neutral-teacher 50% control leaves its contribution at the selected positions unresolved. The remaining ablations hold the expert instruction fixed and test how to transfer its feedback.

Role placement. The role can change either the student's trajectory or the teacher's view of it. To distinguish these effects, we place the role in the teacher, the student, or both, holding the model size, selection fraction, checkpoint, and evaluation protocol fixed. Each trained model is then evaluated with and without the role. This comparison tests the division of labor in OPSRD, where the student chooses the trajectory and the teacher supplies the additional perspective.

Teacher-only conditioning achieves the highest score under the ordinary evaluation prompt (Table 3). Giving the role to both models lowers accuracy by 3.06 points, and giving it only to the student lowers accuracy by 3.80 points. Both paired intervals lie below zero. Adding the role again at evaluation lowers each checkpoint's point estimate, including the teacher-only model. The benefit of training with role feedback therefore persists when the student returns to its ordinary prompt. Conditioning the rollout generator on the role changes the training context in a way that these placement results do not favor.

Distillation objective. The next comparison tests how the student should absorb the role teacher's distribution. We replace forward KL with reverse KL or JSD while retaining the expert teacher and 50% entropy mask. Forward KL gives the highest macro accuracy at all three scales (Figure 3). The clearest separation is at 1.7B, where it exceeds reverse KL by 4.81 points with interval $[2.41, 7.31]$. This result agrees with the motivation in Section 3: teacher-supported alternatives receive a direct learning signal even when the student assigns them little probability.

The larger-model gaps are less conclusive. Forward-minus-reverse intervals cross zero at 4B and 8B, and reverse KL is stronger on 4B AIME 2024. We therefore use forward KL as the default supported most clearly by the primary scale. All divergence variants have one training run; JSD also differs in hardware and is supported by saved aggregate scores. Appendix C.2 gives the objective definitions,

Table 4: Mean macro Avg@12 and sample standard deviation over three training runs at 1.7B and four evaluations of fixed checkpoints at 4B. The fixed 1.7B Base score is 35.74.

Model	Repeat	Method	Macro
1.7B	Training	Role full	39.48 ± 1.32
1.7B	Training	OPSRD	40.25 ± 1.62
4B	Evaluation	Base	60.86 ± 0.67
4B	Evaluation	Answer OPSD	61.50 ± 0.77
4B	Evaluation	OPSRD	62.52 ± 0.78

full task results, and comparison settings. The 4B result in this figure uses the primary evaluation, whereas Table 2 summarizes the repeated evaluations.

Supervision density. Finally, we test whether the gain survives concentrating supervision on uncertain positions. Selecting half the valid positions raises the primary score by 1.94 points over Role full, with interval $[-0.56, 4.44]$. Across three independent training runs, full and selective distillation remain close, with a mean difference of 0.77 points and interval $[-1.33, 2.90]$. The supported finding is that half the supervised positions retain the benefit of role distillation. Both variants still compute the student and teacher forward passes.

The entropy measurements help explain what this restriction does. Selected positions have mean student entropy 0.874, compared with 0.023 at omitted positions. The objective therefore focuses on predictions with substantial uncertainty while leaving nearly settled predictions outside the loss. A smaller sampling-budget pilot also explored 25% and 75% selection, but did not establish a better fraction. Appendix C.4 reports this sensitivity study separately from the full-budget comparison.

4.4 Robustness

We assess whether the gains survive changes in optimization and evaluation sampling. At 1.7B, three independent training runs give OPSRD a mean accuracy of 40.25 ± 1.62, improving the fixed Base evaluation by 4.51 points with interval $[1.54, 7.69]$. Every run improves Base. Role full also retains its gain, consistent with the supervision-density ablation. The repeat results support the value of role distillation more strongly than a further advantage from selecting half the positions.

At 4B, four evaluations of each fixed checkpoint yield a mean gain of $1.67\,\mathrm{pp}$ over Base, with sample standard deviation $0.39\,\mathrm{pp}$ for the paired gains. OPSRD exceeds Base in every evaluation. This agreement is useful because the primary comparison with Answer OPSD changes when evaluation sampling is repeated. Table 4 summarizes all repeats. These evaluations span A100 and H100, so their dispersion includes environment variation as well as decoding randomness; independent-training variation is measured by the separate 1.7B study.

4.5 Further Analysis

How the answers improve. An increase in average accuracy can reflect more reliable solutions to familiar problems or successful attempts on previously unsolved ones. We distinguish these outcomes by pairing Base and OPSRD on every test problem. Figure 4 shows that OPSRD increases the 12-sample success rate on 38 problems and lowers it on 12, with the rest tied. The improvements are most widespread on AIME 2024. This pattern accounts for the stronger AIME gains in the aggregate comparison and the repeated-training summary.

The change in observed coverage is much smaller. Seven problems with no correct Base sample receive a correct OPSRD sample, while six change in the opposite direction. Most improved problems already have at least one successful Base trajectory. The main benefit is a better chance of reaching a correct answer on problems the student sometimes solves already. This is consistent with using a role teacher's local preferences to strengthen reasoning that the ordinary student can already express under its usual prompt.

This distinction connects the evaluation to the training design. OPSRD learns on student-generated prefixes, so useful feedback must act within trajectories the student actually visits. The problem-level result is compatible with that local form of guidance. It also explains why an improvement in

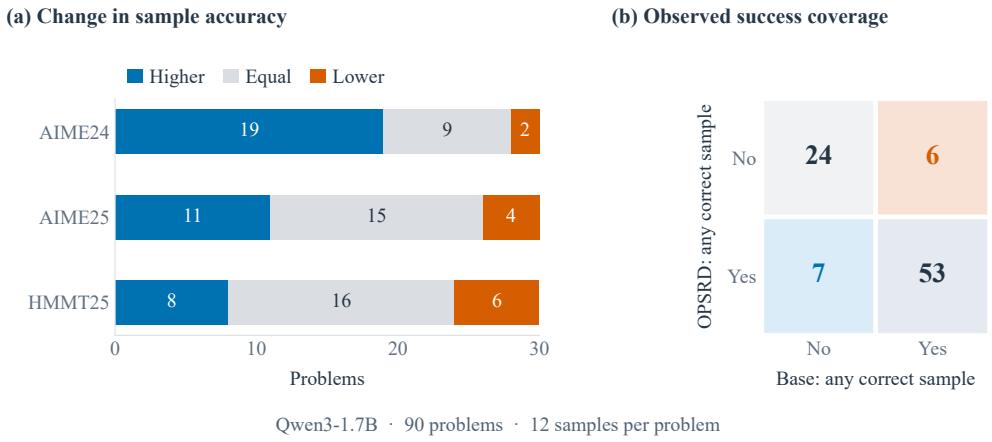

Figure 4: Problem-level changes at 1.7B. Left: problems with higher, equal, or lower sample accuracy than Base. Right: observed success coverage within 12 samples. The main gain is more reliable success on partially solved problems.

Avg@12 can coexist with little change in the number of problems solved at least once. Coverage here is measured at a finite sampling budget; broader claims about newly acquired problem-solving abilities would require a different evaluation.

Answer format and length. We also check whether the primary improvement can be explained by output failures. Format compliance exceeds 99% across the primary conditions, and both Base and OPSRD rarely reach the generation cap. The additional 68 correct generations exceed the differences in malformed and near-cap outputs. These discrepancies are too small individually to explain the gain, supporting the interpretation that the observed change concerns answer accuracy under the shared long-form evaluation budget.

5 Discussion and Conclusion

The experiments connect the role's placement to the distribution it teaches. Direct prompting changes the student's trajectory from its first token. Teacher-only distillation instead evaluates the trajectory the student already produced, and the placement controls favor keeping these two functions separate. Forward KL weights the update by the teacher's probabilities, allowing teacher-supported alternatives to receive a learning signal even when the student rarely predicts them. Equation 7 describes this behavior before clipping, while Appendix B gives the gradient of the implemented objective. Tracking those alternatives through training would test how this local signal produces the observed answer-level gains. Both full and selective distillation improve Base across training seeds. Their paired interval crosses zero, so the selection result supports retaining the benefit at half the supervised positions. The implementation still requires both full forward passes.

Limitations. The evidence covers one model family, one mathematical training corpus, three competition benchmarks, and a matched 1.7B-to-8B scale suite. Three training seeds do not characterize every source of optimization variance. Evaluating other model families and tasks would test the broader reach of this training recipe.

OPSRD turns a fixed expert role into supervision along the student's own reasoning. A frozen instance of the same base model provides the teacher distribution, and the student learns from it through forward KL at uncertain positions. Across the three tested scales, forward KL achieves the highest macro score among the evaluated divergences, with the clearest advantage over reverse KL at 1.7B. Training-seed repeats retain the gain over Base, and the trained student solves problems with an ordinary prompt. A role can therefore be useful even when direct prompting offers no measured improvement. Its teaching value lies in the guidance it supplies along the student's reasoning.

Reproducibility Statement

The supplement contains exact prompts, launch settings, executed training code, and scoring evidence. Scripts regenerate tables, figures, statistics, and the paper. Appendix D defines the resampling units; Appendix B specifies the loss reduction. The release records library versions and the source of each experimental result. Both method diagrams can be regenerated as editable vectors from the supplied figure source.

AI Use Statement

During the experiments, we used large language models (LLMs) as an auxiliary tool for experiment monitoring and management. Specifically, the LLMs were used to monitor experiment logs and runtime information, identify potential execution issues or anomalies, and assist in reporting the status of ongoing experiments. The LLMs did not determine the research questions, experimental methodology, hyperparameter settings, or final experimental conclusions. All experimental configurations, result verification, analysis, and scientific conclusions were determined and validated by the authors. We take full responsibility for the final content and results of this work.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes, 2026. URL https://arxiv.org/abs/2603.25562.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models, 2022. URL https://arxiv.org/abs/2212.10670.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 2026. URL https://arxiv.org/abs/2605.06387.

Yinuo Jiang, Yongjie Ye, Zhou Tao, Xiang Zhuang, Qiang Zhang, Huajun Chen, and Tiankai Li. When teachers mislead: Spurious-signal-aware on-policy distillation, 2026. URL https://arxiv.org/abs/2608.03632.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Junseok Kim, Nakyeong Yang, and Kyomin Jung. Persona is a double-edged sword: Mitigating the negative impact of role-playing prompts in zero-shot reasoning tasks, 2024. URL https://arxiv.org/abs/2408.08631.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation, 2016. URL https://arxiv.org/abs/1606.07947.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4099–4113. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.228. URL https://aclanthology.org/2024.naacl-long.228/.

Enhan Li, Junhao He, and Hongyang Du. CROP: Task relevance via counterfactuals for selective on-policy distillation, 2026. URL https://arxiv.org/abs/2608.13387.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, January 2023. ISSN 1557-7341. doi: 10.1145/3560815. URL http://dx.doi.org/10.1145/3560815.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022. URL https://arxiv.org/abs/2209.15189.

Noah Wang, Z.Y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.878. URL https://aclanthology.org/2024.findings-acl.878/.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions, 2023. URL https://arxiv.org/abs/2212.10560.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. ExpertPrompting: Instructing large language models to be distinguished experts, 2025a. URL https://arxiv.org/abs/2305.14688.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=EgJhwYR2tB.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1028–1043. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.58. URL https://aclanthology.org/2024.acl-long.58/.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL https://arxiv.org/abs/2601.18734.

A EXPERIMENTAL DETAILS

We provide the prompts, hyperparameters, and objective details needed to reproduce the comparisons. The accompanying code package contains the exact launch configurations, random seeds, individual evaluation records, and scripts for rebuilding the reported summaries.

A.1 DATA AND HYPERPARAMETERS

Training uses the train split of `siyanzhao/Openthoughts_math_30k_opsd`, containing 29,434 examples. Role distillation uses the problem field; Answer OPSD additionally provides the solution to its teacher. We reuse the published split without additional filtering or decontamination. The original runs did not pin a dataset revision. Table 5 summarizes training and evaluation, with the same final checkpoint used for all benchmarks.

Table 5: Training and evaluation settings.

Parameter	Setting
Optimizer / steps	AdamW / 100
Learning rate / maximum gradient norm	5×10^{-6} / 0.1
Effective batch size / rollouts per problem	36 / 1
LoRA rank / scale	64 / 128
LoRA projections	q, k, v, o, gate, up, down
Training temperature / top-p / top-k	1.1 / 0.95 / 20
Training rollout cap	1,024 tokens
Distillation temperature / vocabulary	1.1 / Full
Contribution cap at 1.7B and 4B / at 8B	0.05 / 0.06
Selected fraction for OPSRD	0.50 per sequence
Precision / attention	BF16 / SDPA
Evaluation samples per problem	12
Evaluation temperature / top-p	1.0 / 0.95
Evaluation top-k	Disabled
Evaluation generation cap / context limit	38,912 / 40,960 tokens

Answer OPSD is an internal reproduction with the same models, tasks, and metric as the cited study. Our effective batch of 36, SDPA attention, and fixed final checkpoint differ from its batch-32, FlashAttention-2 setup and checkpoint selection. Its published scores and our reproduced scores consequently refer to different training and evaluation configurations.

A.2 PROMPTS

Expert role. The teacher receives the following system message for every problem.

> You are an expert olympiad mathematician. Solve the problem with rigorous, independent reasoning. Identify the key structure before calculating, verify each non-trivial step, reconsider doubtful branches, and check the final answer.

Neutral context. The generic control shares the expert teacher's thinking mode and system-message placement, with the following instruction. It controls for an additional instruction, without matching the expert message's length or semantics.

> You are an assistant. Respond to the user's request clearly and carefully. Follow the requested output format, keep the response relevant, and review the answer before finishing.

Problem and answer format. The training user message contains `Problem:`, the problem, two newlines, and the instruction below. The Qwen3 chat template disables student thinking and enables teacher thinking. Evaluation enables thinking and omits the literal `Problem:` prefix; these choices remain fixed across the prompt and placement controls.

> Please reason step by step, and put your final answer within `\boxed{}`.

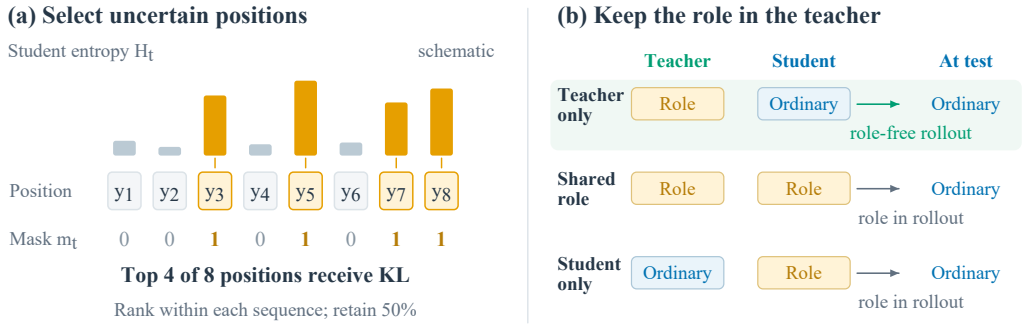

Figure 5: Selection and role placement. (a) Entropy ranking retains four of eight schematic positions. (b) Teacher-only conditioning keeps the student role-free during training and deployment.

Shared-role and student-only variants add the expert message before student sampling. Each checkpoint is evaluated both with and without that system message.

Reference-solution teacher. Answer OPSD encloses the solution between `=== Reference Solution Begin ===` and `=== Reference Solution End ===`. It then adds the following instruction and the boxed-answer request above.

> After reading the reference solution above, make sure you truly understand the reasoning behind each step—do not copy or paraphrase it. Now, using your own words and independent reasoning, derive the same final answer to the problem above. Think step by step, explore different approaches, and don't be afraid to backtrack or reconsider if something doesn't work out:

All teacher variants score the student's sampled completion after their respective contexts. Role and neutral distillation omit the reference solution from the model input.

B Algorithm Details

B.1 Training Step

For each batch, the adapted student first samples one completion per problem. The completion is appended to both contexts, and the student and frozen teacher score identical sampled positions. The teacher pass disables the student's adapter. Teacher probabilities and sampled tokens receive no gradients during student optimization.

Student entropy is computed from full-vocabulary float32 probabilities. Independently for each sequence, we retain the $\max(1, \lceil \rho T \rceil)$ valid positions with highest entropy (Figure 5). Padding and prompt positions are excluded. At the retained positions, we cap individual vocabulary contributions, sum over the full vocabulary, and average over the selected tokens. Entropy computation is chunked to reduce peak memory, preserving the same per-position values as the unchunked calculation.

The loss is evaluated per device and microbatch, after which distributed training averages local losses and accumulates gradients. With unequal selected-token counts, this reduction differs from weighting every token globally across devices. The implementation preserves the local reduction for all comparisons. Both full forward passes remain necessary after entropy selection.

B.2 Clipped Gradient

For a selected position, let $A_t = \{v : d_{t,v} < \tau\}$ denote contributions below the cap. Away from the cap boundary, the derivative of the implemented loss with respect to a student logit is

$$\frac{\partial \ell_t}{\partial z^t_{S,j}} = \frac{1}{\eta} \left(p^t_S(j) \sum_{v \in A_t} p^t_T(v) - \mathbf{1}[j \in A_t] p^t_T(j) \right). \tag{9}$$

Table 6: Per-benchmark scores from the primary evaluation at each scale.

Model	Method	AIME24	AIME25	HMMT25	Macro	Δ Base
Qwen3-1.7B	Base	46.67	36.11	24.44	35.74	0.00
Qwen3-1.7B	Answer OPSD	51.39	39.17	25.56	38.70	+2.96
Qwen3-1.7B	OPSRD	57.78	41.67	26.67	42.04	+6.30
Qwen3-4B	Base	73.61	66.11	41.11	60.28	0.00
Qwen3-4B	Answer OPSD	75.00	68.61	43.89	62.50	+2.22
Qwen3-4B	OPSRD	72.22	67.78	44.44	61.48	+1.20
Qwen3-8B	Base	75.56	69.44	43.33	62.78	0.00
Qwen3-8B	Answer OPSD	75.83	71.67	45.28	64.26	+1.48
Qwen3-8B	OPSRD	77.22	71.39	48.89	65.83	+3.06

Table 7: Per-benchmark divergence results. JSD uses recorded aggregate scores.

Model	Objective	GPU	AIME24	AIME25	HMMT25	Macro
1.7B	Forward KL	A100	57.78	41.67	26.67	**42.04**
1.7B	Reverse KL	A100	50.56	35.56	25.56	37.22
1.7B	JSD	H100	51.39	38.61	22.22	37.41
4B	Forward KL	A100	72.22	67.78	44.44	**61.48**
4B	Reverse KL	A100	74.72	67.50	40.56	60.93
4B	JSD	H100	71.94	65.28	43.06	60.09
8B	Forward KL	A100	77.22	71.39	48.89	**65.83**
8B	Reverse KL	A100	75.00	70.00	46.39	63.80
8B	JSD	H100	75.56	69.17	45.28	63.33

Only contributions below the cap differentiate through $-p_T^t(v)\log p_S^t(v)$. When every contribution is active, the gradient reduces to $(p_S^t(j) - p_T^t(j))/\eta$. The framework selects a one-sided derivative at the cap boundary. Capped vocabulary entries have zero direct derivative but still affect other entries through softmax normalization. Teacher outputs and the entropy ranking remain detached.

C Additional Results

C.1 Benchmark Scores

Table 6 gives per-benchmark results for the primary evaluation at each model size. The 4B values here correspond to one evaluation of the same checkpoints used in the repeated-evaluation summary in the main paper. Each condition has 12 samples for each of the 90 test problems.

C.2 Divergence Ablation

Forward and reverse KL respectively use $\mathrm{KL}(p_T\|p_S)$ and $\mathrm{KL}(p_S\|p_T)$. JSD uses the equally weighted mixture $M = (p_T + p_S)/2$,

$$D_{\mathrm{JSD}} = \tfrac{1}{2}\mathrm{KL}(p_T\|M) + \tfrac{1}{2}\mathrm{KL}(p_S\|M). \tag{10}$$

Each objective retains the full vocabulary, applies the same per-contribution upper cap, and reduces over the selected positions. All variants use the expert teacher, role-free student, 50% entropy fraction, and the training budget in Table 5.

The KL runs use A100 and JSD uses H100. Device counts and microbatch layouts differ in some runs while preserving the nominal effective batch; local token reduction and rollout scheduling can therefore vary. The paired forward-versus-reverse intervals describe problem uncertainty for one training run per condition. They are $[2.41, 7.31]$, $[-1.85, 2.96]$, and $[-0.56, 4.81]$ at 1.7B, 4B, and 8B. JSD scores are supported by contemporaneous aggregate records; individual JSD sample files are unavailable in the release.

C.3 Role Placement

Table 8 separates the placement comparison by benchmark. The role-free student and role teacher define the teacher-only condition. Shared role conditions both models; student only conditions

Table 8: Complete placement ablation, relative to teacher-only training with no test role.

Training role	Test role	AIME24	AIME25	HMMT25	Macro	Δ
Teacher only	No role	57.78	41.67	26.67	42.04	0.00
Teacher only	Expert role	51.11	41.94	25.28	39.44	-2.59
Shared role	No role	52.22	40.83	23.89	38.98	-3.06
Shared role	Expert role	47.50	39.44	23.89	36.94	-5.09
Student only	No role	50.28	39.72	24.72	38.24	-3.80
Student only	Expert role	47.22	38.33	23.33	36.30	-5.74

Table 9: Entropy selection in the primary run.

Quantity	Mean
Selected valid-token fraction	0.500
Selected-position entropy	0.874
Omitted-position entropy	0.023
Unclipped selected divergence	0.194

the rollout generator. All variants use the same 1.7B protocol and selection fraction. Relative to teacher-only training with ordinary evaluation, shared-role and student-only training have paired macro intervals of $[-5.83, -0.37]$ and $[-6.48, -1.30]$ under ordinary evaluation.

C.4 Selection Sensitivity

Selected positions have much higher student entropy than omitted positions (Table 9). This confirms that the mask concentrates the objective on uncertain predictions. The measured divergence also shows a nonzero teacher signal at the retained positions.

The fraction pilot compares 25%, 50%, and 75% selection with four samples per problem and the same generation cap as the main evaluation (Table 10). The 50% reference uses its first four matched samples. The 25% result ties the reference and 75% scores lower. Neither reaches the one-point improvement required for a full-budget follow-up. These exploratory measurements motivate retaining the default fraction; the principal density comparison uses the complete 50% and 100% evaluations.

D Statistical Analysis

Metric. For benchmark b with 30 problems and $S = 12$ samples per problem, let $a_{j,s}$ denote answer correctness. We compute

$$\mathrm{Avg@12}_b = \frac{100}{30S} \sum_{j=1}^{30} \sum_{s=1}^{S} a_{j,s}. \tag{11}$$

The macro score averages the three benchmark scores. Equal problem and sample counts make it identical to pooled accuracy over 1,080 generations. Observed success coverage instead counts problems with any correct sample, which is the quantity contrasted with average accuracy in Figure 4.

Paired intervals. We average each problem's 12 correctness labels, align conditions by benchmark and problem, and resample their paired accuracy differences. Per-task intervals resample 30 problems; pooled intervals resample all 90 problems. Primary comparisons use 10,000 bootstrap draws and the divergence comparison uses 50,000. This preserves the problem as the sampling unit rather than treating repeated generations as independent problems.

Training and evaluation repeats. For three-run training comparisons, each of 30,000 bootstrap draws resamples three runs and 30 problems within each benchmark. The same problem draw is shared across the sampled runs, preserving dependence from the common test set and fixed Base evaluation. Reported standard deviations are sample standard deviations across the stated repeats. The 4B repeats hold trained weights fixed and span A100 and H100; their dispersion combines decoding and environment variation. Training variability is assessed separately at 1.7B.

Table 10: Selection-fraction pilot with four samples per problem.

Fraction	Avg@4	Δ 50%
25%	43.06	0.00
50%	43.06	0.00
75%	38.89	−4.17

Study design. Intervals apply to the stated comparison without adjustment for multiple ablations. Additional training runs followed a favorable primary result, and alternative selection fractions were screened at a smaller sampling budget. These comparisons form an exploratory study. Each summary includes every completed repeat or model scale belonging to that comparison.

E Broader Impact

The experiments use public mathematical problems and model-generated reasoning, with no human subjects or private records. Expertise labels require task-specific validation before deployment because a role can change model behavior unpredictably. Training uses student and teacher forward passes, while inference uses only the adapted student and its ordinary prompt.