OVD: ON-POLICY VERBAL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation transfers reasoning capabilities from large teachers to efficient students. However, token-level on-policy distillation (OPD) constrains student exploration and requires teacher token probabilities, precluding distillation from black-box teachers that provide only text outputs. We introduce *On-policy Verbal Distillation* (OVD), a framework that uses verbal scores from black-box teachers to rank student-generated sub-trajectories, retaining high-scoring ones and replacing low-scoring ones with teacher-generated continuations. We analyze when ranking induced by verbal scores can guide distribution approximation: under a density-ratio calibration condition on acceptance probabilities and bounded teacher-replacement error, we bound the approximation error between the resulting mixed trajectory distribution and a teacher-preferred target. On Web Q&A, OVD achieves 41.09% average EM with teacher feedback at inference, exceeding the strongest evaluated baseline by 5.89 percentage points. On AMC23, OVD-FR improves accuracy over RLVR by 10.0 percentage points (52.5% to 62.5%) after 600 training steps on 128 problems. Further experiments suggest that retaining student-generated prefixes helps preserve exploration and mitigate trajectory-level entropy collapse. OVD also improves training efficiency: resampling selected suffixes rather than entire responses reduces mean per-step training time by 10.2% in the 128-problem setting. Code is available at `https://anonymous.4open.science/r/OVD-review-3BDA/OVD_math/README.md`.

1 INTRODUCTION

Knowledge distillation transfers capabilities from large teachers to smaller policies. On-policy distillation (OPD) learns from student rollouts but typically matches next-token distributions (Hinton et al., 2015; Agarwal et al., 2024). For long-horizon reasoning, particularly in agentic settings, these dense probabilities are costly to retain and unavailable from black-box teachers, while token-level matching may penalize deviations from the teacher at every step, potentially suppressing trajectory-level exploration. These limitations motivate a trajectory-level alternative that uses verbal feedback to shape the distribution of on-policy student rollouts toward trajectories preferred by the teacher.

We introduce *On-policy Verbal Distillation* (OVD). A teacher scores each student sub-trajectory; high-scoring segments are retained, low-scoring ones are replaced with teacher continuations, and the student then resumes generation. This procedure preserves student exploration while steering the induced mixed-trajectory distribution toward a teacher-preferred target; the resulting trajectories are then used for outcome-based policy optimization.

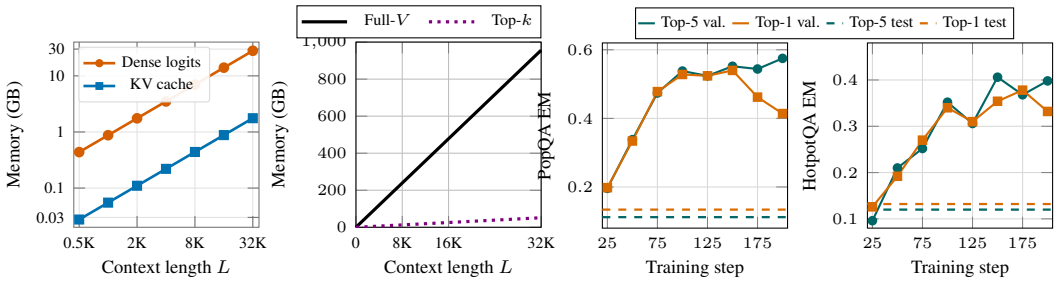
1.1 LIMITATIONS OF TOKEN-LEVEL OPD

Standard OPD. Let π_S and $\pi_{\mathcal{E}}$ denote the student and teacher policies, respectively. OPD (Agarwal et al., 2024) trains on trajectories sampled from π_S while matching the next-token distribution of π_S to the *full-vocabulary* distribution of $\pi_{\mathcal{E}}$ (dense logits over all V tokens at each decoding step). For a problem q, its objective is the reverse KL over next tokens x_ℓ sampled from π_S,

$$\mathrm{KL}\big(\pi_S \| \pi_{\mathcal{E}}\big) = \mathbb{E}\Big[\log \pi_S(x_\ell \mid q, x_{<\ell}) - \log \pi_{\mathcal{E}}(x_\ell \mid q, x_{<\ell})\Big]. \tag{1}$$

For policy optimization, the sampled negative log-ratio $\log \pi_{\mathcal{E}}(x_\ell \mid q, x_{<\ell}) - \log \pi_S(x_\ell \mid q, x_{<\ell})$ serves as the token-level advantage. Its expectation under π_S equals the negative reverse KL, so maximizing the corresponding importance-weighted policy objective minimizes Eq. (1) (Lu & Thinking Machines Lab, 2025). We refer to this full-vocabulary objective as *standard OPD*.

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) Per-trajectory memory. (b) Aggregate (N=32). (c) PopQA. (d) HotpotQA.

Figure 1: Two limitations of token-level distillation. **(a,b)** On Qwen2.5-7B, full-vocabulary FP32+BF16 logits scale linearly with context length: at L=8192, they require 7.0 GB per trajectory versus 0.44 GB for the BF16 KV cache, and approximately 240 GB in aggregate for N=32 versus 13 GB for top-k (k=8192). **(c,d)** On PopQA and HotpotQA, simulator validation EM generally increases over training under top-1/top-5 token-level distillation, whereas held-out live-search EM is much lower; top-5 underperforms top-1 on both datasets.

Findings 1: *Full-vocabulary probability matching creates a memory bottleneck*

Memory Bottleneck. The full-vocabulary objective above requires dense student and teacher distributions over all V tokens at every decoding step of each student-generated trajectory. Figure 1(a,b) shows that dense logit storage grows linearly with context length on Qwen2.5-7B (Yang et al., 2024) and becomes substantially larger when aggregated over N=32 rollouts. Closed-source teachers also typically do not expose such distributions. With a batch of B problems, N trajectories per problem, and rollout length L, a standard full-vocabulary logit-storage budget is

$$\mathcal{M} = B \cdot N \cdot L \cdot V \cdot (d_{32} + d_{16}) = 1.5 \cdot B \cdot N \cdot L \cdot V \cdot d_{32}. \tag{2}$$

Here d_{32}=4 and d_{16}=2 bytes; many stacks materialize logits in both precisions (Wijmans et al., 2024), using BF16 for most training computation and FP32 for numerically sensitive operations such as softmax and KL. Consequently, full-vocabulary logit storage scales as $\mathcal{O}(B \cdot N \cdot L \cdot V)$. At long context lengths and large rollout counts, this scaling makes logit storage a dominant memory bottleneck, motivating sparse alternatives.

Top-k OPD. One extreme retains only the teacher log-probability assigned to each student-sampled token, exactly matching the sampled-token reverse-KL estimator of Lu & Thinking Machines Lab (2025). A broader sparse alternative retains the teacher's k highest-probability tokens per decoding step. For this top-k OPD baseline, we restrict both the student and teacher distributions to this support and renormalize them before matching. This lowers logit-storage costs to $\mathcal{O}(B \cdot N \cdot L \cdot k)$, an approximately V/k-fold reduction, but top-k OPD still relies on token-level probability matching and requires white-box teacher access to obtain truncated logits. How the choice of k affects policy exploration remains an open question.

Findings 2: *Token-level probability matching may restrict exploration*

Token-level matching to the teacher's next-token distribution (Hinton et al., 2015; Gu et al., 2024) may reduce behavioral diversity (Agarwal et al., 2024). In the agentic setting, Figures 1c and 1d show that simulator validation EM generally increases over training under both top-1 and top-5 token-level distillation, whereas held-out live-search EM is much lower, with top-5 underperforming top-1 on both datasets. This pattern suggests that denser token-level matching may narrow policy exploration.

1.2 Distribution Approximation via Ranking

These limitations motivate our central question: *under black-box teacher access, can ranking on-policy student trajectories with discrete verbal scores approximate a teacher-preferred distribution while still allowing the student to explore its trajectory space?*

We use a *verbal scoring interface* in which the teacher provides quality assessments as discrete score tokens rather than full-vocabulary logits. Given a problem q and a student-generated sub-trajectory y,

the teacher induces a distribution over a verbal vocabulary of size $v = 10$. By default, we greedily decode the highest-probability score token and use it as $S(y, q) \in \{0, 1, \ldots, 9\}$, with higher values indicating stronger preference for the sub-trajectory. When logits for these score tokens are available, applying a softmax over the ten logits yields the score distribution, from which scores can instead be sampled. The default greedy procedure remains compatible with black-box access because it does not require the logits to be exposed.

Findings 3: *Ranking induced by verbal scores can select high-quality sub-trajectories*

Figure 2 asks whether the teacher can select promising initial prefixes. In the verbal-score condition, the teacher scores each initial prefix; given a test-time threshold τ_{te}, the student continues from prefixes whose scores are at least τ_{te}. In the *per-prompt random* condition, prefixes are chosen randomly within each prompt, preserving the number selected by verbal-score selection. In the *pooled random* condition, all prefixes are pooled together. We randomly select the same total number of prefixes as in the verbal-score condition, without requiring the selected prefixes to be distributed across prompts in the same way. If verbal scores successfully identify promising initial prefixes, continuing from the selected prefixes should yield higher final-answer EM than either random baseline. Consistent with this prediction, verbal-score selection outperforms both baselines; in particular, its gain over per-prompt random provides evidence that the scores distinguish prefix quality within a prompt.

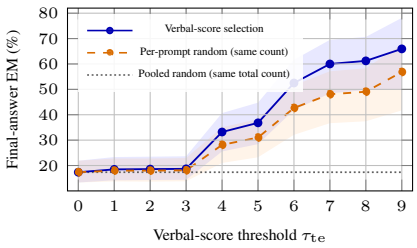


Figure 2: Initial-prefix selection and continuation EM on a fixed 128-problem subset of the DeepScaleR-Preview-Dataset, with 32 fixed student-generated prefixes per problem (4,096 total), reused across thresholds. The teacher scores only prefixes; those scoring at least τ_{te} receive student-generated continuations for EM evaluation. At τ_{te}=8, 441 prefixes remain: EM is 61.22% for verbal-score selection, 49.06% for per-prompt random, and 17.36% for pooled random. Shading denotes 95% confidence intervals.

OVD uses this signal for *verbal rejection sampling*, retaining strong student sub-trajectories and replacing weak ones with teacher continuations. Thresholding verbal scores converts the teacher's ranking into acceptance probabilities a_τ, which reweight student proposals. We analyze when this reweighting recovers a teacher-preferred distribution p_T: under a valid density-ratio condition $a_\tau(y \mid q) \propto p_T(y \mid q)/\pi_S(y \mid q)$, the accepted student component matches p_T. Ranking accuracy alone does not establish this condition. Under approximate calibration and bounded teacher-replacement error at each prefix, we further quantify how these errors accumulate across reasoning steps and bound the resulting deviation from p_T. Section 2.2 formalizes this distributional framework. **Contributions.**

- We introduce OVD, a black-box on-policy distillation framework that steers student rollouts with verbal scores and selective teacher continuations, without requiring teacher logits.
- We identify sufficient conditions under which verbal-score filtering and teacher replacement approximate a teacher-preferred trajectory distribution, and quantify the approximation error.
- We evaluate OVD across eight Web Q&A benchmarks and tool-free mathematics, examining the effectiveness of ranking induced by verbal scores and selective teacher resampling.

2 On-policy Verbal Distillation

In this section, we present OVD, which explores how verbal rejection sampling over on-policy student rollouts can approximate a teacher-preferred trajectory distribution. It then trains the student on the resulting mixed trajectories.

Problem Setup. Consider a teacher model $\pi_{\mathcal{E}}$ and a student policy π_S. Given a problem q, the student policy generates multi-step reasoning trajectories, while the teacher model provides verbal feedback scores. We write a trajectory as $y = (s_1, \ldots, s_T)$, where $t \in \{1, \ldots, T\}$ indexes the reasoning steps, each s_t is a reasoning step (e.g., a sentence or a logical derivation), $y_{<t} = (s_1, \ldots, s_{t-1})$ is the trajectory prefix, and T is the number of steps. We reserve x_ℓ for token-level quantities, where ℓ indexes token positions; in Eq. (1), standard OPD matches $\pi_S(x_\ell \mid q, x_{<\ell})$ to $\pi_{\mathcal{E}}(x_\ell \mid q, x_{<\ell})$ over the vocabulary. By contrast, s_t indexes *coarse* segments of the rollout, not

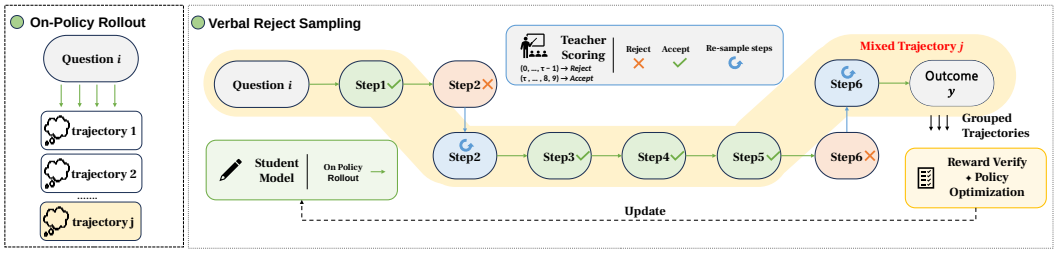


Figure 3: Student on-policy rollouts (left); teacher-guided rejection sampling and simulated search interaction for Web Q&A (right).

single tokens. We use τ for a generic score threshold and write τ_{tr} and τ_{te} for the training- and test-time thresholds, respectively.

2.1 On-policy Rollouts

Rollout Generation. The student explores reasoning paths by sampling candidate sub-trajectories from its current policy. After a teacher replacement, the student resumes generation conditioned on the updated prefix. At each iteration, we sample a batch of B problems $\{q_i\}_{i=1}^{B}$ and generate N on-policy rollouts for each problem, yielding $B \times N$ trajectories in total. Here i indexes problems in the batch and j indexes the N rollouts for each q_i: $y_i^{(j)} \sim \pi_S(\cdot \mid q_i)$ for $j = 1, \ldots, N$. Each trajectory $y_i^{(j)} = \left(s_1^{(i,j)}, \ldots, s_{T_{ij}}^{(i,j)}\right)$ has variable-length reasoning steps. Step boundaries are task-specific. In Web Q&A, the `<search>`, `<information>`, and `<think>` markers provide explicit boundaries (Appendix A.3). Mathematical rollouts have no such markers. We therefore compute the student's next-token entropy along each rollout and use the maximum-entropy position, which often marks a branch in the reasoning path, as the split point; the suffix beginning at that position is treated as a sub-trajectory for verbal rejection sampling. On-policy rollouts let the teacher judge sub-trajectories generated by the current student policy, avoiding the distribution mismatch that arises when training on pre-generated teacher outputs.

Verbal Rejection Sampling. We use the verbal scoring interface in Section 1.2 to score student sub-trajectories. If a sub-trajectory scores below the training threshold τ_{tr}, the teacher resamples and replaces that sub-trajectory. After the teacher's replacement is inserted, the student generates the remaining trajectory from its end. This produces *mixed* trajectories that interleave student-authored prefixes with sub-trajectories resampled by the teacher. Because the teacher replaces only low-scoring sub-trajectories, the training trajectories remain largely grounded in the student's on-policy exploration. The mixed trajectories are used to train the student with outcome-based rewards.

2.2 Distribution Approximation via Verbal Rejection Sampling

Constructing the Mixed Training Distribution. *Classical rejection sampling* (Devroye, 1986) transforms a proposal distribution into a target distribution by accepting proposals according to a suitably bounded target-to-proposal density ratio. Related methods adapt this distribution-shaping principle to different learning objectives. Density-ratio-based methods correct off-policy distribution mismatch using explicit policy ratios (Chung et al., 2018; Schlegel et al., 2019). Reward-model-based statistical rejection sampling selects complete responses for preference optimization (Liu et al., 2023), while reward-based reasoning methods filter high-reward samples (Xiong et al., 2025). *Verbal rejection sampling* uses teacher-assigned verbal scores to reweight the student's on-policy trajectory distribution toward a target $p_T(y \mid q)$ that reflects the teacher's preferences over trajectories. For a fixed teacher and threshold τ, we define the acceptance probability as

$$a_\tau(y \mid q) = \Pr[S(y, q) \geq \tau \mid y, q], \tag{3}$$

where $S(y, q)$ is the teacher-assigned verbal score and $\Pr$ denotes probability over randomness in teacher scoring. With deterministic scoring, this reduces to $a_\tau(y \mid q) = \mathbf{1}[S(y, q) \geq \tau]$. Conditioning student proposals on acceptance induces the verbally filtered trajectory distribution

$$p_{\mathrm{a},\tau}(y \mid q) = \frac{\pi_S(y \mid q)\, a_\tau(y \mid q)}{Z_\tau(q)}, \qquad Z_\tau(q) = \mathbb{E}_{y' \sim \pi_S(\cdot \mid q)}[a_\tau(y' \mid q)], \tag{4}$$

where $Z_\tau(q) > 0$ is the acceptance rate and normalizing constant. Equation (4) characterizes the accepted student component of the training distribution. It follows directly from conditioning on acceptance and holds for any acceptance rule with a nonzero acceptance rate. It captures the same distribution-reweighting principle as classical rejection sampling. Classical rejection sampling chooses acceptance probabilities from a target-to-proposal density ratio, whereas our method obtains them from thresholded verbal scores.

To include teacher resampling, fix a problem q and consider a candidate reasoning step s_t at trajectory prefix $y_{<t}$. We suppress the fixed q in the local conditionals. At this level, $a_\tau(s_t \mid y_{<t})$ is the probability of accepting the candidate, and $Z_\tau(y_{<t}) = \mathbb{E}_{s_t \sim \pi_S(\cdot \mid y_{<t})}[a_\tau(s_t \mid y_{<t})]$ is its average over student proposals. Let $r_{\mathcal{E},\tau}(s_t \mid y_{<t})$ denote the normalized distribution of teacher replacements conditional on rejection, averaging over rejected candidates and feedback when the replacement depends on them. The reasoning step inserted at this prefix follows

$$p_{\mathrm{m},\tau}(s_t \mid y_{<t}) = \underbrace{\pi_S(s_t \mid y_{<t})\, a_\tau(s_t \mid y_{<t})}_{\text{retained student exploration}} + \underbrace{\big(1 - Z_\tau(y_{<t})\big)\, r_{\mathcal{E},\tau}(s_t \mid y_{<t})}_{\text{teacher guidance}}. \tag{5}$$

The student proposal $\pi_S(s_t \mid y_{<t})$ drives exploration by generating candidate reasoning segments. The first term retains accepted student proposals and has total mass $Z_\tau(y_{<t})$, while the second introduces teacher guidance through resampled replacements and has mass $1 - Z_\tau(y_{<t})$. Rejected student proposals also arise from exploration, but are replaced in the training trajectory. After each replacement, the student resumes exploration from the updated prefix. Repeatedly applying the retain-and-replace procedure in Eq. (5) induces a distribution over complete mixed trajectories, denoted by $p_{\mathrm{m},\tau}(y \mid q)$, with $\tau = \tau_{\mathrm{tr}}$ during training. Each policy token inherits the source of its sub-trajectory. We record this source using binary indicators m_ℓ^S and m_ℓ^T, which identify tokens in retained student segments and teacher-resampled segments, respectively. These indicators are used to decompose the policy optimization objective in Section 2.3.

Density-Ratio Condition. To analyze when verbal rejection sampling follows the acceptance rule of classical rejection sampling, we consider the following density-ratio condition on the probability that a trajectory's verbal score meets the threshold:

$$a_\tau(y \mid q) \propto \frac{p_T(y \mid q)}{\pi_S(y \mid q)} \tag{6}$$

Takeaway. *Thresholding verbal scores at τ converts their ranking into acceptance. The mixture approaches the target when acceptance tracks the target-to-student density ratio and the replacement distribution is close to the target.*

Under the exact proportionality condition in Eq. (6), the accepted student distribution matches p_T. For the mixed training distribution to also match p_T, teacher replacements must follow the corresponding target distribution at each prefix where replacement occurs. In practice, approximation error relative to this ideal case arises from two sources: verbal-score acceptance probabilities may deviate from the density ratio, and teacher replacements may deviate from the target.

The approximation question thus concerns both the accepted student component and the teacher replacements. The threshold controls their relative contributions: at prefix $y_{<t}$, teacher replacement is invoked with probability $1 - Z_\tau(y_{<t})$. Appendix A.2 formalizes an *Approximate Density-Ratio Condition* and gives error bounds for the accepted and mixed distributions in Proposition A.2. Figure 4 shows that verbal acceptance generally increases with the teacher-to-student log density ratio before leveling off, supporting the distribution-reweighting rationale of our analysis.

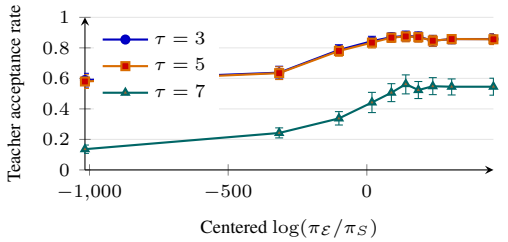


Figure 4: Verbal acceptance versus teacher-to-student log density ratio on 6,400 Math500 answers. Points show within-question-centered bin estimates; bars indicate 95% question-bootstrap intervals.

2.3 Optimization Objective

During training, the retain-and-replace procedure produces mixed trajectories, whose outcome rewards determine group-relative advantages. These advantages are applied to both student-generated

and teacher-resampled tokens: verbal rejection sampling shapes the training data, while outcome-based advantages determine the direction of policy optimization. The complete OVD training pipeline is illustrated in Figure 3.

Reward Design. For trajectories, we compute the *outcome-based reward* $R(y)$ as:

$$R(y) = \mathrm{F}_1(y, y^*) \text{ (Web Q\&A)}, \qquad R(y) = \delta(y, y^*) \text{ (math)}. \tag{7}$$

where y^* denotes the ground-truth reference answer, $\mathrm{F}_1(y, y^*) = \frac{2|y \cap y^*|}{|y|+|y^*|}$ measures word overlap, and $\delta(y, y^*) \in \{0, 1\}$ denotes exact matching under mathematical equivalence (numerical tolerance, algebraic normalization). This design provides soft partial credit for open-ended answers and strict binary feedback for correctness.

Policy Gradient Objective. For each problem q_i, we sample a group of complete trajectories $y_i^{(j)} \sim p_{\mathrm{m},\tau_{\mathrm{tr}}}(\cdot \mid q_i)$, $j = 1, \ldots, N$, using the retain-and-replace procedure in Section 2.2. For each training batch, this sampling distribution is constructed using the student policy before the update, π_S^{old}; the sampled trajectories and their token-source indicators are held fixed during optimization. Inspired by GRPO (Shao et al., 2024), we compute group-relative advantages and student-policy probability ratios on mixed trajectories:

$$A(y_i^{(j)}) = \frac{R(y_i^{(j)}) - \mu_i}{\sigma_i + \epsilon}, \tag{8}$$

$$\rho_\ell = \frac{\pi_S(x_\ell \mid q, x_{<\ell})}{\pi_S^{\mathrm{old}}(x_\ell \mid q, x_{<\ell})}. \tag{9}$$

Here, μ_i and σ_i are the mean and standard deviation of outcome rewards within the mixed-trajectory group for q_i. We recompute old-student log probabilities on the complete mixed trajectories and hold them fixed during optimization. Both expressions apply to student-generated and teacher-resampled tokens. For *Web Q&A*, tool-returned *information* spans remain masked out of the policy-gradient and KL terms. The update applies to search-query and reasoning tokens, including teacher-resampled replacements, rather than environment observations. At every unmasked position, the source indicators satisfy $m_\ell^S + m_\ell^T = 1$.

Trajectories whose sub-trajectories are all accepted remain entirely student-generated. They contribute to policy optimization without matching the teacher's token-level probability distribution, preserving room for student exploration. Teacher-resampled segments provide guidance under the same outcome-based advantages. To distinguish these contributions, define the clipped token objective as

$$g_\ell(y) = \min(\rho_\ell A(y), \mathrm{clip}(\rho_\ell, 1 - \epsilon_c, 1 + \epsilon_c) A(y)), \tag{10}$$

where we use PPO-style clipping (Schulman et al., 2017) with $\epsilon_c = 0.2$. We optimize over these samples from $p_{\mathrm{m},\tau_{\mathrm{tr}}}$, writing $\mathbb{E}_{\mathrm{m}}$ for the expectation over training problems, sampled trajectory groups, their recorded token sources, and unmasked token positions. The objective decomposes by token source as

$$\mathcal{L}_{\mathrm{RL}} = \underbrace{-\mathbb{E}_{\mathrm{m}}\left[m_\ell^S g_\ell(y)\right]}_{\text{learning from student exploration}} + \underbrace{-\mathbb{E}_{\mathrm{m}}\left[m_\ell^T g_\ell(y)\right]}_{\text{learning from teacher guidance}}. \tag{11}$$

Both terms use the same sampling distribution and normalization, so their sum equals $-\mathbb{E}_{\mathrm{m}}[g_\ell(y)]$. For entirely student-generated trajectories, the teacher-guidance term is zero. Positive advantages reinforce tokens from either source, while negative advantages suppress them. Thus, the student learns from its own exploration and from teacher guidance through a shared outcome-based objective.

3 Experiments

3.1 Evaluation Tasks

We evaluate OVD in two complementary settings: Web Q&A tasks in an agentic setting, where trajectories decompose into multiple search and chain-of-thought (CoT) steps, and mathematical reasoning tasks that provide a tool-free setting for isolating the effects of tool use on distillation.

Web Q&A Tasks. We evaluate on eight question-answering benchmarks spanning both single-hop and multi-hop reasoning: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), Musique (Trivedi et al., 2022), Bamboogle (Press et al., 2023), and GAIA. GAIA is the most challenging benchmark in

Table 1: Web Q&A results with Qwen-2.5-3B-Base as the student and Qwen2.5-7B as the teacher. OVD rows give $(\tau_{\text{tr}}, \tau_{\text{te}})$. Best results are in bold. †: score-vocabulary sampling at test time; ‡: process reward model (SmartSearch and StepSearch). *: OPD baselines using either full-vocabulary reverse KL in Eq. (1) or its top-k approximation described in Section 1.1.

	Single-Hop QA			Multi-Hop QA					
Method	**NQ**	**TriviaQA**	**PopQA**	**HotpotQA**	**2Wiki**	**Musique**	**Bamboogle**	**GAIA**	**Avg.**
Qwen-2.5-3B-Base									
RAG	31.60	58.00	15.20	24.20	23.20	8.20	15.20	5.40	22.63
RA-Agent	15.20	28.40	6.60	12.60	16.60	2.60	13.60	3.00	12.33
Search-o1	16.60	31.00	8.20	14.80	22.40	5.20	22.40	2.40	15.38
R1	14.20	34.80	20.80	19.60	28.40	6.40	5.56	1.80	16.45
Search-R1	40.60	60.00	44.20	29.20	32.00	11.20	12.50	7.81	29.69
ZeroSearch	43.00	61.60	41.40	33.80	34.60	13.00	13.89	7.27	31.07
OPD* (reverse KL)	14.40	25.40	13.40	13.20	21.60	2.20	8.80	2.90	14.14
OPD* (top-k logits)	9.80	20.60	11.20	12.00	17.00	3.20	7.20	2.50	11.51
SmartSearch‡	39.80	64.40	41.20	38.40	**45.40**	15.20	**28.80**	9.09	35.20
StepSearch‡	16.20	28.00	20.40	18.20	16.20	3.60	17.60	7.27	18.60
OVD (τ_{tr}=5, τ_{te}=0)	**49.20**	67.20	63.20	39.20	35.20	21.80	15.20	9.09	37.51
OVD (τ_{tr}=5, τ_{te}=5)	47.00	67.40	66.40	38.00	37.60	23.20	24.80	7.88	39.04
OVD† (τ_{tr}=5, τ_{te}=5)	48.80	**68.00**	**66.60**	**39.40**	41.40	**24.20**	**28.80**	**11.50**	**41.09**

our evaluation suite, testing general AI assistants on real-world questions that require combinations of reasoning, multimodal understanding, web browsing, and tool use (Mialon et al., 2024).

Mathematical Reasoning Tasks. We evaluate mathematical reasoning on the nine benchmarks in Table 3; dataset descriptions and sources are provided in Appendix F. The DeepScaleR-Preview-Dataset contains approximately 40,000 mathematics problems (Luo et al., 2025). We use one fixed problem or a fixed set of 128 distinct problems from this dataset for the two training regimes, respectively. The 128-problem set also underlies the prefix-selection analysis in Figure 2. For these tasks, we use `deepseek-ai/DeepSeek-R1-Distill-Qwen-7B` (Guo et al., 2025) as the teacher and Qwen2.5-Math-1.5B as the student. The teacher assigns each reasoning sub-trajectory a single quality score, thereby ranking candidates from incorrect or irrelevant to optimal steps toward the solution; Appendix A.10 gives the remaining implementation details.

3.2 Agentic Search Setting

(i) Comparison with token-level OPD. Table 1 compares OVD with full-vocabulary reverse-KL and top-k logit OPD. Without teacher intervention at test time, OVD achieves 37.51% average EM, compared with 14.14% and 11.51%, respectively, and outperforms both baselines on all eight benchmarks. These results support the effectiveness of verbal guidance over token-level probability matching in this agentic setting. *(ii) Overall performance.* Table 1 reports Web Q&A exact match (EM) on eight test sets for Qwen-2.5-3B-Base. OVD outperforms Search-o1, Search-R1, ZeroSearch, SmartSearch, and StepSearch in average EM. *(iii) Test-time score sampling.* The † variant samples the *test-time* score from the teacher's ten-token score distribution at temperature 0.7 instead of selecting the highest-probability score token. Under matched $(\tau_{\text{tr}}, \tau_{\text{te}})$, *test-time* sampling improves average EM from 39.04% to 41.09%. The teacher's most likely score need not reflect a confident judgment. Test-time sampling accounts for this ambiguity when deciding which student continuations to retain.

Table 2: Effect of the part of the response rewritten during training on Web Q&A. Results are macro-average exact match (%) with real search and $(\tau_{\text{tr}}, \tau_{\text{te}}) = (5, 0)$. No teacher is used at test time.

Sampling Target	Single-Hop	Multi-Hop	Overall
Query Rejection	**59.87**	24.10	37.51
Query+CoT Rejection	54.27	26.53	36.93
CoT-only Rejection	55.93	**27.02**	**37.87**

Findings 4: *Score sampling improves test-time performance by preserving teacher uncertainty*

3.3 Where Should Rejection Sampling Be Applied?

The preferred rejection target depends on the task. In this agentic setting, trajectories comprise search-query and CoT sub-trajectories, and Table 2 compares applying rejection sampling to queries, CoT, or

Table 3: Math accuracy after training on one fixed problem or 128 distinct problems. FR and HES use full-response and high-entropy-suffix resampling. Avg. is the macro average over the nine benchmarks; bold marks each method's highest Avg. Benchmark selection and random-suffix results are given in Appendix A.5.

Method	Steps	AIME24	AMC23	CARP	Middle	College	Gaokao-II	MATH	MAWPS	SAT-Math	Avg.
Initial Student	0	13.3	35.0	45.3	30.7	5.2	35.7	34.3	22.5	78.1	33.36
Training on One Fixed Problem											
RLVR	300	6.7	50.0	51.3	42.6	33.0	50.0	66.0	91.3	81.3	52.46
RLVR	500	13.3	47.5	51.7	40.6	34.7	42.9	66.3	94.1	81.3	52.49
RLVR	600	10.0	50.0	51.7	39.6	34.5	50.0	66.9	94.1	81.3	53.13
RLVR	800	10.0	52.5	51.0	39.6	36.3	57.1	67.3	95.2	78.1	**54.14**
OVD-FR	300	6.7	50.0	50.3	42.6	30.6	57.1	65.8	87.4	81.3	52.42
OVD-FR	500	10.0	52.5	50.5	45.5	34.6	50.0	66.4	93.8	84.4	54.19
OVD-FR	600	10.0	55.0	51.0	42.6	36.5	50.0	67.6	94.9	90.6	**55.36**
OVD-FR	800	13.3	60.0	51.8	40.6	37.4	42.9	68.2	95.2	84.4	54.87
OVD-HES	300	6.7	50.0	51.2	38.6	34.7	50.0	66.5	93.4	81.3	52.47
OVD-HES	500	10.0	50.0	50.6	42.6	36.7	42.9	67.3	94.7	84.4	53.24
OVD-HES	600	10.0	50.0	51.7	40.6	37.6	42.9	67.6	95.3	84.4	53.33
OVD-HES	800	13.3	50.0	51.5	43.6	38.1	57.1	69.1	96.0	81.3	**55.56**
Training on 128 Distinct Problems											
RLVR	300	20.0	52.5	52.6	50.5	38.1	50.0	69.2	96.1	71.9	**55.65**
RLVR	500	20.0	55.0	53.6	47.5	39.2	28.6	70.1	96.4	75.0	53.93
RLVR	600	16.7	52.5	54.2	52.5	39.3	35.7	70.6	96.8	78.1	55.15
RLVR	800	16.7	47.5	54.9	48.5	39.3	42.9	71.1	96.8	78.1	55.08
OVD-FR	300	16.7	55.0	53.0	50.5	37.1	42.9	68.8	95.3	71.9	54.57
OVD-FR	500	13.3	52.5	54.1	49.5	39.0	50.0	69.8	96.1	71.9	**55.13**
OVD-FR	600	16.7	62.5	54.6	44.6	39.4	50.0	70.0	96.3	59.4	54.82
OVD-FR	800	16.7	52.5	54.4	41.6	39.0	42.9	70.8	96.6	53.1	51.95
OVD-HES	300	23.3	50.0	53.7	47.5	38.4	64.3	69.1	95.3	75.0	**57.40**
OVD-HES	500	13.3	55.0	54.3	49.5	39.0	35.7	70.0	96.5	75.0	54.25
OVD-HES	600	6.7	57.5	53.9	46.5	38.7	35.7	70.6	96.5	75.0	53.45
OVD-HES	800	13.3	55.0	53.7	51.5	39.4	28.6	70.5	97.0	75.0	53.77

both during training. At fixed thresholds, query-only rejection performs best on single-hop questions, whereas CoT-only rejection achieves the highest multi-hop and overall EM. Rewriting both queries and reasoning steps offers no advantage over these targeted alternatives. For our search-focused Web Q&A experiments, we use query-only rejection as the default; CoT distillation is examined separately in the mathematical-reasoning experiments.

3.4 Can We Approximate the Teacher Distribution at Test Time?

At test time, setting $\tau = \tau_{\text{te}}$ in Eq. (5) determines the acceptance rate $Z_{\tau_{\text{te}}}$: increasing τ_{te} retains fewer student sub-trajectories and invokes the teacher more often. Thus, τ_{te} controls the strength of test-time selection, trading off final-answer quality against teacher intervention. We compare teacher models with different output distributions: *(i)* a prompt-based teacher agent, configured through prompting without task-specific fine-tuning (dashed curves in Fig. 5), with $\tau_{\text{tr}} = 5$ and $\tau_{\text{te}} \in \{0, 5, 10\}$; and *(ii)* an SFT-based teacher agent, fine-tuned on simulated-search trajectories (solid curves in Fig. 5), with $\tau_{\text{tr}} \in \{5, 10\}$ and $\tau_{\text{te}} \in \{5, 10\}$.

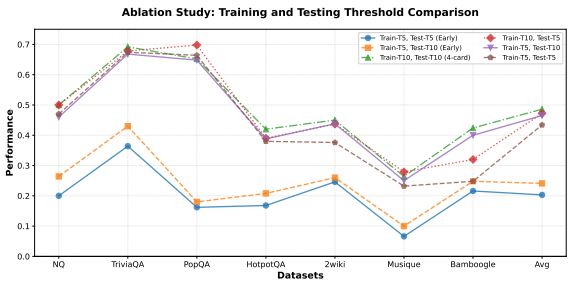


Figure 5: Web Q&A performance under different training- and test-time rejection thresholds. Results are reported for each dataset and as a macro-average (Avg.).

At test time, a moderate threshold of $\tau_{\text{te}} = 5$ outperforms both $\tau_{\text{te}} = 0$ (no teacher intervention) and $\tau_{\text{te}} = 10$ (replacing every scored sub-trajectory). These results suggest that selective teacher guidance during inference is more effective than either accepting all student continuations or replacing them all. *(iii)* Fully replacing the student policy trajectories with those from the teacher during training

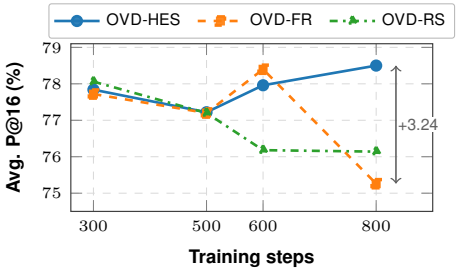


Figure 6: Average P@16 (the percentage of problems with at least one correct answer among 16 sampled responses) on ASDiv (Miao et al., 2020), MAWPS (Koncel-Kedziorski et al., 2016), TABMWP, Olympiad, and AIME24. At 800 steps, OVD-HES reaches 78.50%, versus 76.14% (OVD-RS) and 75.26% (OVD-FR).

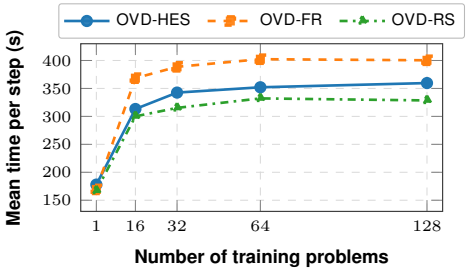


Figure 7: Mean time per step, averaged over 10 updates on four AMD GPUs (batch size 128; eight rollouts per input). Training problems are repeated to keep batch size fixed; shared resources may add timing noise.

(i.e., using $\tau_{\text{tr}} = 10$) does not necessarily yield better gains, as it shifts the mixed training distribution almost entirely toward teacher-generated trajectories, replacing even potentially correct student trajectories and weakening useful exploration.

3.5 Mathematical Reasoning Setting

To study verbal rejection sampling without the confounding effects of tool interaction, we evaluate it on CoT sub-trajectories in tool-free mathematical reasoning.

Trajectory-Space Approximation. Unlike Web Q&A, mathematical rollouts have no explicit markers for step boundaries. We use high-entropy token positions as a heuristic for potential branch points in a reasoning path. Accordingly, OVD-HES splits each rollout at the student's maximum-entropy token, retains the preceding prefix, and asks the teacher to resample the suffix if its score is below τ_{tr}. By contrast, OVD-FR resamples the entire low-scoring trajectory. We evaluate two training regimes across checkpoints from 300 to 800 steps: one fixed problem used throughout optimization and 128 distinct problems. The best HES and RLVR macro averages are 55.56% and 54.14% with one problem (both at step 800), and 57.40% and 55.65% with 128 problems (both at step 300). Compared with full-response replacement, OVD-HES achieves higher best-checkpoint accuracy in both training settings, suggesting that retaining student-generated prefixes can support effective learning while high-entropy token positions guide teacher suffix replacement. We next compare per-step and cumulative training time to assess the computational cost of this approach.

Training Efficiency. Figure 7 reports per-step training costs. As problem diversity grows, suffix resampling becomes faster than OVD-FR. With 128 problems, OVD-HES reduces mean per-step training time by 10.2% relative to OVD-FR. Over 800 training steps, OVD-HES and OVD-RS reduce cumulative training time by 7.4% and 30.5%, respectively, relative to OVD-FR.

Entropy Collapse. *Does policy optimization concentrate probability on a narrow set of reasoning paths, reducing exploration and solution coverage?* (Yue et al., 2025) We use P@16 as a behavioral indicator of this loss of exploration, not as a direct measure of token entropy. At 800 steps in Figure 6, OVD-HES exceeds OVD-FR and OVD-RS by 3.24 and 2.36 points, respectively. This advantage suggests that retaining student-generated prefixes, rather than replacing entire trajectories, mitigates trajectory-level entropy collapse as training progresses.

Findings 5: OVD-*HES mitigates trajectory-level entropy collapse by preserving student exploration in the prefix space while repairing weak continuations.*

4 Conclusion

We introduced *On-policy Verbal Distillation* (OVD), a black-box-compatible framework using verbal scores for selective teacher replacement of student sub-trajectories and GRPO for mixed-trajectory training, without dense teacher logits. Under score-calibration and replacement-quality assumptions, we bound accepted and mixed distributions' deviation from a teacher-preferred target. Experiments on agentic Web Q&A and tool-free mathematics show gains over evaluated baselines, while high-entropy-suffix resampling preserves late-stage solution coverage.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting and revision, clarification of theoretical assumptions and claims, interpretation and presentation of experimental results, checking summary calculations, and LaTeX formatting and template migration. The authors reviewed and revised the AI-assisted material and take full responsibility for the final content of this work, including its text, claims, results, code, and other artifacts.

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 *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

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. pp. 12248–12267, 2024.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. Rest meets react: Self-improvement for multi-step reasoning llm agent. *arXiv preprint arXiv:2312.10003*, 2023.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International conference on artificial intelligence and statistics*, pp. 4447–4455. PMLR, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Wesley Chung, Sina Ghiassian, Somjit Nath, and Martha White. Rejection sampling for off-policy learning. In *NeurIPS 2018 Workshop on Continual Learning*, 2018.

Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*, 2025.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. Towards general agentic intelligence via environment scaling. pp. 17610–17621, 2026.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pp. 10421–10430. PMLR, 2023.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. 2024:32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. Todi: Token-wise distillation via fine-grained divergence control. *arXiv preprint arXiv:2505.16297*, 2025.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1152–1157, 2016.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*, 2020.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

Kevin Lu and Thinking Machines Lab. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation, 2025.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL. https://pretty-radio-b75.notion.site/19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: short papers)*, pp. 1773–1781, 2023.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pp. 9025–9049, 2024.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 975–984, 2020.

Lujie Niu, Haochen Sun, Fangkun Zhao, Sheng Chen, Zimeng Bai, Jiawei Zhang, Caixia Yuan, and Xiaojie Wang. Cotd-po: Chain-of-thought distillation with preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19975–19986, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. Importance resampling for off-policy prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

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, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Giacomo Spigler. Proximal policy distillation. *arXiv preprint arXiv:2407.15134*, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents, 2026.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. F-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*, 2023.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 4602–4625, 2022.

Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Krähenbühl. Cut your losses in large-vocabulary language models. *arXiv preprint arXiv:2411.09009*, 2024.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. Ad-kd: Attribution-driven knowledge distillation for language model compression. *arXiv preprint arXiv:2305.10010*, 2023.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/537d5aa768c2d534016a4d06f87bc8fb-Abstract-Conference.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21805–21830, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1106. URL https://aclanthology.org/2025.emnlp-main.1106/.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

A.1 Comparison of Distillation Methods

Table 4 situates OVD next to common distillation templates discussed in §1.1. Sequence-level SeqKD aligns whole sampled trajectories without per-step logits, typically using rollouts that the teacher has sampled in advance (offline target trajectories); token-level variants supervise every decoding step, either with full-vocabulary logits (the OPD reverse-KL setting of Agarwal et al. (2024)) or with a truncated top-k support that trades accuracy for the memory scaling in Figure 1b. OVD instead scores coarse steps with verbal judgments, avoiding dense vocabulary tensors while keeping on-policy rollouts under GRPO.

Table 4: Comparison of language-model distillation methods. Logit-based methods use either the full vocabulary or the top-k tokens.

Method	Granularity	Feedback	On-Policy	Mem.
SeqKD	Sequence	Samples	✗	Low
Off-Policy KD (full V)	Token	Full-vocab. logits	✗	High
Off-Policy KD (top-k)	Token	Top-k logits	✗	Med.
On-Policy KD (full V)	Token	Full-vocab. logits	✓	High
On-Policy KD (top-k)	Token	Top-k logits	✓	Med.
OVD (Ours)	Step	Verbal	✓	Med.

In the same reading, **Granularity** marks whether feedback attaches to an entire sequence, to each token, or to step-sized segments s_t; **Feedback** distinguishes target samples, full-V or top-k logits, and discrete verbal scores; **On-Policy** flags methods that optimize on trajectories drawn from the student (checkmarks), including token-level OPD and OVD, as opposed to off-policy KD where behavior and learning policies need not coincide; and **Mem.** is a qualitative footprint where full-V matching is highest, top-k and verbal step scoring sit in the middle ground between that cost and low-footprint sequence KD. This layout is qualitatively aligned with the OPD* rows (full-V reverse KL vs. top-k logits) and the OVD configurations with $(\tau_{\mathrm{tr}}, \tau_{\mathrm{te}})$ in Table 1; Table 4 should nevertheless be read as a coarse qualitative summary of memory efficiency and on-policy training across method families, not as a separate empirical claim.

A.2 Theoretical Analysis

We analyze how verbal rejection sampling constructs the training distribution and derive a conditional approximation bound in terms of acceptance calibration and teacher-replacement error. Section A.4 discusses normalization, the scope of the bound, and empirical score-scale comparisons.

Conditions for Exact Matching of the Accepted Component. In Eq. (6), the proportionality constant must be positive and independent of y for fixed q and τ, with acceptance probabilities bounded by one. The student must assign nonzero probability to every trajectory with positive probability under $p_T(\cdot \mid q)$. Under these conditions, substituting the acceptance rule into Eq. (4) and normalizing yields $p_{\mathrm{a},\tau}(y \mid q) = p_T(y \mid q)$. This result concerns the accepted student component; the mixed distribution in Eq. (5) also depends on teacher replacements.

Assumption A.1 (Approximate Density-Ratio Condition)**.** Fix a problem q, trajectory prefix $y_{<t}$, and threshold τ, and suppress q in the following local conditionals. Let $p_T(\cdot \mid y_{<t})$ be a normalized target specified independently of the acceptance rule, with support contained in that of $\pi_S(\cdot \mid y_{<t})$. There exists a constant $c > 0$, independent of s_t, such that $c\, p_T(s_t \mid y_{<t})/\pi_S(s_t \mid y_{<t}) \in [0, 1]$ on the student support and

$$\mathbb{E}_{s_t \sim \pi_S(\cdot \mid y_{<t})}\left[\left| a_\tau(s_t \mid y_{<t}) - c\,\frac{p_T(s_t \mid y_{<t})}{\pi_S(s_t \mid y_{<t})}\right|\right] \le \varepsilon, \tag{12}$$

where $\varepsilon \ge 0$ measures the mean absolute error in acceptance probabilities and $Z_\tau(y_{<t}) > 0$.

The empirical diagnostic in Figure 4 of the main text examines how verbal acceptance varies with a teacher-generation density-ratio proxy. The following proposition states the separate assumptions required for a distribution-approximation guarantee.

Proposition A.2 (Approximation of the Accepted and Mixed Distributions)**.** *Under Assumption A.1, the accepted student distribution at prefix $y_{<t}$ satisfies*

$$\mathrm{TV}(p_{\mathrm{a},\tau}(\cdot \mid y_{<t}), p_T(\cdot \mid y_{<t})) \le \min\left\{1, \frac{\varepsilon}{Z_\tau(y_{<t})}\right\}. \tag{13}$$

If the teacher replacement distribution satisfies $\mathrm{TV}(r_{\mathcal{E},\tau}(\cdot \mid y_{<t}), p_T(\cdot \mid y_{<t})) \le \eta$*, then*

$$\mathrm{TV}(p_{\mathrm{m},\tau}(\cdot \mid y_{<t}), p_T(\cdot \mid y_{<t})) \le \min\{1, \varepsilon + (1 - Z_\tau(y_{<t}))\eta\}. \tag{14}$$

Here $\mathrm{TV}(p, r) = \frac{1}{2}\sum_{s_t} |p(s_t) - r(s_t)|$ *is total variation distance. When* $Z_\tau(y_{<t}) = 1$*, the replacement term vanishes and no condition on teacher replacements is needed.*

Proof. Suppress the fixed conditioning on $y_{<t}$ and write $Z = Z_\tau(y_{<t})$ and $u(s_t) = \pi_S(s_t \mid y_{<t})a_\tau(s_t \mid y_{<t})$. Assumption A.1 gives $\|u - cp_T\|_1 \le \varepsilon$, hence $|Z - c| \le \varepsilon$. Since $p_{\mathrm{a},\tau} = u/Z$,

$$\mathrm{TV}(p_{\mathrm{a},\tau}, p_T) = \frac{\|u - Zp_T\|_1}{2Z} \le \frac{\|u - cp_T\|_1 + |c - Z|}{2Z} \le \frac{\varepsilon}{Z}.$$

Equation (5) gives $p_{\mathrm{m},\tau} = Zp_{\mathrm{a},\tau} + (1 - Z)r_{\mathcal{E},\tau}$. Convexity of total variation therefore yields

$$\mathrm{TV}(p_{\mathrm{m},\tau}, p_T) \le Z\,\mathrm{TV}(p_{\mathrm{a},\tau}, p_T) + (1 - Z)\,\mathrm{TV}(r_{\mathcal{E},\tau}, p_T) \le \varepsilon + (1 - Z)\eta.$$

Total variation is at most one, giving the stated bounds. □

The condition quantifies proportionality of acceptance probabilities to a density ratio, which ranking accuracy alone does not establish. The bounds apply to each fixed prefix: the accepted distribution has error at most $\varepsilon/Z_\tau(y_{<t})$, while the bound for the mixed distribution also accounts for teacher replacement error. Setting $\varepsilon = 0$ recovers exact matching of the accepted component; if the replacement error is also zero, the mixed component matches $p_T(\cdot \mid y_{<t})$.

Proposition A.3 (Approximation of Complete Mixed Trajectories)**.** *Fix a problem* q *and a finite horizon* T *of sub-trajectory transitions. Use a common segment representation and termination rule for the mixed and target processes, padding terminated trajectories with an absorbing symbol. Suppress* q*, and let*

$$P_{\mathrm{m}}(y) = \prod_{t=1}^{T} p_{\mathrm{m},t}(s_t \mid y_{<t}), \qquad P_T(y) = \prod_{t=1}^{T} p_{T,t}(s_t \mid y_{<t}).$$

The target conditionals are specified independently of acceptance and jointly define one target process. Suppose that, for each t*, there is a bound* $\delta_t \in [0, 1]$ *such that*

$$\mathrm{TV}(p_{\mathrm{m},t}(\cdot \mid h), p_{T,t}(\cdot \mid h)) \le \delta_t$$

for every prefix h *reachable under either process. Then*

$$\mathrm{TV}(P_{\mathrm{m}}, P_T) \le 1 - \prod_{t=1}^{T}(1 - \delta_t) \le \min\left\{1, \sum_{t=1}^{T}\delta_t\right\}. \tag{15}$$

In particular, if Proposition A.2 applies at every such nonterminal prefix with local errors $\varepsilon_t(h)$*,* $\eta_t(h)$ *and acceptance rate* $Z_t(h) > 0$*, one may take*

$$\delta_t = \sup_h \min\{1, \varepsilon_t(h) + (1 - Z_t(h))\eta_t(h)\},$$

with zero local error at absorbed prefixes. At a prefix with $Z_t(h) = 0$*, the local bound instead comes directly from the replacement distribution.*

Proof. Couple the two processes sequentially. Whenever their prefixes agree, use a maximal coupling of their next-segment conditionals, whose probability of disagreement is their total variation distance and hence at most δ_t. Once their prefixes differ, continue with any coupling preserving the respective conditionals. If E_t is the event that the prefixes agree through step t, then

$$\Pr(E_t) \ge (1 - \delta_t)\Pr(E_{t-1}), \qquad \Pr(E_T) \ge \prod_{t=1}^{T}(1 - \delta_t).$$

The coupling inequality gives $\mathrm{TV}(P_{\mathrm{m}}, P_T) \le \Pr(E_T^c)$. Finally, $1 - \prod_t(1 - \delta_t) \le \sum_t \delta_t$ and total variation is at most one. Absorbing transitions agree deterministically after joint termination, so padding introduces no additional error. □

Consequently, for any common final-answer success indicator $f(y) \in \{0, 1\}$, the difference $|\mathbb{E}_{P_{\mathrm{m}}} f - \mathbb{E}_{P_T} f|$ is bounded by the right-hand side of Eq. (15). The bound concerns the trajectory-construction process at a fixed student policy. It applies to complete trajectories when termination occurs by T, or to capped trajectories when a generation limit is imposed.

Discussion. The target is specified independently of verbal scores. Ranking accuracy alone does not imply the density-ratio condition, and teacher resampling alone does not guarantee a small replacement error. Proposition A.3 extends local approximation to complete mixed trajectories under uniform prefix-wise bounds and a common sequential representation. It does not establish convergence of the trained student to the target or a reduction in gradient variance. The mixed distribution constructs training data, while the student is updated using the clipped objective defined in the main text.

A.2.1 Empirical Diagnostic of Verbal Acceptance

Figure 8 examines the relationship between verbal acceptance and the teacher-to-student log density ratio on 6,400 base-student answers to 200 Math500 questions, with 32 samples per question. We use the teacher generation policy $\pi_{\mathcal{E}}$ as an observable proxy for the teacher-preferred target p_T. Both models score the same answer text using their respective tokenizers and prompt templates; terminal stop tokens are excluded, and outputs reaching the generation cap are scored as prefixes. After within-question centering, we group answers into ten equal-count bins and estimate the fraction accepted at each threshold, with 95% question-bootstrap intervals.

Acceptance generally rises from the lowest-ratio bins and then levels off, with a similar broad trend in the length-normalized diagnostic. Raising the threshold from $\tau = 3$ or 5 to 7 lowers acceptance substantially; the highest length-normalized bin also shows a decline at $\tau = 7$. These patterns are consistent with verbal filtering favoring answers with higher relative teacher likelihood, but do not establish the proportionality condition in Eq. (6). In particular, $\pi_{\mathcal{E}}$ is only a proxy for p_T, and a binned association does not demonstrate trajectory-wise calibration or bound the replacement error.

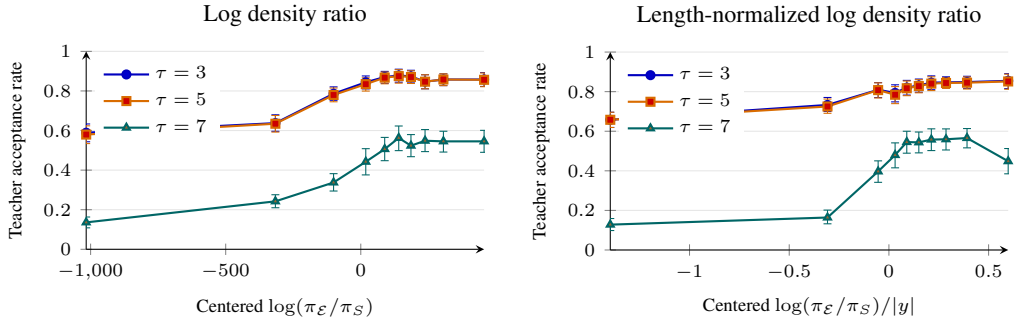


Figure 8: Verbal acceptance $a_\tau(y \mid q) = \Pr[S(y,q) \geq \tau]$ versus teacher-to-student log density ratios for 6,400 base-student answers (200 Math500 questions $\times$ 32 samples). Left: raw; right: length-normalized. Points show acceptance rates in ten equal-count bins after within-question centering; bars indicate 95% question-bootstrap intervals. Teacher generation policy $\pi_{\mathcal{E}}$ is a proxy for p_T, not its definition.

A.3 Web Q&A Rollout Format

Web Q&A serialization. A simplified rollout is:

The closing markers give explicit segmentation points for query, returned-information, and reasoning spans; these define the coarse sub-trajectories for verbal scoring. The `<information>` span is inserted by the search environment and is masked during policy optimization.

Example Web Q&A rollout

Question: What city hosted the 2012 Summer Olympics?
`<think>` Need the host. `</think>`
`<search>` 2012 Olympics host `</search>`
`<information>` London. `</information>`
`<think>` The host is London. `</think>`
`<answer>` London `</answer>`

Table 5: Effect of the teacher's score range on one-problem training with full-response resampling. Scores are accuracy (%), evaluated on AMD GPUs. Avg. 10 and Avg. 23 average the ten displayed benchmarks and all 23 benchmarks, respectively.

Steps	SVAMP	ASDiv	MAWPS	TABMWP	Minerva	Olympiad	SAT-Math	MMLU	Gaokao	AIME24	Avg. 10	Avg. 23
Score scale 0–5, reject if score < 4												
300	89.90	92.28	95.98	81.80	28.68	30.52	75.00	52.02	30.77	10.00	58.69	56.51
500	90.30	91.96	96.22	81.50	28.31	28.89	71.88	51.16	32.97	16.67	58.99	56.95
600	89.90	91.74	96.13	81.70	30.15	29.63	78.12	52.05	30.77	16.67	59.69	57.78
800	89.50	92.01	96.66	82.80	27.57	28.59	81.25	51.86	36.26	13.33	59.98	57.55
Score scale 0–9 (baseline), reject if score < 7												
300	90.20	92.14	96.27	80.10	22.06	30.81	87.50	54.04	35.16	13.33	60.16	56.33
500	90.20	91.83	96.80	82.30	26.84	31.11	78.12	54.41	31.87	16.67	60.01	56.92
600	90.40	91.83	96.76	82.70	26.47	30.37	78.12	53.98	32.97	13.33	59.69	56.96
800	90.00	92.42	96.42	83.60	25.74	31.56	75.00	53.08	29.67	10.00	58.75	57.81
Score scale 0–20, reject if score < 16												
300	90.60	91.83	96.22	80.80	26.10	31.11	81.25	53.25	32.97	6.67	59.08	54.82
500	90.10	91.83	96.13	81.10	28.31	29.48	78.12	53.61	35.16	13.33	59.72	57.38
600	89.90	92.05	96.27	81.30	26.10	30.07	78.12	52.98	34.07	10.00	59.09	56.62
800	90.00	91.92	96.42	81.20	27.94	30.52	75.00	51.56	35.16	16.67	59.64	56.75

All rows use actor-only evaluation with temperature 0 and top-p 1.0. The 0–5 and 0–9 settings use single-token logprob scoring; 0–20 uses constrained integer output.

A.4 Additional Analysis and Score Granularity

Normalization of the Training Distribution. Fix a problem and a student prefix, and suppress this conditioning. Let $a_\tau(s)$ be the probability of accepting a candidate sub-trajectory s, including any randomness in score decoding, and let $Z_\tau = \sum_s \pi_S(s)a_\tau(s)$. For $Z_\tau > 0$, the accepted distribution is $p_{\mathrm{a},\tau}(s) = \pi_S(s)a_\tau(s)/Z_\tau$. Accepting the student candidate or replacing it after rejection gives

$$p_{\mathrm{m},\tau}(s) = \pi_S(s)a_\tau(s) + (1 - Z_\tau)r_{\mathcal{E},\tau}(s) = Z_\tau p_{\mathrm{a},\tau}(s) + (1 - Z_\tau)r_{\mathcal{E},\tau}(s). \tag{16}$$

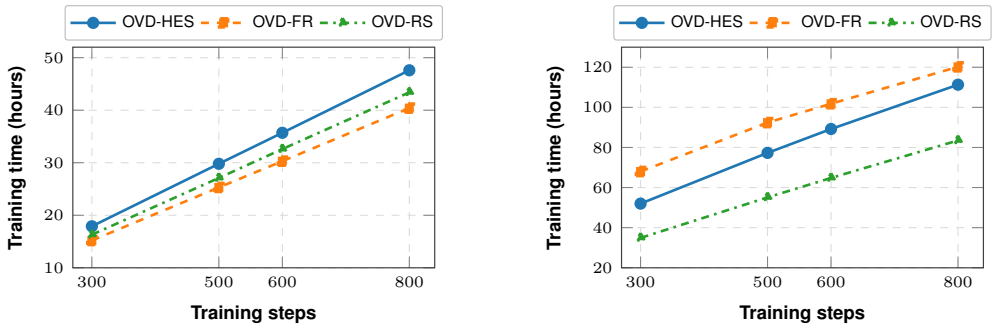
Here $r_{\mathcal{E},\tau}$ is the normalized replacement distribution conditional on rejection at the fixed prefix, averaging over rejected candidates if replacements depend on them. The two components have masses Z_τ and $1 - Z_\tau$, so their sum is normalized. When $Z_\tau = 0$, the mixed distribution consists entirely of replacements; when $Z_\tau = 1$, it equals the student distribution and no replacement distribution is needed.

Distribution Approximation and Policy Optimization. Proposition A.2 bounds the next sub-trajectory distribution at a fixed prefix. Proposition A.3 extends this result to complete mixed trajectories when the local errors are controlled at every reachable prefix under a common sequential representation. The accumulated bound can become uninformative for long horizons unless local errors remain small. Approximation of the training distribution does not imply that policy optimization converges to it. In particular, the policy-gradient identity for π_S cannot be applied directly to expected rewards under a teacher–student mixture. The clipped training objective in the main text is a surrogate evaluated on collected mixed trajectories; these distribution bounds do not establish unbiased gradients for the student's expected return, lower gradient variance, or a convergence rate.

A.4.1 Score Granularity: Mathematical-Reasoning Empirics

We examine the score vocabulary empirically using the same rejection-sampling interface. The 0–9 scale is the default in the main experiments.

Training Cost. Figure 9 reports cumulative wall-clock time across training checkpoints, complementing the per-step measurements in Figure 7.



(a) One fixed training problem.

(b) 128 distinct training problems.

Figure 9: Cumulative wall-clock training time under different resampling strategies. At 800 steps, OVD-HES and OVD-RS take 17.9% and 7.5% more time than OVD-FR with one fixed problem, but 7.4% and 30.5% less time with 128 problems. Panels use different vertical scales.

Table 5 compares three score ranges for mathematical reasoning with full-trajectory resampling. Coarse verbal scores can provide an effective selection signal: at 800 steps, the 0–5 scale outperforms the default 0–9 scale on five benchmarks, with particularly large gains on Gaokao and SAT-Math. Figure 10 tracks average accuracy across training steps on these five benchmarks, selected by their gains over the default at this checkpoint. Their average improvement is 3.65 percentage points. Across all 23 benchmarks, however, the default reaches 57.81%, compared with 57.55% for 0–5 and 56.75% for 0–20. Thus, expanding the score range does not consistently improve performance. The 0–20 setting also changes the scoring format, so this comparison does not isolate score granularity alone.

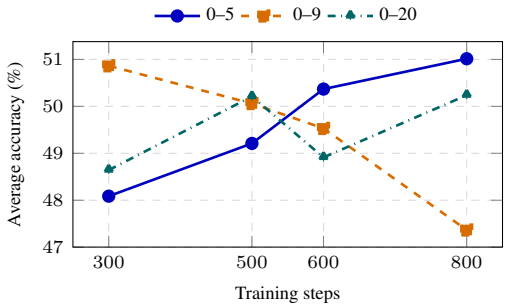


Figure 10: Average accuracy on Gaokao, SAT-Math, AIME24, Minerva, and MAWPS across training checkpoints. These five benchmarks were selected from the ten displayed in Table 5 by the largest gains of 0–5 over 0–9 at 800 steps; the same set is used for every curve and checkpoint. Rejection thresholds are 4, 7, and 16 for the 0–5, 0–9, and 0–20 scales, respectively. Complete results and scoring formats are reported in Table 5.

A.5 Additional Mathematical-Reasoning Results

Random-Suffix Resampling Control. Table 3 reports AMC23 alongside an eight-benchmark subset selected post hoc from 23 tasks, with at most one Gaokao benchmark, to maximize the smaller of the single-problem and 128-problem best-checkpoint HES gains over RLVR across steps 300, 500, 600, and 800. The resulting nine-benchmark set is fixed across all rows. AMC23 scores are restored from the earlier evaluation summary. The nine-benchmark averages are reconstructed as $(8A_8 + A_{\mathrm{AMC23}})/9$, where A_8 is the previously reported eight-benchmark average; their last decimal may be affected by rounding.

Table 6 reports OVD-RS after training on 128 distinct problems, on the same nine benchmarks and checkpoints as Table 3. At 800 steps, its macro average is 56.34%, compared with 53.77% for OVD-HES. The ordering varies across checkpoints, so these results do not establish a consistent macro-average advantage from selecting a high-entropy suffix over a random suffix.

Majority Voting and Solution Coverage. Table 7 reports results on all ten benchmarks after training on 128 distinct problems. M@k measures the accuracy of the majority-selected answer, whereas P@k measures whether the sampled set contains at least one correct answer. At 800 steps, OVD-HES reaches 80.4% average P@16, compared with 79.7% for OVD-FR and 80.1% for OVD-RS; their M@16 averages are 65.5%, 65.1%, and 65.3%, respectively. The P@16 advantage of

Table 6: Random-suffix resampling on 128 training problems. Benchmarks and averaging follow Table 3.

Method	Steps	AIME24	AMC23	CARP	Middle	College	Gaokao-II	MATH	MAWPS	SAT-Math	Avg.
Training on 128 Distinct Problems											
OVD-RS	300	16.7	47.5	52.8	44.6	35.8	42.9	68.2	93.0	81.3	53.62
OVD-RS	500	6.7	52.5	53.8	46.5	36.6	50.0	69.4	93.6	78.1	54.13
OVD-RS	600	16.7	60.0	54.4	44.6	38.0	50.0	70.7	95.8	71.9	55.78
OVD-RS	800	16.7	57.5	53.9	45.5	39.0	42.9	70.8	96.3	84.4	**56.34**

Table 7: Sampling results for the three resampling strategies after training on the fixed 128-problem DeepScaleR subset, with $(\tau_{\mathrm{tr}}, \tau_{\mathrm{te}}) = (7, 0)$. M@$k$ is majority-vote accuracy; tied votes count as incorrect. P@k counts a question as correct if any of the k answers is correct. We sample 16 answers per question and use a fixed-seed random subset for $k = 8$. Temperature is 0.6 and top-p is 1.0.

(a) Majority-vote accuracy

Steps	SVAMP		ASDiv		MAWPS		TABMWP		Minerva		Olympiad		SAT-Math		MMLU		Gaokao		AIME24		Avg.	
	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16	M@8	M@16
OVD-FR																						
300	91.0	91.6	93.0	93.8	97.0	97.6	88.5	89.8	32.7	34.9	36.6	39.4	81.2	90.6	53.5	56.5	36.3	40.7	20.0	26.7	63.0	66.1
500	92.2	92.1	93.4	94.0	97.1	97.2	89.2	90.5	34.6	35.3	34.4	37.2	87.5	90.6	55.8	58.2	35.2	46.2	20.0	20.0	63.9	66.1
600	92.0	93.0	93.2	93.7	97.0	97.4	88.6	90.2	35.3	38.2	34.4	36.6	84.4	84.4	54.7	57.7	39.6	49.5	20.0	23.3	63.9	66.4
800	91.7	92.0	93.1	93.2	97.0	97.1	87.9	89.9	36.0	37.5	33.0	35.4	84.4	81.2	54.8	57.3	42.9	44.0	23.3	23.3	64.4	65.1
OVD-HES																						
300	91.3	92.2	93.7	94.0	97.2	97.4	88.8	90.9	33.1	37.5	36.4	38.8	78.1	87.5	54.9	57.3	40.7	39.6	26.7	30.0	64.1	66.5
500	91.5	91.9	93.0	93.3	97.2	97.5	87.1	89.6	31.2	34.6	36.0	37.9	81.2	87.5	53.2	56.4	41.8	46.2	20.0	23.3	63.2	65.8
600	92.2	92.2	93.3	94.0	97.3	97.5	90.5	91.3	36.0	36.4	36.0	37.9	87.5	93.8	53.3	56.8	45.1	41.8	23.3	26.7	65.5	66.8
800	91.8	92.4	93.0	93.6	97.2	97.6	88.7	90.1	30.9	34.6	34.1	37.3	84.4	84.4	53.7	56.7	35.2	45.1	23.3	23.3	63.2	65.5
OVD-RS																						
300	91.4	92.0	93.0	93.8	97.4	97.7	88.4	89.5	31.2	36.8	36.3	38.2	90.6	93.8	54.3	57.1	36.3	40.7	20.0	23.3	63.9	66.3
500	92.2	92.9	93.6	94.0	97.1	97.6	88.8	90.6	32.7	34.2	34.8	36.0	84.4	90.6	54.8	56.8	38.5	41.8	23.3	26.7	64.0	66.1
600	91.7	92.3	93.8	94.3	97.2	97.7	89.6	90.4	29.0	33.1	36.6	37.5	84.4	81.2	54.4	57.5	40.7	40.7	26.7	23.3	64.4	64.8
800	92.4	92.7	93.4	93.9	97.6	97.5	87.4	89.4	28.3	32.0	35.4	36.4	84.4	90.6	54.4	56.8	36.3	40.7	20.0	23.3	63.0	65.3

(b) Pass@k

Steps	SVAMP		ASDiv		MAWPS		TABMWP		Minerva		Olympiad		SAT-Math		MMLU		Gaokao		AIME24		Avg.	
	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16	P@8	P@16
OVD-FR																						
300	96.5	97.6	96.9	97.5	98.6	98.8	94.9	95.9	49.3	55.5	50.9	56.4	96.6	96.9	89.7	95.6	70.7	76.9	33.1	40.0	77.7	81.1
500	96.1	96.9	96.8	97.4	98.6	98.7	95.4	96.6	49.3	55.5	50.8	56.6	98.2	100.0	89.0	94.8	70.4	74.7	32.3	36.7	77.7	80.8
600	96.3	97.2	96.7	97.2	98.5	98.7	95.8	96.8	53.1	60.3	50.0	56.0	95.4	96.9	88.7	94.9	70.4	74.7	35.7	43.3	78.1	81.6
800	95.9	96.7	96.5	97.0	98.5	98.7	95.0	96.1	49.9	54.4	49.2	54.5	95.2	100.0	87.9	94.1	70.1	75.8	27.9	30.0	76.6	79.7
OVD-HES																						
300	96.1	97.2	96.9	97.2	98.7	98.9	95.5	96.8	53.1	59.2	52.9	59.6	96.0	96.9	89.3	95.2	71.0	76.9	32.7	36.7	78.2	81.5
500	96.2	97.3	96.9	97.6	98.7	98.9	95.1	96.5	47.0	52.6	50.9	56.4	98.8	100.0	89.2	95.3	69.8	75.8	32.2	36.7	77.5	80.7
600	96.0	96.9	96.9	97.4	98.7	98.9	95.7	96.6	52.1	56.6	51.2	56.9	96.8	96.9	89.1	95.1	66.1	71.4	34.1	40.0	77.7	80.7
800	96.1	97.1	96.8	97.3	98.7	98.9	95.4	96.4	44.9	50.7	51.1	56.6	95.2	96.9	88.7	94.7	67.0	72.5	37.4	43.3	77.1	80.4
OVD-RS																						
300	95.9	96.7	97.0	97.6	98.6	98.8	95.3	96.4	49.8	54.8	52.0	57.5	96.8	96.9	89.1	94.9	70.4	75.8	33.3	40.0	77.8	80.9
500	96.1	97.2	97.0	97.5	98.6	98.8	95.6	96.9	47.5	53.7	50.4	56.1	98.3	100.0	89.6	95.3	68.4	75.8	31.4	36.7	77.3	80.8
600	96.7	97.8	97.2	97.7	98.8	98.9	95.2	96.3	43.8	49.3	49.9	54.7	98.3	100.0	89.0	94.9	68.3	73.6	31.6	33.3	76.9	79.7
800	96.1	96.9	97.0	97.4	98.7	98.7	95.6	96.5	49.0	55.9	49.8	54.8	95.2	96.9	89.0	94.9	68.8	75.8	31.6	33.3	77.1	80.1

OVD-HES is therefore smaller across all ten benchmarks than on the five-benchmark subset shown in Figure 6. The ordering also varies across checkpoints: at 600 steps, OVD-FR has the highest average P@16 (81.6%). These results distinguish solution coverage from majority-vote accuracy and show that the benefit of preserving student prefixes depends on the benchmark and training checkpoint.

Table 8 reports the complete two-example scaling results omitted from the main-text comparison.

Table 8: Math accuracy after training on two problems. All methods use Qwen2.5-Math-1.5B and are evaluated without a teacher. OPD* denotes token-level distillation at 800 steps.

Method	Steps	SVAMP	ASDiv	MAWPS	TABMWP	Minerva	Olympiad	SAT-Math	MMLU	Gaokao	AIME24	Avg.
Scaling RLVR (2 random examples)												
Two-example	300	73.7	76.2	80.4	70.8	16.5	30.2	81.2	51.1	35.7	6.7	52.2
Two-example	500	77.1	80.0	86.0	75.17	16.5	27.7	81.2	51.3	42.9	13.3	55.1
Two-example	600	75.9	78.1	84.6	75.6	16.5	31.1	84.4	49.6	35.7	13.3	54.5
Two-example	800	81.1	80.0	87.6	73.7	15.8	29.6	81.2	49.0	35.7	13.3	54.7
OPD* (reverse KL)	800	80.5	79.3	87.0	73.2	15.6	29.3	80.8	48.9	35.4	13.0	53.8
OPD* (top-k logits)	800	81.5	80.3	88.0	74.0	16.0	29.8	81.5	49.2	36.0	13.6	55.1
OVD *(2 random examples, τ_{tr}=7, τ_{te}=0)*												
OVD	300	78.6	81.8	87.1	72.3	15.1	29.5	81.2	50.2	50.0	6.7	55.2
OVD	500	79.8	81.1	87.7	74.9	17.3	29.3	75.0	50.3	35.7	10.0	54.1
OVD	600	76.3	77.7	82.6	74.0	17.3	30.4	87.5	50.6	28.6	10.0	53.5
OVD	800	**83.1**	**83.3**	**90.6**	**77.6**	**18.0**	**31.6**	**84.4**	49.4	35.7	13.3	**56.7**

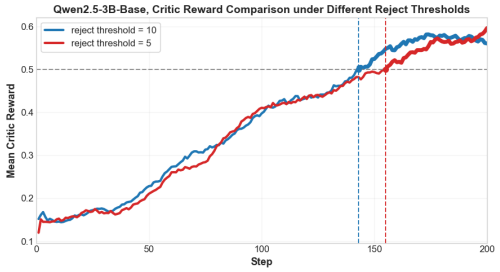


Figure 11: Reward convergence on Qwen-2.5-7B across training steps under different *training-time* thresholds τ_{tr}. Dashed horizontal line: reward 0.5; vertical markers: first 0.5 crossing for each τ_{tr} curve.

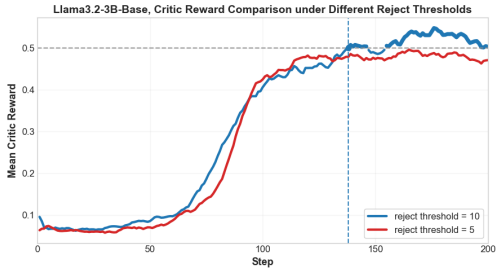


Figure 12: Reward convergence on LLaMA-3.2-3B across training steps under different *training-time* thresholds τ_{tr}. Dashed horizontal line: reward 0.5; vertical marker: τ_{tr}=10 first crossing 0.5.

A.6 Illustrative Mathematical-Reasoning Problems

The box accompanying Figure 13 shows two example problems illustrating the mathematical-reasoning training setting.

A.7 Reward Convergence Across Training Steps

We analyze the reward convergence behavior of OVD across different training steps while varying the *training-time* verbal rejection threshold τ_{tr} (the test-time gate τ_{te} used when reporting EM is fixed and not swept in these curves). Figure 11 and Figure 12 illustrate the reward trajectories for Qwen-2.5-7B and LLaMA-3.2-3B models respectively, demonstrating that appropriate training threshold settings lead to faster convergence and higher final rewards. The results show that τ_{tr} significantly impacts learning dynamics, with moderate values (5–7) typically achieving the best balance between exploration and exploitation.

A.8 Valid Search Across Training Steps

We report the *valid search* metric during Web Q&A training, which measures the average number of search interactions per episode that are considered valid by the simulated environment (higher values indicate more consistent and well-formed tool-use behavior).

Figure 13 compares OVD and a token-level OPD baseline (QueryLogitOPD, using top-k query logits for on-policy distillation) over the same training horizon.

Illustrative Mathematical-Reasoning Problems

Problem 1 : The lock opens only if a specific three-digit number is entered. An attempt consists of randomly selecting three digits from a given set of five. The code was guessed correctly only on the last of all attempts. How many attempts preceded the successful one? Let's think step by step and output the final answer within $\square$.

Answer: 124

Problem 2 : For the ellipse $25x^2 - 100x + 4y^2 + 8y + 16 = 0$, find the distance between the foci. Let's think step by step and output the final answer within $\square$.

Answer: $\frac{2\sqrt{462}}{5}$

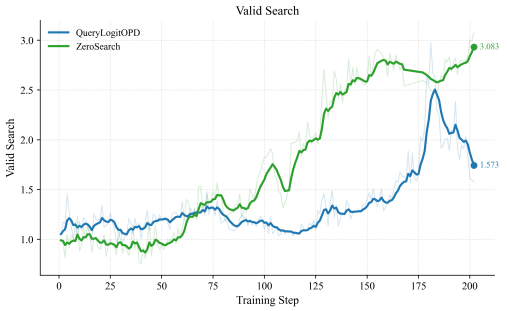


Figure 13: **Valid search count across training steps.** OVD (green curve) maintains higher valid-search rates throughout training compared to QueryLogitOPD (blue curve). The gap widens as training progresses, demonstrating that verbal step-level feedback helps the student policy learn more consistent search behaviors without requiring dense token-level logit matching at every query generation step.

A.9 Verbal Scoring Prompt

We present the detailed prompt template used for verbal scoring in OVD. The prompt instructs the teacher model to evaluate the quality of each reasoning step on a discrete 0–9 scale, providing structured feedback without requiring access to token-level probability distributions. This design enables step-level quality assessment while maintaining compatibility with black-box teacher models. The scoring rubric divides the scale into four meaningful ranges: incorrect/irrelevant (0–2), partially correct (3–5), correct and well-reasoned (6–8), and optimal (9), allowing the teacher to express nuanced judgments about reasoning quality. By outputting only a single score token, this approach dramatically reduces memory overhead compared to full-vocabulary logit storage, as analyzed in Section 1.1.

Verbal Scoring Prompt

Problem: q
Previous Steps: $s_1, s_2, \ldots, s_{t-1}$
Current Step: s_t
Rate the quality of the current reasoning step on a scale of 0–9, where:

- 0–2: Incorrect or irrelevant
- 3–5: Partially correct but incomplete
- 6–8: Correct and well-reasoned
- 9: Optimal step toward the solution

Output only a single number from 0 to 9.

A.10 Hyperparameters and Implementation Details

A.10.1 Implementation Details

We provide detailed hyperparameter configurations and implementation details to ensure the reproducibility of our experiments.

Computational Infrastructure. All experiments are conducted on a high-performance computing cluster equipped with 8 AMD Instinct MI210 accelerators, each featuring 64 GB of HBM2e memory. The GPUs are based on the CDNA 2 microarchitecture (gfx90a). Distributed training is performed using 4 GPUs per training instance. GPU acceleration is supported via the ROCm platform.

Teacher Models. OVD uses a task-specific teacher $\pi_{\mathcal{E}}$ to score student sub-trajectories and resample low-scoring ones through the verbal interface in Section 2.1. Teacher-side verbal scoring during training is served on NVIDIA A100 GPUs (40 GB) for low-latency inference alongside student rollouts. For *Web Q&A*, simulated search and scoring use `sunhaonlp/Simulation_LLM_google_7B_V2`; for *mathematical reasoning*, the critic is `deepseek-ai/DeepSeek-R1-Distill-Qwen-7B` (Guo et al., 2025), applied at each reasoning step in the same serving regime.

Baselines To evaluate the effectiveness of OVD, we compare our method with the following baselines. (i) *Vanilla Prompting Methods*: This category includes Direct Answer, Chain-of-Thought (CoT) (Wei et al., 2022), and standard Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). Direct Answer generates responses without explicit reasoning, CoT encourages step-by-step reasoning, and RAG retrieves relevant documents before answering. (ii) *Advanced Search Methods*: We consider RA-Agent and Search-o1 (Li et al., 2025), which iteratively search for relevant information and perform multi-step reasoning with retrieved context. (iii) *RL-Based Reasoning Methods*: This category includes R1 (Guo et al., 2025), Search-R1 (Jin et al., 2025), and ZeroSearch (Sun et al., 2025). In R1, the student policy is trained to perform in-depth reasoning based solely on its internal knowledge. Search-R1 enables the student policy to interact with a real search engine during inference. ZeroSearch learns search-like reasoning without actual retrieval through simulated search environments. (iv) *Additional search baselines (Qwen-2.5-3B-Base)*: We further include SmartSearch (Wen et al., 2026) and StepSearch (Zheng et al., 2025) in Table 1 under the same evaluation protocol.

A.10.2 Hyperparameters

OPD* top-k logit baseline. Token-level OPD* rows with *top-k logits* (Table 1 and Table 8) use the truncated support described in paragraph Memory Bottleneck: at each decoding step we retain only the teacher's k highest-probability tokens, restrict both the student and teacher distributions to that support, and renormalize them before computing the reverse-KL objective. We fix k=8192 for all reported top-k runs, which matches the top-k memory curves in Figure 1b under $V \approx 152$K (Qwen2.5-7B-class vocabulary).

We employ different hyperparameters for trajectory scoring depending on the task type.

Mathematical Reasoning. For math tasks, we use `deepseek-ai/DeepSeek-R1-Distill-Qwen-7B` (Guo et al., 2025) as the teacher model. The teacher evaluates reasoning trajectories by computing verbal scores on a 0–9 scale based on the prompt template described in Appendix A.9. By default, we greedily decode the highest-probability score token; score-vocabulary sampling variants use a temperature of 0.7. The student policies are initialized from pre-trained base models without instruction tuning, with Qwen2.5-Math-1.5B used for mathematical tasks.

We train the Generalized Reward Preference Optimization (GRPO) framework with carefully selected hyperparameters to ensure stable and effective optimization under long-horizon rollouts. We adopt Qwen2.5-Math-1.5B as the base student policy for mathematical reasoning tasks.

The training batch size is set to 128, with dynamic batch sizing enabled to accommodate variable-length responses. The maximum prompt length is 1,024 tokens, and the maximum response length is capped at 3,072 tokens, allowing sufficient capacity for long-form reasoning trajectories.

We use a constant learning rate of 1×10^{-6} for policy optimization. For each prompt, rollout is performed by sampling 8 candidate responses with a temperature of 0.6, and rollouts are generated using a vLLM-based inference engine with a tensor parallel size of 2. To stabilize training and prevent policy drift, we apply KL-divergence regularization at both the actor and algorithm levels, using a low-variance KL formulation with a coefficient of 0.001, together with an additional KL control term using the same coefficient.

Web Q&A. For web search tasks, we compare prompt-based and SFT-based teacher agents built from the same pretrained backbone. The prompt-based teacher uses the base model directly and relies on an instruction prompt to simulate search-engine outputs. The SFT-based teacher starts from the same base model and is further fine-tuned on simulated search data following ZeroSearch (Sun et al., 2025). The teacher backbone is Qwen2.5-7B-Base or Qwen2.5-14B-Base. Training-time

student rollouts and teacher scoring use these teacher simulators only; all reported Web Q&A test EM numbers use the Serper API as the sole *live* search tool at inference, aligned with Search-R1 and ZeroSearch evaluation. During training, each teacher provides step-level verbal scores evaluating both query quality and reasoning coherence. We set $\tau_{\text{tr}} = 7$ during teacher trajectory generation. For the 14B teacher rows in Appendix D (Table 12), we use the 14B-based teacher agent. For Web Q&A tasks, the student policies are initialized from the pre-trained base models Qwen2.5-3B-Base and LLaMA-3.2-3B-Base without instruction tuning.

Prompt-Based Search Simulation Example. The illustrative example below shows how an instruction prompt turns a search query into a simulated result. It is not the verbatim experimental prompt or a recorded model output.

Illustrative prompt and simulated search output

Instruction prompt:
Simulate a search engine. Given a search query, return a concise result with a title and a factual snippet relevant to the query. Generate the result without accessing a live search engine.

Search query:
2012 Summer Olympics host city

Simulated search output:
Title: London 2012 Summer Olympics
Snippet: London, United Kingdom, hosted the 2012 Summer Olympics from 27 July to 12 August 2012.

We train the Generalized Reward Preference Optimization (GRPO) framework with carefully selected hyperparameters to ensure stable and effective optimization. We adopt Llama-3.2-3B (`meta-llama/Llama-3.2-3B`) and Qwen-2.5-3B (`Qwen/Qwen-2.5-3B`) as the base student policies. Training is conducted for a total of 203 optimization steps.

The training batch size is set to 64, while the validation batch size is 512. The maximum prompt length is 4,096 tokens, and the maximum response length is capped at 500 tokens, allowing sufficient capacity for both mathematical reasoning and web-based question answering tasks.

We use a learning rate of 1×10^{-6} with a linear warmup schedule covering 95% of the total training steps. Policy optimization is performed within the GRPO framework, using a mini-batch size of 32 and a micro-batch size of 16. To stabilize training and prevent policy drift, we apply KL-divergence regularization with a coefficient of 0.001, using a low-variance KL formulation.

B Web Q&A Exact Match on LLaMA-3.2-3B-Base

This section reports the companion Web Q&A benchmark sheet for the LLaMA-3.2-3B-Base student. We use the same eight datasets, exact-match metric, column abbreviations, and $(\tau_{\text{tr}}, \tau_{\text{te}})$ notation as Table 1 in the main paper (Triv.=TriviaQA, Hot.=HotpotQA; 2W, Mus., Bam., GAI as there). The table lists search baselines (Search-o1, Search-R1, ZeroSearch) and OVD variants under different $(\tau_{\text{tr}}, \tau_{\text{te}})$ settings; OVD† denotes score-vocabulary sampling at test time, consistent with the main table. OVD markedly outperforms these search baselines on average exact match, mirroring the large margins observed for the Qwen-2.5-3B-Base student in Table 1.

Table 9: Web Q&A exact match (%) with LLaMA-3.2-3B-Base as the student. Benchmark abbreviations and threshold settings follow Table 1.

LLaMA-3.2-3B-Base									
Method	**NQ**	**Triv.**	**Pop**	**Hot.**	**2W**	**Mus.**	**Bam.**	**GAI**	**Avg.**
Search-o1	24.20	48.40	8.80	19.40	17.40	6.00	32.00	1.20	19.68
Search-R1	41.20	60.00	44.00	29.60	31.60	13.60	19.44	4.69	30.52
ZeroSearch	23.80	41.00	18.00	22.00	26.80	4.80	11.10	3.03	18.69
OVD (τ_{tr}=10, τ_{te}=0)	39.40	57.40	48.20	31.40	31.00	14.00	15.20	7.27	30.58
OVD (τ_{tr}=10, τ_{te}=5)	43.40	62.20	54.80	31.00	29.20	18.40	25.60	7.88	34.06
OVD (τ_{tr}=10, τ_{te}=10)	41.60	57.80	33.00	34.00	21.00	**34.40**	21.00	7.27	31.01
OVD (τ_{tr}=5, τ_{te}=0)	44.80	**64.80**	54.60	33.60	33.60	18.80	22.40	**9.09**	35.19
OVD (τ_{tr}=5, τ_{te}=5)	**46.80**	64.20	54.80	**37.00**	**36.20**	22.20	27.20	8.48	37.01
OVD (τ_{tr}=5, τ_{te}=10)	45.60	64.00	**57.20**	35.20	35.60	18.00	**32.80**	7.88	**37.04**
OVD† (τ_{tr}=5, τ_{te}=5)	45.80	64.40	54.80	34.20	**36.20**	22.20	28.00	7.27	36.51
OVD† (τ_{tr}=10, τ_{te}=5)	42.00	57.80	54.40	31.00	34.40	17.20	28.00	8.48	34.16

C Test-Time Threshold Ablation

Table 10 reports the complete Web Q&A results for $\tau_{\text{te}} \in \{0, 5, 10\}$ with $\tau_{\text{tr}} = 5$ fixed.

Table 10: Effect of the test-time rejection threshold on Web Q&A exact match (%). The student is Qwen-2.5-3B-Base, the teacher is Qwen2.5-7B, and $\tau_{\text{tr}} = 5$. Best results are in bold.

	Single-Hop QA			Multi-Hop QA					
τ_{te}	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	GAIA	Avg.
0	**49.20**	67.20	63.20	**39.20**	35.20	21.80	15.20	9.09	37.51
5	47.00	**67.40**	**66.40**	38.00	37.60	23.20	24.80	7.88	39.04
10	46.00	66.80	64.80	38.80	**43.80**	**25.00**	**40.00**	**10.30**	**41.94**

Table 11: Greedy versus sampled teacher scores on Web Q&A, with Qwen-2.5-3B-Base as student and Qwen2.5-7B as teacher. Greedy decoding takes the most likely score; sampling uses temperature 0.7 over the ten score tokens. We report exact match (%) at $\tau_{\text{te}} = 5$. Best results are in bold.

Score Decoding	τ_{tr}	**NQ**	**TriviaQA**	**PopQA**	**HotpotQA**	**2Wiki**	**Musique**	**Bamboogle**	**GAIA**	**Avg.**
Greedy	5	47.00	67.40	66.40	38.00	37.60	23.20	24.80	7.88	39.04
Sampling	5	**48.80**	**68.00**	**66.60**	**39.40**	**41.40**	**24.20**	**28.80**	11.50	**41.09**
Sampling	10	25.60	39.40	43.20	22.20	15.80	10.80	12.80	**12.72**	22.82

Table 11 compares greedy score decoding with score-vocabulary sampling and includes the sampling result at the higher training threshold $\tau_{\text{tr}} = 10$.

D SFT Teacher Scale

Table 12 reports per-dataset EM and F1 for the 7B vs. 14B SFT teacher ablation. Bold EM entries match the strongest 7B-teacher configuration from Table 1; F1 includes estimated values where noted in the caption.

We hold the student at Qwen-2.5-3B-Base and compare SFT search teachers built on Qwen2.5-7B-Base versus Qwen2.5-14B-Base, with $\tau_{\text{tr}} = 10$ and $\tau_{\text{te}} \in \{0, 5, 10\}$. Macro-average EM is higher with the 7B teacher than the 14B teacher at each τ_{te} under this schedule (40.97% vs. 38.41% at $\tau_{\text{te}} = 0$; 42.79% vs. 39.59% at 5; 43.96% vs. 41.09% at 10). The pattern is not explained by a single outlier dataset: the 7B teacher leads on GAIA and on several multi-hop benchmarks at comparable thresholds, even though a larger backbone might be expected to yield stricter or more informative verbal judgments. We therefore treat teacher scale as a knob that must be validated rather than assumed monotonic when rejection schedules are held fixed.

E Related Work

This section reviews reinforcement learning methods for improving language models, covering preference optimization, inducing reasoning abilities, and agent training.

E.1 Post-training Methods for Language Models

Proximal Policy Optimization. Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017) has become a standard approach for aligning LLMs with human preferences. Stiennon et al. (2020) first demonstrated the effectiveness of training reward models from human feedback for open-ended language generation tasks, which was later scaled to instruction-following settings by Ouyang et al. (2022). The RLHF paradigm typically involves training a reward model on human preference data and then fine-tuning the LLM using policy gradient methods, among which Proximal Policy Optimization (PPO) (Schulman et al., 2017) has been most widely adopted due to its stability and sample efficiency. Ziegler et al. (2019) first applied PPO to fine-tune language models from human preferences, establishing the foundation for modern RLHF pipelines. More recently, Ahmadian et al. (2024) revisited vanilla policy gradient methods, showing that they can achieve competitive performance while reducing implementation complexity. Despite its effectiveness, PPO-

Table 12: Web Q&A results with 7B and 14B SFT teachers and a Qwen-2.5-3B-Base student. Training uses $\tau_{\text{tr}} = 10$; test thresholds are 0, 5, and 10. Scores are EM and F1 (%). Unavailable F1 scores are estimated from server logs; GAIA F1 is extrapolated from a fitted EM-to-F1 trend. Bold EM scores mark the best 7B-teacher setting for each benchmark and Avg.

Teacher	τ_{te}	NQ		TriviaQA		PopQA		HotpotQA		2Wiki		Musique		Bamboogle		GAIA		Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SFT teacher: Qwen2.5-7B-Base																			
7B	0	**51.00**	53.89	68.80	73.57	69.00	56.58	39.40	46.69	42.00	46.42	21.00	27.58	23.20	35.59	**13.33**	25.56	40.97	45.73
7B	5	50.00	52.89	67.80	72.57	**69.80**	57.38	39.00	46.29	43.70	48.12	**27.90**	34.48	32.00	44.39	12.12	24.69	42.79	47.60
7B	10	50.00	48.88	**69.20**	70.81	65.20	45.33	**42.00**	48.04	**45.00**	46.07	26.40	33.50	**42.40**	48.42	11.51	24.26	**43.96**	45.66
SFT teacher: Qwen2.5-14B-Base																			
14B	0	48.60	51.49	66.40	71.17	64.80	52.38	38.40	45.69	38.60	43.02	22.80	29.38	16.80	29.19	10.90	23.82	38.41	43.27
14B	5	47.40	50.29	65.80	70.57	66.40	53.98	37.00	44.29	43.60	48.02	23.20	29.78	24.80	37.19	8.50	22.10	39.59	44.53
14B	10	48.00	50.89	66.60	71.37	65.40	52.98	39.00	46.29	41.80	46.22	24.00	30.58	33.60	45.99	10.30	23.39	41.09	45.96

based training requires maintaining multiple models and is computationally expensive, often requiring thousands of GPU hours.

Direct Preference Optimization. Direct Preference Optimization (DPO) (Rafailov et al., 2023) reformulates the RLHF objective to directly optimize the policy without explicit reward modeling, significantly simplifying the training pipeline. Subsequent work has proposed various improvements: IPO (Azar et al., 2024) provides a general theoretical framework addressing overfitting issues in DPO, ORPO (Hong et al., 2024) eliminates the need for a reference model through odds ratio-based optimization, and SimPO (Meng et al., 2024) further simplifies the objective with a reference-free reward formulation.

Reasoning via Reinforcement Learning. Recent work demonstrates that reinforcement learning (RL) and environment interaction provide effective supervision for enhancing reasoning and agentic capabilities of LLMs. Existing methods can be broadly categorized by their supervision source: (1) *Self-Training RL* bootstraps reasoning from model-generated rationales, including iterative refinement in STaR (Zelikman et al., 2022), generate-filter-finetune iterations for math and coding in ReST-EM (Singh et al., 2023), and ReST-style self-improvement of ReAct agents (Aksitov et al., 2023); (2) *Policy Optimization* directly optimizes reasoning behaviors with RL objectives, where GRPO substantially improves mathematical reasoning (Shao et al., 2024), pure RL induces emergent step-by-step reasoning transferable via distillation (Guo et al., 2025), and recent variants enhance training stability and implicit process supervision (Yu et al., 2025; Fang et al., 2026); (3) *Environment-Grounded Agents* learn through interaction with external or simulated environments, enabling search-augmented reasoning (Jin et al., 2025), simulated retrieval behaviors (Sun et al., 2025), and improved generalization via environment scaling (Fang et al., 2025). Effectively leveraging environment-derived supervision remains an open problem.

E.2 Knowledge Distillation for Large Language Models

Knowledge distillation (Hinton et al., 2015) is a core technique for compressing large models into smaller ones, and LLM distillation methods can be broadly categorized by training paradigm and supervision signal.

Token-Level Distillation. Token-level methods provide dense supervision at each decoding step, offering richer gradient signals than sequence-level approaches. These methods can be categorized by their divergence objectives: (1) *Forward KL* minimizes $\mathrm{KL}(\pi_{\mathcal{E}} \| \pi_S)$, encouraging the student to cover the teacher's probability mass (Hinton et al., 2015; Gu et al., 2024); (2) *Reverse KL* minimizes $\mathrm{KL}(\pi_S \| \pi_{\mathcal{E}})$, promoting mode-seeking behavior that prevents the student from assigning probability mass to unlikely regions (Gu et al., 2024); (3) *Adaptive schemes* dynamically adjust the distillation objective based on token importance, using fine-grained divergence control (Jung et al., 2025) or attribution-based token selection (Wu et al., 2023).

Sequence-Level Distillation. Sequence-level methods use complete outputs or reasoning traces rather than matching every next-token distribution, providing global and structured learning signals. These approaches can be grouped by the form of sequence supervision: (1) *Output Imitation* directly

trains the student on teacher-generated sequences (Kim & Rush, 2016), with extensions based on imitation learning (Lin et al., 2020), symbolic supervision (West et al., 2022), and unified f-divergence formulations (Wen et al., 2023); (2) *Trajectory Traces* leverage intermediate reasoning steps as supervision. Inspired by Chain-of-Thought prompting (Wei et al., 2022), teachers generate step-by-step rationales that are distilled into smaller students (Ho et al., 2023; Magister et al., 2023; Fu et al., 2023), enabling effective transfer of multi-step reasoning ability; (3) *Policy-Based Reasoning Distillation* applies preference optimization to generated chains, as in CoTD-PO (Niu et al., 2025). Related work in general reinforcement learning, such as Proximal Policy Distillation (Spigler, 2024), combines student-driven distillation with PPO but does not study language-model reasoning sequences. Notably, distilling structured rationales can even enable smaller models to outperform larger ones under limited data (Hsieh et al., 2023).

On-Policy vs Off-Policy Distillation. Off-policy distillation pre-generates training data from teacher models, but suffers from distribution mismatch. Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024) addresses this by training on student-generated samples and allows a mixture of on-policy and fixed-data sampling. More recently, Ye et al. (2025) proposed black-box on-policy distillation, which extends on-policy learning to settings where the teacher model's internal probabilities are inaccessible. However, their approach still requires a discriminative model to match the distribution. To address this limitation, our work incorporates verbal scoring from teacher agents, eliminating the need for token-level probability distillation while providing dense step-level supervision for reasoning tasks.

F MATHEMATICAL EVALUATION DATASETS

We describe the datasets used in Table 3 and Figure 6, with evaluated question counts. Evaluation files come from the Qwen2.5-Math evaluation release.

AIME24 (30 problems). AIME24 contains problems from the 2024 American Invitational Mathematics Examination, a competition for students who qualify through the AMC. The questions require mathematical problem solving and have integer-valued answers. See the organizer's overview. Evaluation file: `aime24`.

AMC23 (40 problems). AMC23 comprises a 40-question subset drawn from the 2023 AMC 12A and AMC 12B competitions. We use the problems and answers provided in the evaluation release. Evaluation file: `amc23`.

ASDiv (2,215 problems). ASDiv (Miao et al., 2020) contains English math word problems spanning diverse linguistic patterns and elementary-school problem types, with answers and solution formulas. Evaluation file: `asdiv`.

CARP (976 problems). CARP focuses on computation-intensive middle-school algebra, with natural-language solutions and annotated expression-flow graphs in its original release. We evaluate the English test split. See the authors' dataset release. Evaluation file: `carp_en`.

Middle (101 problems). Middle is the Chinese-language CN Middle School 24 set, drawn from the 2024 high-school entrance examinations. It includes school-level numerical, algebraic, and geometric questions, including multiple-choice items. The Qwen2-Math evaluation overview identifies this benchmark separately from CMATH. Evaluation file: `cn_middle_school`.

College (2,818 problems). CollegeMath contains textbook exercises across seven areas: algebra, precalculus, calculus, vector calculus, probability, linear algebra, and differential equations. We use the 2,818-question test file, which records the source textbook and mathematical topic for each item. Evaluation file: `college_math`.

Gaokao-II (14 problems). Gaokao-II contains Chinese mathematics questions from the 2024 Gaokao Paper II. The released evaluation subset comprises eight single-choice, three multiple-choice, and three fill-in-the-blank questions; it is not the complete examination paper. Evaluation file: `gaokao2024_II`.

MATH (5,000 problems). MATH contains competition mathematics problems with worked solutions, covering a range of topics and difficulty levels. We evaluate its 5,000-problem test split rather than the smaller MATH500 subset. See the original dataset paper. Evaluation file: `math`.

MAWPS (2,065 problems). MAWPS brings together arithmetic and algebra word problems from several sources. Solving these problems requires translating the quantities and relationships described in text into mathematical operations. We use the 2,065-question collection in the evaluation release. See the original dataset paper. Evaluation file: `mawps`.

Olympiad (675 problems). Olympiad denotes the mathematics evaluation subset of OlympiadBench (He et al., 2024), not the full bilingual, multimodal benchmark. It contains competition-level problems with worked solutions. Evaluation file: `olympiadbench`.

SAT-Math (32 problems). SAT-Math contains SAT-style mathematics questions with answer choices, including equations, functions, and geometry. We evaluate the 32 questions provided in the released file, not a complete SAT examination. The College Board overview describes the broader SAT mathematics content. Evaluation file: `sat_math`.

TABMWP (1,000 problems). TABMWP (Lu et al., 2022) pairs mathematical word problems with tabular data. We use the evaluation subset with text-form tables, answers, and worked solutions. Evaluation file: `tabmwp`.