SPEC-Distill: Towards Efficient Speculative Distillation for Reinforcement Learning with Verifiable Rewards

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) improves mathematical reasoning, but sparse sequence-level rewards make training heavily dependent on long, repeated rollout generation. On-policy distillation (OPD) makes these rollouts more informative by asking a larger teacher to score every student trajectory token by token, but this inserts large-teacher forward passes into every RL step and requires online resources beyond standard student-only RL. We propose *SPEC-Distill*, a speculative prefix distillation framework in which the teacher shapes rollout construction rather than defining an auxiliary reward. Instead of querying the teacher during training, SPEC-Distill runs it *once* offline to build a reusable pool of expert trajectories. During RL, the student treats these trajectories as speculative drafts, accepts a prefix under a lenience-controlled criterion, completes the response under its own policy, and is updated only by verifiable rewards with standard GRPO. Against OPD, this removes online teacher scoring; against standard autoregressive rollouts, accepted prefixes reduce the number of fresh student-generated tokens. In the primary non-think setting, SPEC-Distill improves Qwen3-4B from 66.1 to 73.2 average accuracy over GRPO, and outperforms OPD on the two scales. Compared with full-response expert reuse such as Luffy, SPEC-Distill reaches comparable accuracy on Qwen3-1.7B while reducing online rollout tokens by about 60% and rollout time from 10.90 to 7.06 hours. Analysis further shows that teacher prefixes are best used as an early scaffold, after which reusing the student's own rollouts provides the strongest balance between expert guidance and on-policy exploration.

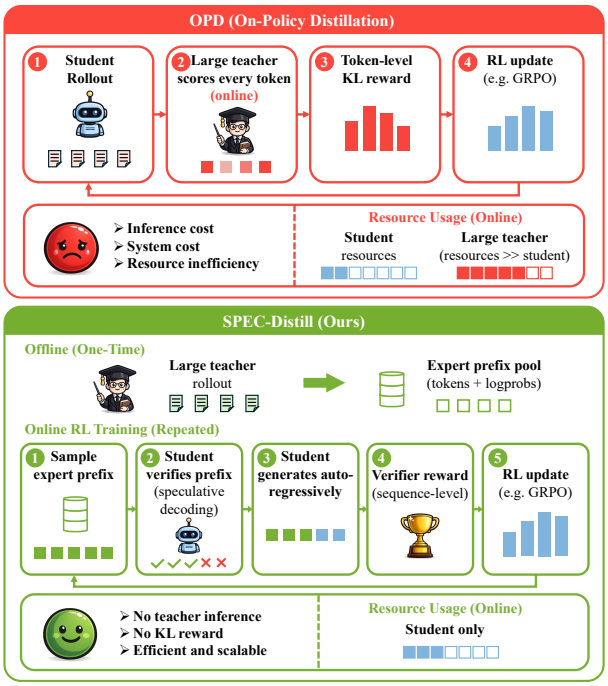


Figure 1: Online teacher scoring versus offline prefix reuse. OPD couples each RL step to large-teacher inference in order to construct token-level KL rewards. SPEC-Distill instead pays the teacher cost once to build an expert prefix pool; online training then verifies reusable prefixes with the student, completes each response under the student policy, and optimizes only verifier rewards.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a practical recipe for eliciting stronger reasoning in language models, especially on mathematical and coding tasks where correctness can be checked deterministically (Cobbe et al., 2021; Chen et al., 2021; Shao et al., 2024; DeepSeek-AI, 2025). By replacing learned reward models with exact-match answers, symbolic verifiers, or unit tests, recent GRPO-style systems have driven rapid progress in math-centric post-training (Shao et al., 2024; DeepSeek-AI, 2025; Wang et al., 2025). However, RLVR remains rollout-intensive: verifier rewards are typically sparse and sequence-level, so each update requires

many long samples before the policy receives useful learning signals. On-policy distillation (OPD) attempts to make these rollouts more informative by adding a stronger teacher to the RL stage: the student samples trajectories, the teacher scores them online token by token, and teacher-student divergence provides dense supervision(Agarwal et al., 2024; Lu and Lab, 2025). Regardless of how the teacher is deployed, OPD requires an additional large-teacher inference pass over student trajectories before every policy update. When the teacher is much larger than the student, this online scoring path can dominate memory, compute, and scheduling overhead beyond the student-only RL loop.

This raises a complementary question: can the student still benefit from a stronger teacher without invoking that teacher to score every token? We answer by separating the teacher's exploratory role from the verifier's evaluative role. The teacher should guide **where** the student explores, while the verifier remains the judge of **what** the student optimizes. This separation suggests a different form of distillation: the teacher should shape the student's training experience, not become an online reward model.

We propose **SPEC-Distill**, a distillation framework for RLVR built around this principle. SPEC-Distill first runs a stronger teacher offline to collect expert trajectories. During online training, these trajectories serve as speculative drafts: the student verifies each draft token by token, accepts only a compatible prefix, and then completes the response under its own policy. The completed rollouts are scored solely by the verifier and optimized with the standard GRPO objective, with no teacher forward pass or KL-based reward inside the RL loop. After the initial teacher-prefix stage, SPEC-Distill switches the speculative source to the student's own cached rollouts, so guidance gradually shifts from teacher experience to self-generated experience as the policy improves.

We evaluate SPEC-Distill primarily in the non-think RLVR setting across Qwen3-1.7B, 4B, and 8B students. SPEC-Distill improves average accuracy over standard GRPO from 50.9 to 55.2, 66.1 to 73.2, and 73.2 to 76.0, respectively, while keeping the online loop close to vanilla GRPO: student rollout, verifier scoring, and policy update. The strongest gap appears on Qwen3-4B, where SPEC-Distill outperforms OPD by 14.1 average points (73.2 vs. 59.1) while using fewer online rollout tokens (242.5M vs. 372.3M) and removing online teacher scoring. In the matched Qwen3-1.7B comparison with Luffy, SPEC-Distill trades only 1.0 average point for a much smaller online rollout budget, reducing online rollout tokens from 660.0M to 264.3M and rollout time from 10.90 to 7.06 hours.

Contributions. We summarize our contributions as follows.

- We reinterpret the teacher's role in RLVR distillation: rather than providing dense token-level rewards on every student rollout, the teacher supplies offline speculative prefixes that guide rollout generation, while optimization remains verifier-driven.
- We propose SPEC-Distill, a multi-stage distillation framework that keeps the online RL loop free of teacher forward passes and reduces fresh rollout generation by reusing accepted speculative prefixes.
- We identify the speculative source schedule as a key design choice and show that teacher-to-self reuse after the initial teacher-prefix stage outperforms weaker reuse strategies.
- We validate SPEC-Distill on instruction-tuned non-think RLVR across student scales, with an additional base-model zero-RL check in the appendix, showing gains over GRPO and OPD where available with a favorable rollout-efficiency trade-off.

2 Related Work

2.1 RLVR and Rollout-Level Experience Reuse

Reinforcement learning with verifiable rewards (RLVR) improves reasoning by replacing learned reward models with deterministic verifiers such as answer checkers, symbolic evaluators, and unit tests (Cobbe et al., 2021; Lightman et al., 2023; Chen et al., 2021). Recent math post-training systems scale these signals with algorithms such as GRPO, which normalizes rewards within sampled response groups and applies a clipped policy-gradient update (Shao et al., 2024; DeepSeek-AI, 2025; Wang et al., 2025). Because these verifiers typically provide only sequence-level success or failure, the rollout distribution plays a central role in determining which reasoning paths receive useful feedback. Under sparse sequence-level rewards,

prior work improves rollouts either by reusing expert experience, as in LUFFY's teacher-generated reasoning traces (Yan et al., 2025), or by improving rollout efficiency, as in SPEC-RL's speculative reuse of cached student rollouts (Liu et al., 2025). Both lines can be viewed as changing how candidate trajectories are proposed, while leaving the verifier as the training signal.

SPEC-Distill connects these two directions. It uses teacher trajectories as high-quality drafts, but reuses them through speculative verification rather than injecting full expert responses. The student accepts only the prefix compatible with its current policy, completes the response itself, and is updated by the original verifier reward. Thus, teacher experience affects the rollout path rather than replacing verifier-driven optimization.

2.2 Knowledge Distillation

Knowledge distillation trains a student to match a stronger teacher through softened predictions, demonstrations, or KL-based imitation losses (Hinton et al., 2015; Kim and Rush, 2016; Sanh et al., 2019; Gu et al., 2024; Ko et al., 2024). Off-policy distillation can prepare teacher outputs in advance, but optimizes toward fixed teacher data rather than trajectories induced by the current student. On-policy distillation reduces this mismatch by querying a teacher on student-generated trajectories: GKD minimizes teacher-student divergence on student outputs (Agarwal et al., 2024), while OPD-style RLVR methods use teacher-student KL as dense reward (Lu and Lab, 2025). This yields token-level supervision on the student's own samples, but it also changes the online RL pipeline from verifier-only feedback to teacher-rescored feedback. The trade-off is that every student trajectory must be rescored by the teacher before the policy update, adding substantial overhead beyond standard student-only RL.

SPEC-Distill avoids the OPD bottleneck by paying the teacher cost once offline. Unlike full-response expert-reuse methods such as LUFFY, however, our main efficiency gain comes from prefix-level speculative reuse: the student verifies an offline expert draft, reuses only the accepted prefix, and generates the remaining continuation itself. This keeps the teacher outside the online loop while also reducing the number of fresh on-policy rollout tokens. In this sense, SPEC-Distill sits between offline imitation and online teacher scoring: it reuses teacher structure as a proposal source, but keeps the policy update and final credit assignment tied to verifier rewards.

3 Method

3.1 Problem Setup

Let $\mathcal{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ denote the training prompt set. The student policy π_θ generates a response $\mathbf{y}$ for a prompt $\mathbf{x}\in\mathcal{X}$, written as $\mathbf{y}\sim\pi_\theta(\cdot\mid\mathbf{x})$. A deterministic verifier assigns each completed trajectory $(\mathbf{x},\mathbf{y})$ a binary reward $R(\mathbf{x},\mathbf{y})\in\{0,1\}$. Group Relative Policy Optimization (GRPO; Shao et al. 2024) samples a group of G responses $\{\mathbf{y}_1,\ldots,\mathbf{y}_G\}$ for each prompt, normalizes the rewards within the group to obtain advantages $\hat{A}_i$, and updates the policy with a clipped surrogate objective:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta)=\mathbb{E}_{\substack{\mathbf{x}\sim\mathcal{X}\\ \{\mathbf{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|\mathbf{x})}}\Big[\sum_{i=1}^{G}\min\big(\rho_i\hat{A}_i,\ \mathrm{clip}(\rho_i,1-\epsilon,1+\epsilon)\,\hat{A}_i\big)\Big], \tag{1}$$

where $\rho_i=\pi_\theta(\mathbf{y}_i\mid\mathbf{x})/\pi_{\theta_{\mathrm{old}}}(\mathbf{y}_i\mid\mathbf{x})$ and ϵ is the clipping threshold.

Speculative decoding in RL. SPEC-RL (Liu et al., 2025) treats previously generated trajectories as speculative drafts during on-policy training. A cached trajectory proposes a sequence of tokens, and the current policy π_θ accepts each draft token y_t with probability

$$\alpha_t=\min\Big(1,\ \ell\cdot\frac{\pi_\theta(y_t\mid\mathbf{x},\mathbf{y}_{<t})}{\hat{\pi}(y_t\mid\mathbf{x},\mathbf{y}_{<t})}\Big), \tag{2}$$

where $\hat{\pi}$ is the distribution under which the draft was generated and $\ell\geq1$ is a lenience parameter. Raising ℓ above 1 relaxes the criterion so that more draft tokens are reused even when the current policy assigns them lower probability, which reduces the number of tokens that must be produced from scratch. SPEC-Distill reuses this acceptance machinery, but changes the source of the drafts.

3.2 Speculative Prefix Distillation

SPEC-Distill turns teacher guidance into a rollout-construction mechanism for RLVR. It consists of an offline cache-construction stage and an online speculative rollout stage. Offline, a frozen teacher is run once to build a reusable cache of expert drafts. Online, early RL steps draw drafts from this teacher cache; after the teacher-prefix phase, the

draft source switches to the student rollout cache. The accepted teacher tokens serve only as a rollout prefix; the verifier reward and policy update remain unchanged. This staged design is important: simply replacing SPEC-RL's draft source with a teacher cache for all of training would keep rollout proposals anchored to a static offline distribution, whereas SPEC-Distill uses the teacher as an early scaffold and then lets the student recycle its own evolving experience.

The rest of this subsection formalizes the three modules used by SPEC-Distill. We first define the offline teacher cache $\mathcal{C}_T$, then introduce a source schedule that chooses between teacher drafts and student drafts during online RL, and finally describe the shared speculative verification rule that turns a draft into an accepted prefix plus a student-generated continuation. Algorithm 1 summarizes the complete pipeline.

Offline expert experience generation. Before RL begins, the frozen teacher π_T generates expert drafts offline. For each prompt $\mathbf{x} \in \mathcal{X}$, we sample G teacher drafts

$$\mathbf{y}_i^T \sim \pi_T(\cdot \mid \mathbf{x}), \qquad i = 1, \ldots, G,$$

and store each draft together with its teacher token probabilities

$$\mathbf{p}_i^T = [\pi_T(y_{i,t}^T \mid \mathbf{x}, \mathbf{y}_{i,<t}^T)]_{t=1}^{|\mathbf{y}_i^T|},$$

and insert the pair into the teacher cache:

$$\mathcal{C}_T(\mathbf{x}) \leftarrow \mathcal{C}_T(\mathbf{x}) \cup \{(\mathbf{y}_i^T, \mathbf{p}_i^T)\}.$$

The teacher cache is the only teacher artifact consumed by the RL stage and the teacher model itself is neither loaded nor queried during online optimization.

Online source schedule. During online RL, each rollout draws its draft from one of two speculative source caches. For the first K_T optimization steps, drafts are sampled from the offline teacher cache $\mathcal{C}_T$ with lenience ℓ_T. Afterward, drafts are sampled from the student cache $\mathcal{C}_S$ with lenience ℓ_S. The student cache stores previously completed student rollouts together with the token probabilities needed for acceptance. It is refreshed from completed student rollouts during training, so the second phase reuses trajectories produced by student snapshots that actually appeared in the RL loop. In practice, we choose K_T to cover the initial teacher-prefix pass over the training prompts, so $\mathcal{C}_S(\mathbf{x})$ has been populated before it is used as the draft source.

Algorithm 1 SPEC-Distill

Require: Prompts $\mathcal{X}$, teacher π_T, student π_θ, group size G, RL steps K, teacher-prefix steps K_T, leniences ℓ_T, ℓ_S.
Output: updated student policy
Offline expert experience generation
$\mathcal{C}_T, \mathcal{C}_S \leftarrow \emptyset$
for $\mathbf{x} \in \mathcal{X}$ **do**
  for $i = 1, \ldots, G$ **do**
    $\mathbf{y}_i^T \sim \pi_T(\cdot \mid \mathbf{x})$
    $\mathbf{p}_i^T \leftarrow [\pi_T(y_{i,t}^T \mid \mathbf{x}, \mathbf{y}_{i,<t}^T)]_{t=1}^{|\mathbf{y}_i^T|}$
    $\mathcal{C}_T(\mathbf{x}) \leftarrow \mathcal{C}_T(\mathbf{x}) \cup \{(\mathbf{y}_i^T, \mathbf{p}_i^T)\}$
  end for
end for
Online speculative prefix distillation
for $k = 1, \ldots, K$ **do**
  for $\mathbf{x}$ in batch $\mathcal{B}_k$ sampled from $\mathcal{X}$ **do**
    $\mathcal{C}_S^{\text{new}}(\mathbf{x}) \leftarrow \emptyset$
    for $i = 1, \ldots, G$ **do**
      if $k \leq K_T$ **then**
        $(\mathbf{y}_i^d, \mathbf{p}_i^d) \sim \mathcal{C}_T(\mathbf{x})$
        $\ell \leftarrow \ell_T$
      else
        $(\mathbf{y}_i^d, \mathbf{p}_i^d) \sim \mathcal{C}_S(\mathbf{x})$
        $\ell \leftarrow \ell_S$
      end if
      $\mathbf{y}_i \leftarrow \textsc{SpecRollout}(\pi_\theta, \mathbf{x}, \mathbf{y}_i^d, \mathbf{p}_i^d, \ell)$
      $\mathbf{p}_i^\theta \leftarrow [\pi_\theta(y_{i,t} \mid \mathbf{x}, \mathbf{y}_{i,<t})]_{t=1}^{|\mathbf{y}_i|}$
      $\mathcal{C}_S^{\text{new}}(\mathbf{x}) \leftarrow \mathcal{C}_S^{\text{new}}(\mathbf{x}) \cup \{(\mathbf{y}_i, \mathbf{p}_i^\theta)\}$
    end for
    $r_i \leftarrow R(\mathbf{x}, \mathbf{y}_i), \quad i = 1, \ldots, G$
    $\mathcal{C}_S(\mathbf{x}) \leftarrow \mathcal{C}_S^{\text{new}}(\mathbf{x})$
  end for
  Update π_θ using $\{(\mathbf{x}, \mathbf{y}_i, r_i) : \mathbf{x} \in \mathcal{B}_k,\ i = 1, \ldots, G\}$
end for

Draft verification and continuation. We now describe the SPECROLLOUT procedure used in Algorithm 1:

$$\textsc{SpecRollout}(\pi_\theta, \mathbf{x}, \mathbf{y}^d, \mathbf{p}^d, \ell)$$

Given a cached draft $\mathbf{y}^d = (y_1^d, \ldots, y_L^d)$ for prompt $\mathbf{x}$, $\mathbf{p}^d = (p_1^d, \ldots, p_L^d)$ stores the token probabilities assigned by the model that generated the draft. For teacher drafts, these probabilities come from the offline teacher cache $\mathcal{C}_T$; for student drafts, they come from the cached student rollout in $\mathcal{C}_S$.

The current student first evaluates the draft tokens in parallel under its own policy:

$$p_i^{\text{curr}} = \pi_\theta(y_i^d \mid \mathbf{x}, \mathbf{y}_{<i}^d), \qquad i = 1, \ldots, L.$$

Each draft token is then accepted sequentially with probability

$$\tilde{\alpha}_i = \min\left(1,\ \ell \cdot \frac{p_i^{\text{curr}}}{p_i^d}\right). \tag{3}$$

Let n be the first rejected position, i.e.,

$$n = \min\{i : u_i > \tilde{\alpha}_i\}, \qquad u_i \sim \mathcal{U}(0,1),$$

and set $n = L + 1$ if all draft tokens are accepted. The accepted prefix is $\mathbf{y}^d_{<n}$. When $n \leq L$, the remaining draft suffix is discarded and the current student generates a fresh continuation:

$$\mathbf{y}^\theta_{\geq n} \sim \pi_\theta(\cdot \mid \mathbf{x}, \mathbf{y}^d_{<n}).$$

If all draft tokens are accepted, we set $\mathbf{y}^\theta_{\geq n}$ to the empty sequence, so the returned rollout reduces to the full draft $\mathbf{y} = \mathbf{y}^d$. The final rollout $\mathbf{y}$ returned by SPECROLLOUT is $\mathbf{y}^d_{<n} \oplus \mathbf{y}^\theta_{\geq n}$. The completed trajectories are scored by the same verifier reward $R(\mathbf{x}, \mathbf{y})$ and then passed to the base GRPO update in Equation (1).

Algorithm 1 summarizes the rollout construction process of SPEC-Distill. The teacher is used only in the offline cache-construction step, where it generates the fixed teacher cache $\mathcal{C}_T$ together with token probabilities. During online RL, each rollout first samples a cached draft according to the source schedule: early steps draw from $\mathcal{C}_T$, while later steps draw from the refreshed student cache $\mathcal{C}_S$. Both sources are consumed by the same SPECROLLOUT procedure, which verifies a compatible prefix and lets the current student generate the remaining continuation. After each rollout batch, completed student trajectories are written into $\mathcal{C}_S$ for later self-reuse. Thus, after the offline teacher pass, the online training loop only requires the student, the rollout engine, and the verifier.

3.3 Key Properties of SPEC-Distill

Teacher-free inner loop. After the offline stage, the teacher is neither loaded nor forwarded during RL optimization; the online loop needs only the student, the rollout engine, and the verifier. This is in direct contrast to OPD-style recipes that must rescore student trajectories with the teacher during training (Agarwal et al., 2024; Lu and Lab, 2025).

Prefix-level controllability. Teacher information enters SPEC-Distill only through rollout construction. Its strength is therefore controlled by the source schedule and the accepted-prefix length, rather than by an auxiliary loss or a teacher-scored reward. In particular, K_T determines how long drafts are drawn from the teacher cache $\mathcal{C}_T$, while ℓ_T and ℓ_S control how permissively teacher and student drafts are reused. Larger lenience values increase the expected accepted-prefix length and reduce fresh student generation, whereas smaller values return control earlier to the current student policy. This provides an interpretable way to balance teacher imitation and student exploration without changing the verifier reward or the GRPO update.

Plug-and-play and efficiency. Since SPEC-Distill only modifies how rollouts are constructed, it can be paired with RLVR algorithms such as GRPO, PPO, REINFORCE, and their variants while leaving reward computation and policy updates unchanged. Its efficiency benefit comes from reusing accepted draft prefixes. Although accepted draft tokens still require student verification, they avoid fresh autoregressive sampling for the accepted segment. In our rollout setting, this substantially reduces online rollout tokens and wall-clock rollout time.

4 Experiments

4.1 Experimental Setup

Training data. Following the DeepMath-6K setting used in SPEC-RL (Liu et al., 2025), we sample 6,144 verifiable reasoning problems from DeepMath-103K (He et al., 2025) for RL training. We refer to this prompt set as DEEPMATH-6K and keep it fixed across all methods.

Settings and models. We evaluate SPEC-Distill primarily in the non-think RLVR setting, with detailed hyperparameters summarized in Appendix A.1 and Table 3. Our primary setting follows G-OPD (Yang et al., 2025b) and uses Qwen3 models (Yang et al., 2025a) in **non-think** mode, where the students are Qwen3-1.7B, Qwen3-4B, and Qwen3-8B, and the teacher is Qwen3-14B. This setting tests whether expert prefixes improve RLVR even when both the teacher and student are already strong; Appendix A.3 reports a complementary zero-RL check with base checkpoints. All experiments are implemented with VERL (Sheng et al., 2024) and VLLM (Kwon et al., 2023).

Baselines. We compare SPEC-Distill with three categories of baselines. First, we include the raw **Base model** and standard RLVR with **GRPO** (Shao et al., 2024), which measure performance without expert guidance. Second, we compare with **Luffy** (Yan et al., 2025), which guides exploration by mixing one full expert trajectory into each GRPO response group. Third, we compare

Method	Efficiency			Performance							
	Off-policy Tokens	On-policy Tokens	Rollout Time / Hours	AMC23	AIME24	AIME25	GSM8K	MATH500	Olympiad	MMLU-STEM	Avg.
Student: *Qwen3-1.7B*											
Base model	–	–	–	44.8	11.2	10.0	81.2	71.2	35.4	72.6	46.6
GRPO†	–	462.7	7.07	51.2	15.3	12.4	84.0	77.2	43.4	72.7	50.9
OPD‡	–	559.4	12.04	49.5	16.1	12.9	82.9	74.4	40.7	72.5	49.9
Luffy†	257.8	660.0	10.90	**64.4**	22.3	**19.6**	**85.7**	**82.4**	**47.7**	71.5	**56.2**
SPEC-Distill†	257.8	**264.3**	**7.06**	60.0	**24.4**	17.9	**85.7**	78.4	46.4	**73.6**	55.2
Student: *Qwen3-4B*											
Base model	–	–	–	66.7	21.0	18.8	91.2	81.8	48.1	83.9	58.8
GRPO†	–	231.6	**8.19**	75.9	37.4	27.7	**93.0**	86.8	57.0	**84.6**	66.1
OPD‡	–	372.3	8.55	67.5	23.1	18.1	90.8	82.8	47.9	83.8	59.1
Luffy†	257.8	727.7	25.68	**89.5**	**57.8**	**47.6**	92.9	**91.0**	57.5	35.1	67.3
SPEC-Distill†	257.8	**242.5**	9.30	87.8	52.3	42.3	92.7	90.8	**63.1**	83.4	**73.2**
Student: *Qwen3-8B*											
Base model	–	–	–	71.6	26.0	19.7	93.2	85.4	51.6	85.6	61.9
GRPO†	–	568.7	12.99	86.7	52.2	38.9	94.5	91.6	61.5	**87.3**	73.2
SPEC-Distill†	257.8	**234.8**	**12.49**	**89.4**	**60.6**	**45.0**	**95.1**	**92.6**	**62.7**	86.6	**76.0**

Table 1: Results of SPEC-Distill and baselines on Qwen3 models in non-think mode across three student scales. For the 1.7B, 4B, and 8B students, all teacher-based methods use Qwen3-14B as the teacher. All methods are compared at matched training checkpoints to ensure fair comparison. † denotes online-teacher free while ‡ **requires** online teacher during training.

with **OPD** (Lu and Lab, 2025), which uses online teacher scoring to construct dense rewards. For OPD, we follow the open-source matched non-think setting of G-OPD (Yang et al., 2025b).

Evaluation. We evaluate mathematical reasoning on AMC23 (Mathematical Association of America, 2026a), AIME24/25 (Mathematical Association of America, 2026b), GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), and OlympiadBench (He et al., 2024), and report STEM knowledge performance on MMLU-STEM (Hendrycks et al., 2021a). We use AVG@16 for AMC23 and AVG@32 for AIME24/25. Different tables average over different benchmark subsets, so averages are comparable within each table but not necessarily across tables.

4.2 Non-Think Distillation

Table 1 reports the main non-think results across three Qwen3 student scales. SPEC-Distill consistently improves over GRPO, raising average accuracy from 50.9 to 55.2 on Qwen3-1.7B, 66.1 to 73.2 on Qwen3-4B, and 73.2 to 76.0 on Qwen3-8B. The gains are concentrated on competition-style benchmarks such as AMC23, AIME, and OlympiadBench, where sparse verifier rewards make exploration hardest. This supports the role of offline teacher prefixes as rollout guidance rather than as a replacement for verifier-driven learning.

The same table also shows a favorable efficiency trade-off. Compared with OPD, which requires online teacher scoring, SPEC-Distill improves average accuracy by $+5.3$ points on Qwen3-1.7B and $+14.1$ points on Qwen3-4B while using fewer online rollout tokens (264.3M vs. 559.4M, and 242.5M vs. 372.3M). Compared with Luffy, SPEC-Distill is within roughly one average point on Qwen3-1.7B but reduces online rollout tokens from 660.0M to 264.3M and rollout time from 10.90 to 7.06 hours. On Qwen3-4B, Luffy remains strong on math benchmarks but also produces much longer rollouts and drops sharply on MMLU-STEM, which may indicate mild math-domain over-specialization rather than uniformly broad gains. Thus, SPEC-Distill improves over GRPO and OPD while approaching full-response expert reuse with substantially less online rollout work; reward trajectories are shown in Appendix A.2.

4.3 Efficiency Analysis

The efficiency benefit of SPEC-Distill is that teacher inference is paid once to build the offline cache, after which the RL loop reuses stored drafts as prefix proposals without online teacher scoring. By contrast, OPD-style distillation must rescore student trajectories with the teacher at every RL step to construct token-level KL rewards, so its teacher cost grows with both training steps and rollout length. Table 1 makes this accounting concrete. For all students, SPEC-Distill uses the same 257.8M offline teacher-token budget as Luffy, but it consumes much fewer online rollout tokens: 264.3M on Qwen3-1.7B compared with Luffy's

Method	Post-prefix Reuse	AMC23	AIME24	AIME25	GSM8K	MATH500	Olympiad	MMLU STEM	AVG
GRPO	–	51.2	15.3	12.4	84.0	77.2	43.4	72.7	50.9
SPEC-Distill	None	56.7	18.1	13.6	85.0	76.4	44.7	72.6	52.4
SPEC-Distill	Random	55.9	19.7	17.0	84.8	**78.6**	44.0	72.8	53.3
SPEC-Distill	Speculative	**60.0**	**24.4**	**17.9**	**85.7**	78.4	**46.4**	**73.6**	**55.2**

Table 2: Ablation study on the post-prefix reuse strategy, evaluated on Qwen3-1.7B with a Qwen3-14B teacher. All SPEC-Distill variants use the same offline teacher cache $\mathcal{C}_T$ and the same first-K_T teacher-prefix phase; they differ only after the source schedule switches away from $\mathcal{C}_T$. **None** reverts to fresh GRPO-style rollouts; **Random** reuses prefixes with uniformly random truncation points; **Speculative** (default) samples drafts from $\mathcal{C}_S$ and applies the lenience-controlled acceptance criterion of Equation (3).

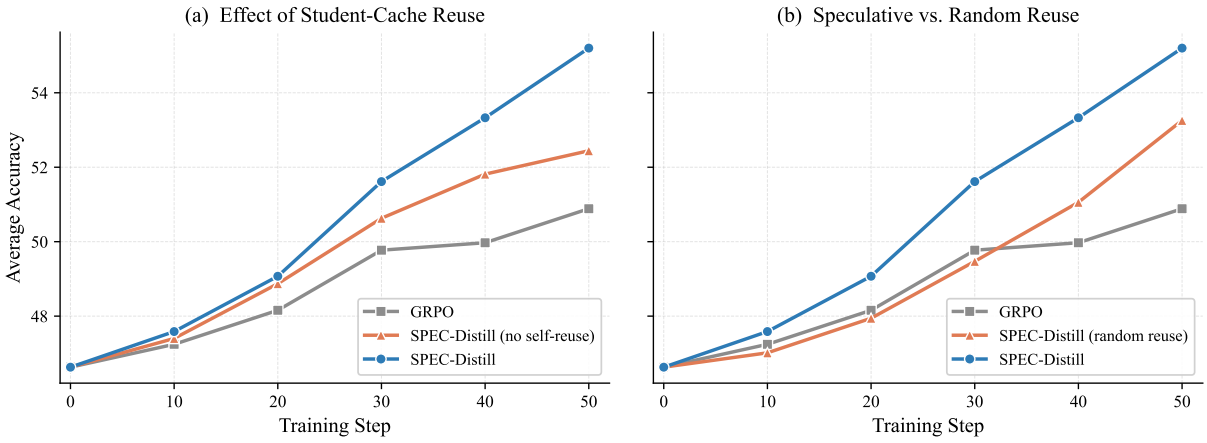


Figure 2: Step-level average accuracy during training for the two ablation experiments on Qwen3-1.7B (non-think) with a Qwen3-14B teacher. **(a)** After the teacher-prefix phase, removing student-cache reuse (orange) narrows the gap to GRPO (gray), whereas the full SPEC-Distill (blue) steadily pulls ahead. **(b)** Random prefix reuse (orange) improves over GRPO but lags behind the speculative acceptance criterion used by SPEC-Distill (blue), especially in later training steps.

660.0M, and 242.5M on Qwen3-4B compared with Luffy's 727.7M. Against OPD, the comparison is even more favorable online: OPD requires 559.4M and 372.3M student rollout tokens on the 1.7B and 4B students, respectively, and additionally requires online teacher scoring over those trajectories. By contrast, SPEC-Distill has no teacher forward pass in the RL loop and reduces the number of student-generated rollout tokens by reusing accepted prefixes. This is the cost regime targeted by SPEC-Distill: keep the expert's exploratory benefit, but make the repeated online loop close to student-only RL; Appendix A.4 breaks down prefix reuse by student scale.

4.4 Ablation Study

We ablate the post-prefix rollout strategy in SPEC-Distill using Qwen3-1.7B with a Qwen3-14B teacher. All variants use the same offline teacher cache $\mathcal{C}_T$ and the same first-K_T teacher-prefix phase. They differ only after the source schedule switches away from $\mathcal{C}_T$, when the default method samples drafts from the student cache $\mathcal{C}_S$; Appendix A.5 separately varies the teacher-prefix duration. Table 2 and Figure 2 report the results.

Effect of student-cache reuse. After the first-K_T teacher-prefix phase, SPEC-Distill by default samples drafts from the student cache $\mathcal{C}_S$ and applies the same speculative acceptance criterion. Removing this student-cache reuse entirely ("None" in Table 2), so that post-prefix training reverts to fresh GRPO-style rollouts, reduces the average from 55.2 to 52.4. The drop is concentrated on the competition-level benchmarks where exploration is hardest: AIME24 falls from 24.4 to 18.1 (-6.3) and AMC23 from 60.0 to 56.7 (-3.3). Teacher-prefix distillation alone already lifts performance above GRPO (52.4 vs. 50.9), but the student-cache phase contributes a further $+2.8$ points by reinforcing the reasoning patterns absorbed from the teacher—keeping those prefix

structures in the rollout distribution prevents the student from drifting back toward its pre-distillation behaviour. As Figure 2(a) shows, the gap between the full method and the no-student-cache-reuse variant widens steadily over training, indicating that self-reuse has a compounding effect rather than a one-time benefit.

Speculative vs. random reuse. We next compare the lenience-controlled acceptance criterion (Equation (3)) against a simpler baseline that draws from the same student cache $\mathcal{C}_S$ but selects truncation points uniformly at random. Random reuse (avg. 53.3) improves over no reuse (52.4), confirming that prefix reuse provides a training signal beyond vanilla GRPO. However, the speculative criterion (avg. 55.2) outperforms random reuse by $+1.9$ points, with the largest gaps on the hardest benchmarks (AIME24: $+4.7$, AMC23: $+4.1$, Olympiad: $+2.4$). Figure 2(b) illustrates the divergence: the two curves track each other in the first 20 steps but then separate, suggesting that the speculative criterion becomes increasingly important as the student matures and its distribution shifts further from the cached drafts. The advantage stems from tying the reuse decision to the student's own token-level confidence: the speculative acceptance keeps reused prefixes closer to the current policy and places the continuation point where the student's distribution genuinely diverges from the draft, rather than at an arbitrary position.

Teacher footprint during self-reuse. To better understand why post-prefix self-reuse helps, we use exact-prefix overlap with the offline teacher trajectory as a conservative proxy for how much teacher-induced rollout structure remains in the current rollout distribution. For each rollout, we compute the longest common prefix with the corresponding teacher draft and normalize it by the current rollout length. Figure 3 shows that this footprint behaves very differently after the teacher-prefix phase: the **None** variant drops from roughly 11% during the teacher-prefix phase to about 1–2$\%$ in later epochs, whereas the **Speculative** variant keeps it around 10% immediately after the switch before gradually decaying to 2–3$\%$. This explains the $+2.8$ average improvement over the no-reuse variant in Table 2: self-reuse keeps teacher-derived prefix structure available long enough for the student to benefit from it, while the gradual decay indicates that the policy is not locked to fixed teacher trajectories and can continue verifier-driven exploration.

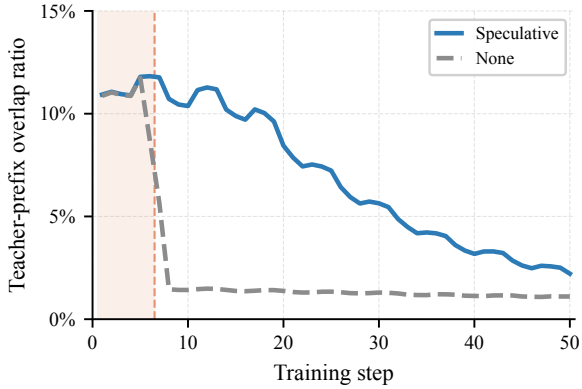


Figure 3: Persistence of teacher-prefix structure during training. We measure teacher footprint as the fraction of the current rollout, in characters, that exactly matches the prefix of the corresponding offline teacher draft. The shaded region denotes the initial teacher-prefix phase. For **None**, the footprint drops sharply after the first epoch, indicating that fresh GRPO-style rollouts quickly lose teacher-induced rollout structure. For **Speculative**, the footprint stays substantially higher after the teacher stage and then gradually decays, suggesting that student-cache reuse preserves an early teacher scaffold while still allowing the policy to move away from the fixed teacher distribution.

5 Conclusion

We introduced SPEC-Distill, a speculative prefix distillation framework that moves the teacher from the online reward path to the offline rollout-proposal path. A cached teacher trajectory provides an initial draft; the student accepts only a compatible prefix, completes the rollout under its own policy, and learns from verifier rewards with standard GRPO. This keeps the online stage close to student-only RL while changing the rollout distribution through accepted expert and self prefixes. Across non-think Qwen3 settings, this design improves over standard RLVR and OPD-style distillation when available, while reducing online rollout work. The ablations further show that teacher prefixes work best as an early scaffold followed by speculative self-reuse, which preserves useful rollout structure without keeping the policy fixed to the teacher. The teacher-footprint analysis supports this view: self-reuse keeps a measurable trace of teacher-induced structure after the initial prefix stage, then lets it decay as the student policy becomes more autonomous. Overall, SPEC-Distill suggests that in verifier-driven RL, strong teachers can be especially useful as offline sources of exploration rather than online reward models.

Limitations

Our experiments focus on teacher–student pairs with compatible tokenization and response formats, which keeps speculative prefix verification simple; extending the same idea to more heterogeneous model families is an interesting direction. We also use fixed lenience values and source schedules, and adaptive schedules that track the student's evolving capability may further improve the teacher-to-self transition. Finally, we evaluate primarily on mathematical reasoning, while domains with reliable automated feedback, such as code generation, are natural targets for future study.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. 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*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

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

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *arXiv preprint arXiv:2309.06180*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Bingshuai Liu, Ante Wang, Zijun Min, Liang Yao, Haibo Zhang, Yang Liu, Anxiang Zeng, and Jinsong Su. 2025. Spec-rl: Accelerating on-policy reinforcement learning via speculative rollouts. *arXiv preprint arXiv:2509.23232*.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/on-policy-distillation.

Mathematical Association of America. 2026a. American mathematics competitions. https://maa.org/student-programs/amc/. Accessed: 2026-05-21.

Mathematical Association of America. 2026b. Maa invitational competitions. https://maa.org/maa-invitational-competitions/. Accessed: 2026-05-21.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *Preprint*, arXiv:2504.14945.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2025b. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*.

A Experimental Details

A.1 Hyperparameters

Setting	Non-think RLVR	Zero-RL
Student	Qwen3-1.7B/4B/8B	Qwen3-1.7B-Base
Teacher	Qwen3-14B	Qwen3-14B-Base
Max response length	16,384	4,096
Group size	8	8
Prompt batch size	1,024	1,024
Teacher-prefix phase	1 epoch	1 epoch
Lenience	$\ell_T = e^4$, $\ell_S = e^{0.5}$	$\ell_T = e^4$, $\ell_S = e^{0.5}$

Table 3: Experimental settings for the main non-think RLVR experiments and the appendix zero-RL check.

Table 3 summarizes the training hyperparameters used in the main non-think RLVR experiments and the appendix zero-RL. The main experiments use instruction-tuned Qwen3 checkpoints in non-think mode across three student scales, while the complementary zero-RL uses a Qwen3-1.7B-Base student with a Qwen3-14B-Base teacher. The zero-RL run uses a shorter 4,096-token response budget; otherwise, we keep the group size, effective prompt batch size, teacher-prefix duration, and lenience values aligned with the main setting. After the first teacher-prefix epoch, SPEC-Distill switches from the offline teacher cache to student-cache self-reuse as described in Section 3.2. For implementation details beyond Table 3, the main GRPO, SPEC-Distill, Luffy, and OPD runs share a maximum prompt length of 1,024, learning rate 5×10^{-7}, PPO mini-batch size 256, and vLLM rollouts on 8 GPUs. GRPO and SPEC-Distill use the same verifier-driven GRPO update; SPEC-Distill only changes rollout construction through the offline teacher cache and speculative prefix acceptance. Luffy uses Qwen3-14B expert traces as off-policy targets with one expert prefix per group. OPD uses Qwen3-14B online to compute teacher log-probabilities, with token-level rollout correction threshold 5.0 and reverse-KL advantages; in the aligned OPD run, the KL-loss, KL-control, and entropy coefficients are set to 0.

A.2 Reward Training Dynamics

Figure 4 complements the held-out results in Table 1 by showing the verifier reward observed during RL training. Across student scales, SPEC-Distill generally matches or exceeds the GRPO reward trajectory while using the offline teacher-prefix pool only as rollout guidance and keeping the policy update verifier-driven. The separation is clearest on Qwen3-1.7B and Qwen3-4B. For Qwen3-8B, the curves are closer because the GRPO baseline already reaches high reward early in training and because the Qwen3-14B teacher is much closer in model scale to the 8B student than to the smaller students, leaving less additional teacher knowledge for the student to absorb. These traces should be interpreted as optimization dynamics rather than held-out accuracy.

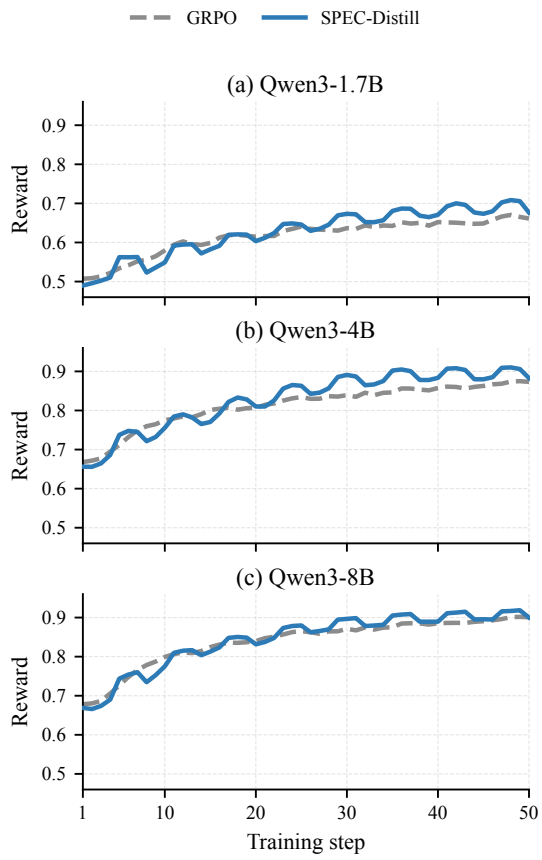


Figure 4: Training reward dynamics for GRPO and SPEC-Distill across Qwen3 student scales in non-think mode. Curves are smoothed with a centered three-step moving average for readability.

A.3 Zero-RL Distillation

Table 4 reports a complementary zero-RL check from the Qwen3-1.7B-Base checkpoint with a Qwen3-14B-Base teacher. Starting from the same base checkpoint, SPEC-Distill improves the four-benchmark average from 59.1 with GRPO to 63.8. It also reduces online rollout work from 554.8M to 189.0M tokens and rollout time from 7.75 to 3.63 hours, while using 53.4M offline teacher tokens to build the draft cache. These results suggest that teacher-prefix guidance can also help when starting from a base model, while our main comparisons remain focused on the non-think RLVR setting.

Method	Efficiency			Performance				
	Off-policy Tokens	On-policy Tokens	Rollout Time / Hours	GSM8K	MATH500	Olympiad	MMLU STEM	AVG
Student: *Qwen3-1.7B-Base*								
Base model	–	–	–	59.1	45.0	16.7	39.3	40.0
GRPO	–	554.8	7.75	82.0	65.2	28.3	60.7	59.1
SPEC-Distill	53.4	**189.0**	**3.63**	**85.1**	**68.6**	**32.4**	**68.9**	**63.8**

Table 4: Zero-RL distillation result on Qwen3-1.7B-Base. SPEC-Distill uses Qwen3-14B-Base as the offline teacher; token counts are in millions, and AVG averages the four reported benchmarks.

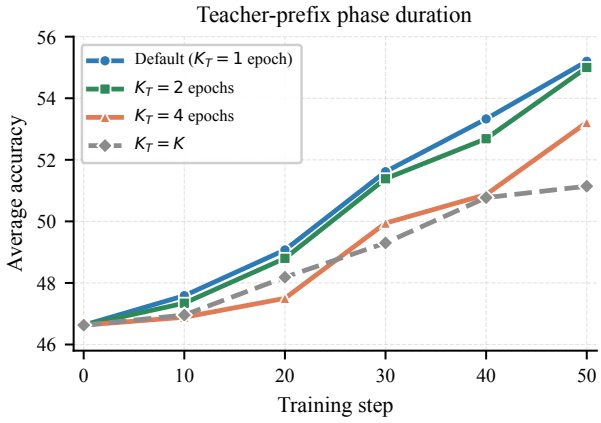


Figure 5: Step-level average accuracy when varying the teacher-prefix phase duration on Qwen3-1.7B with a Qwen3-14B teacher. The default source schedule samples drafts from $\mathcal{C}_T$ for one epoch and then switches to $\mathcal{C}_S$. Longer schedules keep $\mathcal{C}_T$ as the draft source for two epochs, four epochs, or all online RL steps.

A.4 Scale-wise Prefix Reuse Under a Fixed Teacher

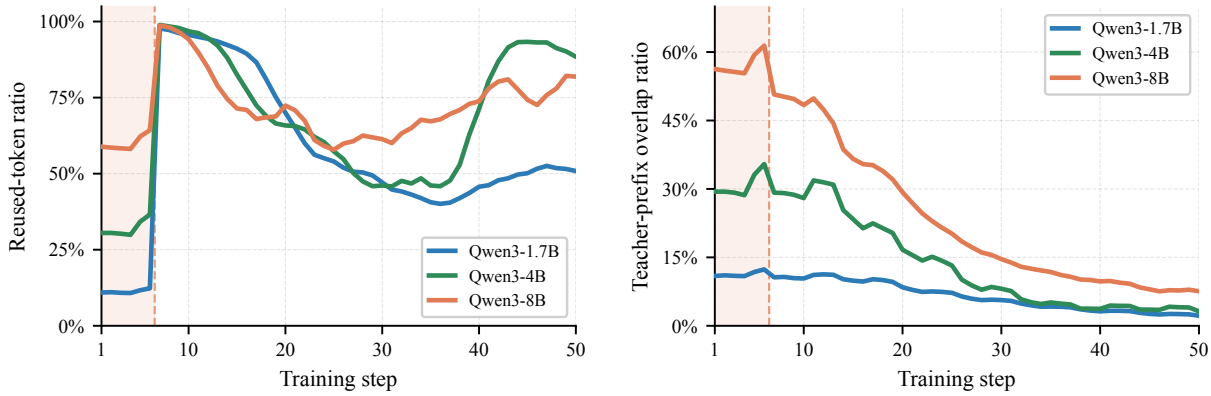
Figure 6 compares the three non-think Qwen3 students while holding the teacher fixed to Qwen3-14B. During the initial teacher-prefix epoch, larger students accept substantially longer teacher drafts: the average reused-token ratio is 11.4% for Qwen3-1.7B, 32.4% for Qwen3-4B, and 60.4% for Qwen3-8B. This indicates that teacher–student compatibility improves with scale inside the same model family: larger students place higher probability on the teacher's next tokens and therefore pass more speculative verification steps.

After the source switches to $\mathcal{C}_S$, the accepted-prefix ratio remains high for all students, but the exact teacher-prefix footprint separates the scales more clearly. The post-switch teacher-prefix overlap averages 6.4%, 12.8%, and 22.2% for Qwen3-1.7B, Qwen3-4B, and Qwen3-8B, respectively, before decaying in later steps. Thus, self-reuse is not simply copying the fixed teacher cache: all students increasingly move away from exact teacher trajectories, but larger students retain a stronger measurable trace of teacher-induced rollout structure under the same teacher. This scale effect complements the reward dynamics in Figure 4: the 8B student is closer to the teacher distribution and reuses more teacher-aligned prefixes, but the final accuracy gain is also more limited because the Qwen3-14B teacher is relatively close in scale to the Qwen3-8B student, leaving less additional teacher knowledge to transfer than for the smaller students.

A.5 Teacher-Prefix Phase Duration

In Figure 5 and Table 5, we further ablate the source schedule by varying K_T, the length of the teacher-prefix phase before switching from the offline teacher cache $\mathcal{C}_T$ to the student cache $\mathcal{C}_S$. This experiment complements the post-prefix reuse ablation in Section 4.4: there, all variants share the same one-epoch teacher-prefix phase and differ after the switch; here, all variants keep the same speculative acceptance rule and differ only in how long $\mathcal{C}_T$ remains the draft source before self-reuse begins. The default setting uses one full pass over DEEPMATH-6K, which both exposes the student to teacher-guided prefixes and gives the student cache enough completed rollouts to serve as the next-stage source.

The default one-epoch K_T schedule reaches the best final average accuracy (55.2), while using $\mathcal{C}_T$ for two epochs remains close (55.0) and longer teacher-prefix phases degrade performance (53.2 for four epochs and 51.1 when $K_T = K$). This pattern supports the teacher-to-self schedule used in the main experiments: the offline teacher cache is most useful as an early source of high-quality rollout structure, but repeatedly drawing drafts from the same static teacher cache throughout training can anchor proposals to a fixed distribution and reduce the student's opportunity to explore with its own improving policy. Importantly, the $K_T = K$ variant still does not query the teacher



(a) Accepted draft-token ratio.

(b) Exact teacher-prefix overlap.

Figure 6: Scale-wise prefix reuse under the same Qwen3-14B teacher. The shaded region marks the initial teacher-prefix epoch; after this point, SPEC-Distill switches the draft source from the offline teacher cache $\mathcal{C}_T$ to the student cache $\mathcal{C}_S$. **(a)** reports the fraction of response tokens supplied by accepted draft prefixes. **(b)** reports the exact longest-common-prefix overlap with the corresponding offline teacher trajectory, used as a conservative proxy for teacher-structure reuse.

online; it only keeps reusing the fixed offline cache throughout RL. The degradation therefore isolates a scheduling effect rather than a change in teacher compute: fixed teacher drafts are useful for bootstrapping exploration, but the later stages benefit from drafts produced by the evolving student distribution.

Teacher-prefix Phase	Step	AMC23	AIME24	AIME25	GSM8K	MATH500	Olympiad	MMLU STEM	AVG
Default ($K_T = 1$ epoch)	0	44.8	11.2	10.0	81.2	71.2	35.4	72.6	46.6
	10	43.1	13.2	10.9	80.9	73.6	38.4	73.0	47.6
	20	47.2	16.2	10.9	82.6	72.4	41.0	73.2	49.1
	30	54.2	17.6	13.0	83.5	76.6	43.3	73.1	51.6
	40	56.4	20.4	15.6	84.3	78.2	44.4	74.0	53.3
	50	60.0	24.4	17.9	85.7	78.4	46.4	73.6	55.2
$K_T = 2$ epochs	0	44.8	11.2	10.0	81.2	71.2	35.4	72.6	46.6
	10	44.7	11.7	10.9	81.3	73.8	36.9	72.1	47.3
	20	47.3	14.5	10.1	82.9	72.2	41.2	73.4	48.8
	30	51.6	17.4	13.5	84.6	77.6	41.0	74.0	51.4
	40	55.8	19.7	14.9	84.2	77.2	43.4	73.6	52.7
	50	59.2	22.6	17.6	84.3	80.0	46.4	74.9	55.0
$K_T = 4$ epochs	0	44.8	11.2	10.0	81.2	71.2	35.4	72.6	46.6
	10	45.5	12.2	9.4	81.4	72.2	36.6	70.9	46.9
	20	43.9	13.3	9.8	81.3	71.8	38.5	73.9	47.5
	30	49.1	15.6	12.7	82.6	74.6	41.9	73.1	49.9
	40	51.1	17.0	13.2	82.0	76.6	43.1	73.1	50.9
	50	56.4	21.8	15.8	84.1	77.2	44.3	72.9	53.2
$K_T = K$	0	44.8	11.2	10.0	81.2	71.2	35.4	72.6	46.6
	10	42.8	11.2	9.9	82.0	71.6	38.7	72.5	47.0
	20	46.1	14.9	10.9	81.3	73.6	38.2	72.3	48.2
	30	50.0	14.6	11.1	83.0	73.8	40.4	72.2	49.3
	40	52.7	16.1	13.2	84.1	76.6	40.6	72.1	50.8
	50	50.3	18.3	15.1	83.2	76.8	41.6	72.7	51.1

Table 5: Step-level results for the source-schedule ablation that varies the teacher-prefix phase duration on Qwen3-1.7B with a Qwen3-14B teacher. K_T follows Algorithm 1: during the first K_T optimization steps, drafts are sampled from the offline teacher cache $\mathcal{C}_T$ before the schedule switches to the student cache $\mathcal{C}_S$. We report K_T in equivalent epochs over DEEPMATH-6K; $K_T = K$ keeps using $\mathcal{C}_T$ for all online RL steps and never switches to student-cache self-reuse. AVG averages the seven reported benchmark columns.