RIVAD: REASONING WITH INFORMATION-VARIANT VIEWS FOR ADAPTIVE MULTI-MODE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibit strong multi-step reasoning capabilities, yet transferring these abilities to compact models remains challenging. Existing methods typically rely on either teacher-generated trajectories, which provide stable supervision but may not reflect the prefixes encountered by the student at inference, or student-generated trajectories, which reduce this mismatch at higher training cost. We introduce RIVAD, a reasoning distillation framework that combines information-variant self-distillation with adaptive multi-mode training. Training integrates three complementary modes: canonical off-policy distillation from an external teacher, privileged-context self-distillation, and on-policy distillation over fresh student-generated responses. In the self-distillation mode, RIVAD constructs information-variant views by exposing a self-teacher to different partial, order-preserving subsets of the canonical reasoning, while the student receives only the original question. An adaptive training strategy dynamically adjusts the contribution of different supervision modes according to the student's learning progress, allowing the training allocation to evolve throughout optimization. Together, these components combine stable canonical targets, information-variant privileged guidance, and feedback on student-visited prefixes in a unified framework for distilling multi-step reasoning into compact language models.

1 INTRODUCTION

Knowledge distillation (KD) provides an effective way to transfer the capabilities of large language models (LLMs) into smaller and more deployable models (Hinton et al., 2015). In the LLM era, a substantial line of work has explored *black-box distillation*, where students learn from responses or reasoning traces generated by stronger models without access to their internal predictive distributions (Luo et al., 2025; Liu et al., 2026; Yang et al., 2026b). In contrast, *white-box distillation* directly exploits teacher predictive distributions, providing substantially finer-grained supervision. Recent approaches have improved white-box distillation through more effective divergence objectives (Gu et al., 2024; Ko et al., 2024; 2025; Kim et al., 2026; Shin et al., 2026), token-wise divergence control (Jung et al., 2025), and module-aware supervision (Yang et al., 2026a). Despite these advances, existing methods typically adopt a fixed or manually scheduled source of supervision, leaving relatively underexplored how different distillation modes should be allocated as the student evolves.

A key challenge is that the relative usefulness of different supervision sources may change throughout training. Off-policy distillation relies on teacher-generated trajectories, providing stable and high-quality supervision but potentially exposing the student to states that differ from those induced by its own policy. On-policy distillation instead queries the teacher on student-generated trajectories, improving alignment with the student's inference distribution (Agarwal et al., 2024), but introduces additional generation and teacher-evaluation costs and may be less informative when the student is still weak. Self-distillation provides a complementary, student-relevant source of supervision (Zhao et al., 2026), but may reinforce the student's existing errors or biases. Early in training, teacher-generated trajectories can provide more reliable supervision when the student's reasoning is still underdeveloped, whereas student-induced trajectories may become increasingly relevant as the student improves. These modes therefore offer different forms of supervision whose relative utility can vary over training, motivating an *adaptive multi-mode distillation* strategy rather than a fixed mixture or predetermined training schedule.

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.

For self-distillation, we consider privileged context as a way to construct a stronger supervision signal from the student itself. A privileged self-teacher conditions on information unavailable to the student, while the student continues to reason from the original question alone (Zhao et al., 2026). However, recent evidence shows that on-policy self-distillation on long-CoT models can provide only short-lived gains and may even degrade reasoning performance as training proceeds (Shen et al., 2026), suggesting that self-distillation should be treated as a state-dependent signal rather than a uniformly beneficial objective throughout training. We instantiate this signal using the structure of the reference response by constructing partial privileged contexts from subsets of intermediate reasoning steps while preserving their original order. These partial views provide the self-teacher with additional reasoning context, whereas the student always conditions only on the original question, yielding a structured privileged-context signal that can be adaptively incorporated into our multi-mode training framework.

To address these challenges, we propose **Reasoning Inter-Step Relation Distillation with Adaptive Multi-Mode Training (RIVAD)**, a unified white-box distillation framework that jointly considers *where supervision comes from* and *how reference reasoning is exposed as privileged context*. RIVAD integrates three complementary supervision modes: off-policy distillation on canonical responses, on-policy distillation over student-generated responses, and privileged-context self-distillation from a student-derived self-teacher. Rather than using fixed proportions or a predetermined transition between supervision modes, RIVAD periodically updates their selection probabilities according to the student's current learning state, allowing the supervision source to evolve throughout training. Within the self-distillation mode, the canonical response is segmented into reasoning steps, and the self-teacher is conditioned on partial, order-preserving views of these steps while the student continues to receive only the original question. Figure 1 illustrates the overall RIVAD framework and its adaptive routing mechanism.

Our main contributions are summarized as follows:

- We propose **RIVAD**, a unified white-box distillation framework that combines off-policy, on-policy, and self-distillation and adaptively selects among these complementary supervision modes throughout training.
- We introduce a privileged-context self-distillation mode in which a student-derived self-teacher is conditioned on partial, order-preserving views of the canonical reasoning, providing student-compatible supervision while the student itself receives only the original question.
- Experiments across multiple reasoning benchmarks and student model scales demonstrate that RIVAD consistently improves over strong white-box distillation baselines, with ablations validating the effectiveness of adaptive multi-mode distillation

2 Related Work

Knowledge Distillation Objectives. Knowledge distillation has long been used to compress autoregressive language models, ranging from token-level distribution matching (Hinton et al., 2015; Sanh et al., 2019) and sequence-level distillation (Kim & Rush, 2016) to rationale-based transfer for reasoning models (Hsieh et al., 2023; Magister et al., 2023). Recent white-box LLM distillation has increasingly focused on improving the form of the supervision signal. MiniLLM (Gu et al., 2024) replaces conventional forward KL with reverse KL, while DistiLLM (Ko et al., 2024) introduces skew KL together with an efficient adaptive off-policy strategy. DistiLLM-2 (Ko et al., 2025) further couples the source of generated data with a contrastive objective. More recent methods explore alternative forms of distributional supervision: TSD-KD (Kim & Baek, 2026) combines token-selective direct distillation with indirect preference feedback, CSD (Kim et al., 2026) aligns relative logit structure through concrete score matching, and AMiD (Shin et al., 2026) introduces a family of α-mixture assistant distributions to bridge the teacher–student capacity gap. These methods primarily refine what information is transferred or how teacher and student predictions are matched; RIVAD instead retains standard forward KL and focuses on the supervision configuration under which this signal is obtained.

On-Policy and Trajectory-Aware Distillation. Beyond the distillation objective itself, another line of work studies the trajectory on which teacher supervision is applied. ImitKD (Lin et al., 2020)

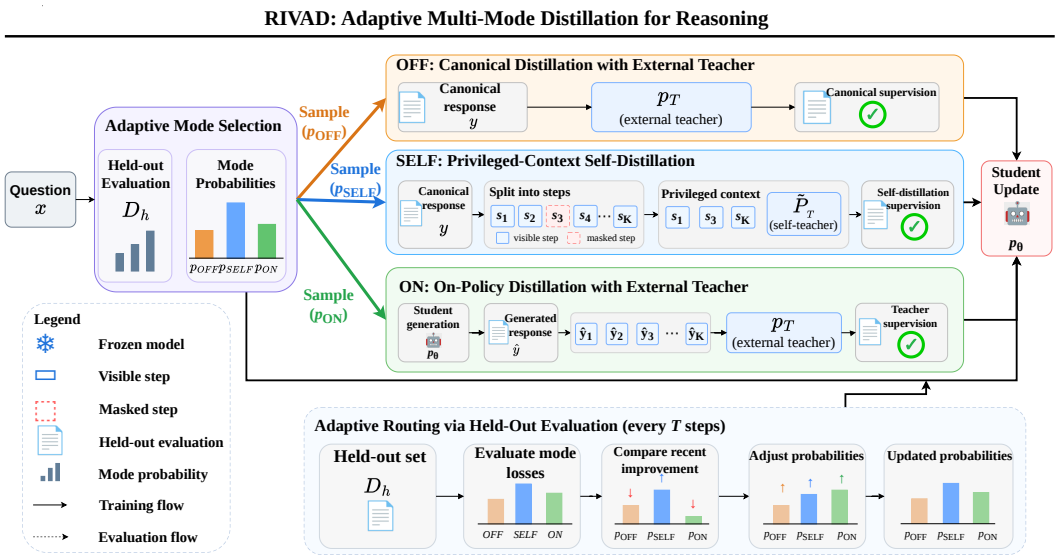


Figure 1: **Overview of RIVAD.** The method adaptively selects among OFF, SELF, and ON distillation modes, and uses held-out evaluation to adjust mode probabilities, enabling dynamic allocation of training updates for effective reasoning distillation.

casts autoregressive distillation as imitation learning and exposes the student to states induced by its own predictions, while GKD (Agarwal et al., 2024) generalizes distribution matching to student-generated sequences. DistiLLM (Ko et al., 2024) adaptively incorporates student-generated outputs to reduce the cost of on-policy training, and Speculative KD (Xu et al., 2025) constructs trajectories through interleaved student proposals and teacher corrections. DistiLLM-2 (Ko et al., 2025) further shows that teacher- and student-generated samples can benefit from different training objectives. Collectively, these studies establish trajectory source as an important design dimension alongside the divergence being optimized. RIVAD extends this design space by treating off-policy, on-policy, and self-distilled supervision as distinct training configurations and adaptively allocating updates among them according to observed learning progress.

Privileged-Context and Self-Distillation. A complementary direction studies distillation with information available only to the teacher during training. Generalized distillation formalizes this setting through privileged information (Lopez-Paz et al., 2016). For LLM reasoning, Self-Distilled Reasoner (Zhao et al., 2026) instantiates this principle through on-policy self-distillation, where a teacher view of the same model conditions on a reference solution while the student observes only the question. Recent work further shows that privileged self-distillation is sensitive to how the teacher context is constructed: it can degrade long-chain-of-thought reasoning under unfavorable supervision (Kaur et al., 2026), while Purified OPSD (Shen et al., 2026) separates reference-induced supervision from the question-conditioned signal. Other analyses compare alternative privileged reasoning views (Zhang et al., 2026) and study which forms of privileged context are most effective for a self-teacher (Tian et al., 2026). These findings motivate treating privileged information as a configurable supervision source rather than a fixed complete solution. RIVAD constructs information-variant self-distillation from partial, order-preserving views of canonical reasoning and integrates this configuration with external-teacher off-policy and on-policy distillation through a common adaptive routing framework.

3 Method

3.1 Problem Setup and Output Distillation

Let $\mathcal{D} = \{(x, y)\}$ be a reasoning dataset, where x is a question and y its canonical response. We train a student model p_θ using three complementary distillation modes, $m \in \{\text{OFF}, \text{ON}, \text{SELF}\}$.

At each optimizer update, one mode is selected for training. The three modes instantiate different teacher–trajectory configurations while sharing the same output-level distillation objective. Their precise definitions are given in Section 3.2.

We consider an external teacher p_T and a self-teacher derived from the student $p_{\widetilde{T}}$. The external teacher remains frozen throughout training. The self-teacher is initialized from the student, receives no gradients, and remains fixed between routing checkpoints. After each held-out routing evaluation, it is refreshed as $p_{\widetilde{T}} \leftarrow p_\theta$. All models share the same output vocabulary for token-level distillation, although their hidden dimensions need not match.

For the selected mode m, let r denote the response to score, $q \in \{p_T, p_{\widetilde{T}}\}$ the teacher providing the target distribution, and $\widetilde{x}$ the mode-specific teacher context. The student always receives only the original question x. At response position i, the student and teacher predictive distributions are

$$p_i = p_\theta(\cdot \mid x, r_{<i}), \qquad q_i = q(\cdot \mid \widetilde{x}, r_{<i}). \tag{1}$$

Under teacher forcing, the teacher and student evaluate the same response sequence r, so their token-level predictions are aligned at each response position. Although their conditioning contexts may differ across modes, both models predict the token at position i given the same response prefix $r_{<i}$. We then apply temperature τ to q_i and p_i and optimize the standard forward KL objective

$$\mathcal{L}_{\mathrm{KD}} = \frac{\tau^2}{N} \sum_{i=1}^{N} D_{\mathrm{KL}}(q_i \,\|\, p_i), \tag{2}$$

where N is the number of tokens in the selected response. In implementation, we distill over the teacher's top-k logits and renormalize both distributions on this shared support.

3.2 Complementary Distillation Modes

The three modes differ in the trajectory on which distillation is performed and in the source of the teacher target, as summarized in Table 1.

Table 1: Properties of each distillation mode

Mode	Response r	Teacher q	Teacher context $\widetilde{x}$
OFF	Canonical y	External teacher p_T	x
ON	Fresh student $\hat{y}$	External teacher p_T	x
SELF	Canonical y	Self-teacher $p_{\widetilde{T}}$	$x \oplus \mathrm{Template}(c_h)$

Canonical off-policy distillation (OFF). OFF performs distillation on the canonical response from the training dataset. Accordingly, we set $r = y$, $q = p_T$, and $\widetilde{x} = x$, such that the student and the frozen external teacher score the same canonical trajectory conditioned on the original question.

Fresh on-policy distillation (ON). In ON, the supervised response is generated by the current student rather than taken from the canonical dataset response. Specifically, we sample a fresh rollout $r = \hat{y} \sim p_\theta(\cdot \mid x)$ and set $q = p_T$ and $\widetilde{x} = x$. The frozen external teacher and the student then score the same student-generated trajectory under teacher forcing. Since a new rollout is sampled from the current policy, the supervised trajectory changes as the student evolves during training.

Privileged-context self-distillation (SELF). In SELF, the student is distilled on the canonical response y, but the target distribution is provided by the self-teacher. Accordingly, we set $r = y$ and $q = p_{\widetilde{T}}$. The student receives only the original question x, whereas the self-teacher additionally receives a partial view of the canonical reasoning as privileged context.

For SELF only, we split the canonical response y into an ordered sequence of reasoning steps $z_1, \ldots, z_L$. We construct the privileged-context by randomly hiding a subset of steps. For $L \geq 2$, we first sample a drop rate $d \sim \mathcal{U}(0, d_{\max})$, and then independently sample

$$h_j \sim \mathrm{Bernoulli}(d), \qquad j = 1, \ldots, L, \tag{3}$$

where $h_j = 1$ indicates that reasoning step z_j is hidden from the self-teacher context. We ensure that at least one step is hidden and at least one step remains visible; if all steps receive the same mask value, one step is randomly flipped. The visible steps are concatenated in their original order to form

$$c_h = \bigoplus_{j:h_j=0} z_j, \qquad \widetilde{x} = x \oplus \mathrm{Template}(c_h). \tag{4}$$

Thus, different masks expose the self-teacher to different partial views of the same canonical reasoning while the student continues to condition only on x. If fewer than two complete reasoning steps are available, we use no privileged reasoning context. Context truncation, when required, is performed only at complete step boundaries.

Unlike conventional self-distillation that relies on student-generated responses (Zhao et al., 2026), our SELF mode keeps the canonical trajectory fixed and instead strengthens the self-teacher through privileged reasoning context. This avoids directly training on potentially noisy student-generated trajectories while still allowing the target to evolve with the student. Randomly masking reasoning steps creates multiple partial conditioning views and prevents the self-teacher from observing the complete solution. As a result, SELF provides a context-variant yet trajectory-controlled form of self-distillation.

3.3 Adaptive Multi-Mode Selection

The three modes provide complementary forms of supervision, but their relative utility can change as the student evolves. OFF provides stable supervision on canonical trajectories, ON follows states visited by the current student policy, and SELF relies on an evolving self-teacher. We therefore adapt the allocation of training updates according to the student's observed progress under each mode. The full routing update is summarized in Algorithm 1 in Appendix B.

Mode sampling. We initialize $\rho^{(1)}_{\mathrm{ON}}$ and $\rho^{(1)}_{\mathrm{SELF}}$ so that OFF receives the largest initial probability, with

$$\rho^{(1)}_{\mathrm{OFF}} = 1 - \rho^{(1)}_{\mathrm{ON}} - \rho^{(1)}_{\mathrm{SELF}}. \tag{5}$$

Routing interval $t \geq 1$ spans the training updates between checkpoints $t-1$ and t. Throughout this interval, the sampling probabilities remain fixed at $\rho^{(t)}$. At each optimizer update within the interval, rank zero samples a single distillation mode from a categorical distribution over $\mathrm{OFF}, \mathrm{ON}, \mathrm{SELF}$:

$$m \sim \mathrm{Categorical}\left(\rho^{(t)}_{\mathrm{OFF}}, \rho^{(t)}_{\mathrm{ON}}, \rho^{(t)}_{\mathrm{SELF}}\right). \tag{6}$$

and broadcasts the result to all distributed workers. All gradient-accumulation microbatches within that update use the selected mode. This gives each update a single teacher–trajectory configuration and requires evaluating only the selected training objective.

Held-out routing signal. Let $\mathcal{D}_h$ be a fixed held-out routing set, disjoint from the training data and never used for gradient updates. At routing checkpoint t, we evaluate SELF and ON separately on $\mathcal{D}_h$. For $m \in \{\mathrm{SELF}, \mathrm{ON}\}$, let $D^{(t)}_m$ denote the held-out distillation loss obtained by applying the same forward-KL objective as in Equation 2 under mode m. SELF uses the canonical responses with a fixed context mask for each held-out example and the frozen self-teacher, whereas ON uses deterministic responses freshly generated by the current student and scored by the external teacher. Only the output-level distillation loss is used for routing; other training objectives do not enter the routing signal.

Let $\rho^{(t)}_m$ denote the sampling probability used during routing interval t, initialized from the probabilities above. Starting from $t = 1$, we measure the relative improvement of mode m between two consecutive checkpoints as

$$I^{(t)}_m = \frac{D^{(t-1)}_m - D^{(t)}_m}{D^{(t-1)}_m + \epsilon_a}, \qquad m \in \{\mathrm{SELF}, \mathrm{ON}\}. \tag{7}$$

Progress-aware probability update. After checkpoint $t \geq 1$, we increase the allocation of mode m when its relative progress falls below a target δ:

$$\rho_m^{(t+1)} = \begin{cases} \min\left\{\rho_m^{(t)} + \eta_m, \rho_m^{\max}\right\}, & I_m^{(t)} < \delta, \\ \rho_m^{(t)}, & \text{otherwise}, \end{cases} \qquad m \in \{\text{SELF}, \text{ON}\}. \tag{8}$$

Here $\eta_m > 0$ is the allocation increment. ON and SELF are updated independently, and OFF receives the residual probability,

$$\rho_{\text{OFF}}^{(t+1)} = 1 - \rho_{\text{ON}}^{(t+1)} - \rho_{\text{SELF}}^{(t+1)}. \tag{9}$$

We require $0 \leq \rho_m^{(1)} \leq \rho_m^{\max}$ and $\rho_{\text{ON}}^{\max} + \rho_{\text{SELF}}^{\max} \leq 1$, ensuring a valid distribution at every checkpoint. The caps bound how much budget can be transferred from OFF. The resulting schedule is monotone in ON and SELF allocation, while the timing of each increase depends on observed progress rather than a prescribed training phase.

While OFF dominates initially, its updates may also reduce ON and SELF losses through transfer. Their probabilities need not increase as long as the observed progress remains sufficient. When progress slows, the scheduler allocates more direct supervision to the corresponding mode. This is a progress-triggered rule: it does not impose a separate competence threshold or require ON and SELF to become useful at the same checkpoint.

Full objective and training. For a selected mode m, let $r^m = (r_1^m, \ldots, r_{N_m}^m)$ denote the mode-specific response defined in Section 3.2. For OFF and SELF, we combine the forward-KL objective in Equation 2 with the standard language-model loss

$$\mathcal{L}_{\text{CE}}^m = -\frac{1}{N_m} \sum_{i=1}^{N_m} \log p_\theta \left(r_i^m \mid x, r_{<i}^m\right). \tag{10}$$

ON does not use the CE term and is optimized only with the KD objective. The overall objective is therefore

$$\mathcal{L}^m = \begin{cases} (1-\beta)\mathcal{L}_{\text{CE}}^m + \beta \mathcal{L}_{\text{KD}}^m, & m \in \{\text{OFF}, \text{SELF}\}, \\ \mathcal{L}_{\text{KD}}^m, & m = \text{ON}, \end{cases} \qquad \beta \in [0,1]. \tag{11}$$

The three modes otherwise differ only in the response, teacher, and teacher context used to construct their supervision. At each optimizer update, one mode is sampled according to the current routing probabilities, and only the student parameters θ are updated, while both the external teacher and self-teacher remain gradient-free. At routing checkpoints, the held-out improvements of SELF and ON are computed using Equation 7, their sampling probabilities are updated according to Equation 8, and the self-teacher is synchronized with the current student. The complete training procedure is provided in Appendix 2.

3.4 Theoretical Justification

We provide a simple analysis of how repeated mode sampling affects the mode-specific discrepancy. At the iteration t, we denote $m(t)$ as the selected mode.

For a target mode m, let D_t^m denote its discrepancy after update t. When mode m is selected, the observed progress signal motivates the local contraction

$$D_t^m \leq D_{t-1}^m \exp\left(-I_m^{(t)}\right), \qquad m(t) = m, \tag{12}$$

where $I_m^{(t)}$ is the relative improvement defined in Equation 7.

The model is updated by

$$\theta_{t+1} = \theta_t - \eta g_t^{m(t)},$$

where $\eta > 0$ is the learning rate and $g_t^{m(t)} = \nabla \mathcal{L}_{\text{train}}^{m(t)}(\theta_t)$.

Theorem 1 (Cumulative routing dynamics)**.** *Suppose that all gradient norms are upper-bounded by G and $D_t^m \geq a$ for all t and m, we have*

$$D_T^m \leq D_0^m \exp\left(-\sum_{t=1}^{T} \mathbf{1}\{m(t) = m\} I_m^{(t)} + \frac{\eta G^2}{a} \sum_{t=1}^{T} \mathbf{1}\{m(t) \neq m\} \right). \tag{13}$$

Theorem 1 separates the evolution of D^m into two competing terms: cumulative progress from updates that directly optimize mode m, and accumulated perturbation from updates assigned to other modes. This provides the rationale for the progress-aware rule in Equation 8. When the observed progress $I_m^{(t)}$ becomes insufficient, increasing ρ_m raises the frequency of direct updates and correspondingly reduces the probability mass assigned to potentially interfering updates. Conversely, while the current allocation already provides sufficient progress, no additional routing budget is required. The proof and the local justification of Equation 13 are given in Appendix A.

4 Experiments

4.1 Experimental Settings

Models. Our primary experiments distill Qwen2.5-14B-Instruct into Qwen2.5-1.5B-Instruct (Qwen Team, 2024). To evaluate generalization between model families, we also consider Gemma2-9B-Instruct as the teacher and Gemma2-2B-Instruct as the student (Gemma Team, 2024).

Baselines. We benchmark our method against two groups of knowledge-distillation methods. **On-Policy Methods:** We include DistiLLM (Ko et al., 2024), MiniLLM (Gu et al., 2024), GKD (Agarwal et al., 2024) with $\beta = 0.9$, Speculative KD (Xu et al., 2025), TSD-KD (Kim & Baek, 2026), AMiD (Shin et al., 2026), and CSD (Kim et al., 2026). These methods train the student using student-generated responses. **Off-Policy Methods:** We use classical supervised KD (Hinton et al., 2015) and sequence-level KD (Kim & Rush, 2016), both of which learn from a fixed collection of teacher-generated responses.

Training and Evaluation. We train on approximately 78K UltraInteract instances (Yuan et al., 2024), following the data setup used by TSD-KD (Kim & Baek, 2026). On-policy methods use student-generated responses to prompts sampled from UltraInteract, whereas off-policy methods use teacher-generated responses to the same prompts. We evaluate all models on a broad collection of reasoning benchmarks. Mathematical reasoning is measured on GSM8K (Cobbe et al., 2021), GSM-Plus (Li et al., 2024), Minerva Math (Lewkowycz et al., 2022), MATH (Hendrycks et al., 2021b), and MMLU-Pro Math (Wang et al., 2024). Scientific and STEM reasoning is evaluated on MMLU-STEM (Hendrycks et al., 2021a) and SciQ (Welbl et al., 2017); program synthesis on MBPP (Austin et al., 2021); and broad reasoning on BBH (Suzgun et al., 2023). We report accuracy using the standard evaluation protocol of each benchmark. Additional implementation details and default configurations are provided in Table 5 in Appendix B, while additional experimental results are presented in Appendix D.

4.2 Main Results and Analysis

Table 2 reports the main results for the Qwen2.5-14B-Instruct $\rightarrow$ Qwen2.5-1.5B-Instruct setting, while results for the additional Gemma2 teacher–student pair are reported in Table 6. For the prior baselines, we adopt the results reported by TSD-KD (Kim & Baek, 2026), which does not report standard deviations or standard errors. We independently evaluate CSD and AMiD under our evaluation setup and report their corresponding standard errors. Results for our method, RIVAD, are obtained from our own experiments, with standard errors reported for the completed runs.

Overall performance. RIVAD achieves the highest average accuracy of 55.5%, compared with 49.5% for the strongest prior baseline, CSD, corresponding to a 12.1% relative improvement. It also improves substantially over the initial student (44.6%), yielding a 24.4% relative gain in average accuracy. Among the evaluated student methods, RIVAD obtains the best performance on six of the

Table 2: Performance comparison on Qwen2.5-14B-Instruct (Teacher) → Qwen2.5-1.5B-Instruct (Student). STEM denotes MMLU-STEM, and Pro-Math denotes MMLU-Pro-Math. Numbers in parentheses denote standard errors.

Method	GSM8K	GSM-Plus	MATH	MBPP	SciQ	STEM	Pro-Math	BBH-COT	Avg.
Qwen2.5-14B-Instruct	80.3	59.7	21.7	78.9	86.4	70.5	77.6	60.8	67.0
Qwen2.5-1.5B-Instruct	57.1	38.8	16.9	38.4	85.5	49.5	34.4	36.5	44.6
Sequence-Level KD	56.5	38.2	16.5	40.6	87.9	48.9	32.6	36.6	44.7
Supervised-KD	56.3	37.8	17.9	41.0	86.6	48.4	35.3	36.4	45.0
DistiLLM	57.2	38.7	18.6	42.1	85.7	47.8	35.2	36.5	45.2
Speculative KD	58.0	39.2	18.5	41.6	86.7	47.8	34.8	37.2	45.5
MiniLLM	57.7	39.7	17.8	42.2	84.1	48.2	36.2	36.0	45.2
GKD ($\beta = 0.9$)	57.9	39.9	18.1	41.8	86.6	47.7	35.7	36.2	45.5
TSD-KD	60.1	40.5	26.1	42.1	**93.0**	50.0	36.9	40.2	48.6
CSD	$61.5_{\pm 1.34}$	$41.5_{\pm 0.48}$	$29.3_{\pm 0.61}$	$43.8_{\pm 2.22}$	$81.9_{\pm 1.22}$	$51.3_{\pm 0.86}$	$44.4_{\pm 1.35}$	$\mathbf{42.1}_{\pm 0.54}$	49.5
AMiD	$61.1_{\pm 1.0}$	$42.1_{\pm 0.45}$	$28.3_{\pm 0.55}$	$43.6_{\pm 2.25}$	$86.7_{\pm 1.17}$	$51.6_{\pm 0.96}$	$40.0_{\pm 1.13}$	$42.0_{\pm 0.64}$	49.4
RIVAD (Ours)	$\mathbf{68.8}_{\pm 1.27}$	$\mathbf{50.2}_{\pm 0.49}$	$\mathbf{53.0}_{\pm 0.67}$	$\mathbf{49.2}_{\pm 2.24}$	$82.4_{\pm 1.20}$	$\mathbf{52.6}_{\pm 0.86}$	$\mathbf{45.5}_{\pm 1.35}$	$42.0_{\pm 0.54}$	**55.5**

eight benchmarks. In comparison, recent strong distillation approaches such as TSD-KD, CSD, and AMiD achieve average accuracies of 48.6%, 49.5%, and 49.4%, respectively. These results indicate that RIVAD provides a substantial overall improvement over both conventional and recent white-box distillation baselines.

Pronounced gains on mathematical reasoning. The largest improvements are concentrated on mathematical reasoning tasks. Relative to the strongest prior result on each benchmark, RIVAD improves GSM8K from 61.5% to 68.8% (11.9% relative improvement), GSM-Plus from 42.1% to 50.2% (19.2%), and MATH from 29.3% to 53.0% (80.9%). RIVAD also improves MBPP from 43.8% to 49.2%, corresponding to a 12.3% relative gain. By contrast, the gains on MMLU-STEM and MMLU-Pro-Math are more modest, at 1.9% and 2.5% relative to their strongest prior baselines. This pattern suggests that the benefits of RIVAD are particularly pronounced on tasks that require extended mathematical and multi-step reasoning.

Task-dependent behavior. The improvement is not uniform across all benchmarks. On SciQ, RIVAD reaches 82.4%, below the 93.0% obtained by TSD-KD, while on BBH-COT it performs comparably to CSD (42.0% vs. 42.1%). Together with the substantially larger gains on GSM8K, GSM-Plus, and MATH, these results suggest that RIVAD primarily improves complex multi-step reasoning rather than uniformly increasing performance across all task categories.

4.3 Ablation study

Cross-Model-Family Generalization. To examine whether the proposed framework is specific to the Qwen model family, we further evaluate it on Gemma. The results in Table 6 show that the proposed method remains effective across a different model architecture and pretraining family, with consistent improvements over the corresponding baselines. This suggests that the benefit of adaptive multi-mode distillation is not tied to a particular student family, but can transfer across heterogeneous model families.

Contribution of different distillation modes Tables 3 and 4 show that the effectiveness of ON- and OFF-policy supervision varies across models and tasks. On Qwen, ON performs slightly better than OFF when used alone (41.12 vs. 40.28), while the two modes achieve comparable performance on Gemma (35.1 vs. 34.6). In contrast, adding SELF consistently improves both policy-based configurations: the average score increases from 40.28 to 54.12 for OFF + SELF and from 41.12 to 55.58 for ON + SELF on Qwen, and from 34.6 to 39.9 and 35.1 to 37.0 on Gemma, respectively. This suggests that SELF provides complementary supervision to both ON- and OFF-policy signals. The full multi-mode comparison further shows the benefit of adaptive mode combination. On Qwen, RIVAD achieves 54.76, slightly outperforming uniform mixing of the three modes (54.38), while achieving the best results on MBPP and STEM. On Gemma, RIVAD reaches 40.0, substantially

Table 3: Ablation study on the contribution of different distillation modes using Qwen.

Training Strategy	GSM8K	GSM-Plus	MATH	MBPP	STEM	Avg.
SELF-only baseline						
SELF	45.4	33.6	28.8	48.8	51.1	41.54
OFF-policy comparison						
OFF	56.3	37.8	17.9	41.0	48.4	40.28
OFF + SELF	68.5	50.0	52.6	47.6	52.0	54.12
ON-policy comparison						
ON	57.7	39.7	17.8	42.2	48.2	41.12
ON + SELF	**71.8**	**52.3**	**53.4**	48.2	52.2	**55.58**
Full multi-mode comparison						
ON + OFF	57.2	38.7	18.6	42.1	47.8	40.88
Uniform $(1/3,\ 1/3,\ 1/3)$	70.0	50.3	52.4	47.4	51.8	54.38
ON + OFF + SELF (RIVAD)	68.8	50.2	53.0	**49.2**	**52.6**	54.76

Table 4: Ablation study on the contribution of different distillation modes using Gemma.

Training Strategy	GSM8K	GSM-Plus	MATH	MBPP	STEM	Avg.
SELF-only baseline						
SELF	5.23	5.99	6.38	28.6	21.3	13.5
OFF-policy comparison						
OFF	53.6	36.8	17.9	39.0	25.5	34.6
OFF + SELF	**55.0**	**37.6**	23.7	38.4	44.7	39.9
ON-policy comparison						
ON	54.4	36.8	17.5	**40.6**	45.9	35.1
ON + SELF	50.8	34.3	21.8	38.0	40.1	37.0
Full multi-mode comparison						
ON + OFF	54.3	35.8	17.8	39.6	25.2	34.5
Uniform $(1/3,\ 1/3,\ 1/3)$	54.1	36.5	22.8	39.1	44.8	39.5
ON + OFF + SELF (RIVAD)	54.8	37.0	**24.1**	38.0	**46.1**	**40.1**

improving over the ON + OFF configuration (34.5). Overall, these results suggest that while the relative contribution of ON and OFF depends on the model, combining them with SELF and adaptively balancing their contributions provides a more robust distillation strategy than relying on a fixed or single mode.

5 Conclusion

In this work, we introduced RIVAD, a white-box reasoning distillation framework that adaptively coordinates multiple sources of supervision throughout training. RIVAD combines canonical off-policy distillation from an external teacher, on-policy distillation over fresh student responses, and privileged-context self-distillation, in which a lagged self-teacher is provided with partial, order-preserving views of reference reasoning while the student conditions only on the original question. Rather than relying on a fixed mixture of these supervision modes, RIVAD dynamically adjusts their training exposure according to the student's evolving learning progress. Experiments across Qwen2.5 and Gemma2 teacher–student settings demonstrate consistent improvements over prior distillation baselines. In the Qwen2.5-14B $\rightarrow$ Qwen2.5-1.5B setting, RIVAD achieves an average accuracy of 55.5, improving over the strongest prior baseline by 6.0 points, with particularly large gains on mathematical reasoning benchmarks. These results highlight the importance of adapting not only the amount of supervision during distillation, but also its source and information content as the student evolves.

AI use statement

Generative AI was used to assist manuscript drafting, literature discovery, and checking the correspondence between the method description and implementation. The method also uses a teacher language model to generate supporting context for training. *Before submission, complete this disclosure*

with other AI-assisted research activities, if any, and describe the verification actually performed by the authors.

REPRODUCIBILITY STATEMENT

The method, routing algorithm, and current configuration are described in Section 3 and Appendix B. The experimental record additionally includes data provenance, exact model and evaluation revisions, run configurations, and random seeds. Our code will be released upon acceptance.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. doi: 10.48550/arXiv.2108.07732. URL https://arxiv.org/abs/2108.07732.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. doi: 10.48550/arXiv.2408.00118. URL https://arxiv.org/abs/2408.00118.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander 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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. ToDi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8078–8091. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.409. URL https://aclanthology.org/2025.emnlp-main.409/.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026. doi: 10.48550/arXiv.2607.05184. URL https://arxiv.org/abs/2607.05184.

Minsang Kim and Seung Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a89331e2ecbbd9d9618bc94717a91155-Abstract-Conference.html.

Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-Chul Moon. Distillation of large language models via concrete score matching. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1bfc9f74afa91b9b8add5a97a97001a1-Abstract-Conference.html.

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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3083–3096, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-Plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.163. URL https://aclanthology.org/2024.acl-long.163/.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18041–18059. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.822. URL https://aclanthology.org/2026.acl-long.822/.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Yijia Luo, Yulin Song, Xingyao Zhang, Jiaheng Liu, Weixun Wang, GengRu Chen, Wenbo Su, and Bo Zheng. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long CoT distillation. *arXiv preprint arXiv:2503.16385*, 2025. doi: 10.48550/arXiv.2503.16385. URL https://arxiv.org/abs/2503.16385.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-short.151. URL https://aclanthology.org/2023.acl-short.151/.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. URL https://arxiv.org/abs/1910.01108.

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, et al. Purified opsd: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026.

Donghyeok Shin, Yeongmin Kim, Suhyeon Jo, Byeonghu Na, and Il-Chul Moon. AMiD: Knowledge distillation for LLMs with α-mixture assistant distribution. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/54a468d19bc5727800096d9c7c51a423-Abstract-Conference.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Kanghui Tian, Siyuan Liu, Tianxiang Jiang, Shuai Dong, Yizhuo Li, Tian Ding, Yuan Guo, Songze Li, Haowen Hou, Congcong Wang, and Yi Wang. What should a self-teacher see? privileged context design for on-policy self-distillation. *arXiv preprint arXiv:2609.25623*, 2026. doi: 10.48550/arXiv.2609.25623. URL https://arxiv.org/abs/2609.25623.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract.html.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-Generated Text*, pp. 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher–student gap through interleaved sampling. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a2747a3844ca1e4667fbff3f558eb39b-Abstract-Conference.html.

Wenqi Yang, Jianjun Li, Zhibo Zhang, Mingqian Ding, and Yushen Fang. MARD: Module-aware reasoning distillation for language models with adaptive supervision. In *Proceedings of the 64th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 37680–37695. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1749. URL https://aclanthology.org/2026.acl-long.1749/.

Yuming Yang, Mingyoung Lai, Wanxu Zhao, Xiaoran Fan, Zhiheng Xi, Mingqi Wu, Chiyue Huang, Jun Zhao, Haijun Lv, Jian Tong, Yunhua Zhou, Yicheng Zou, Qipeng Guo, Tao Gui, Qi Zhang, and Xuanjing Huang. Which reasoning trajectories teach students to reason better? a simple metric of informative alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42123–42150. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.1950. URL https://aclanthology.org/2026.acl-long.1950/.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. Advancing LLM reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024. doi: 10.48550/arXiv.2404.02078. URL https://arxiv.org/abs/2404.02078.

XiuYu Zhang, Wei Chow, Junfeng Fang, Xingyu Zhu, Zhenkai Liang, and Tat-Seng Chua. What does privileged information add to on-policy self-distillation? *arXiv preprint arXiv:2609.20612*, 2026. doi: 10.48550/arXiv.2609.20612. URL https://arxiv.org/abs/2609.20612.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. doi: 10.48550/arXiv.2601.18734. URL https://arxiv.org/abs/2601.18734.

A Proof of Theorem 1

Proof. For a target mode m, Equations 12 and 13 give two possible transitions at update t. Using the indicator $\mathbf{1}\{\cdot\}$, they can be written jointly as

$$D_t^m \le D_{t-1}^m \exp\Bigg(-\mathbf{1}\{m(t)=m\} I_m^{(t)} + \mathbf{1}\{m(t)\neq m\}\frac{\epsilon}{a}\Bigg). \tag{14}$$

Applying Equation 14 recursively from $t=1$ to T yields

$$\begin{aligned} D_T^m &\le D_0^m \prod_{t=1}^{T} \exp\left(-\mathbf{1}\{m(t)=m\} I_m^{(t)} + \mathbf{1}\{m(t)\neq m\}\frac{\epsilon}{a}\right) \\ &= D_0^m \exp\left(-\sum_{t=1}^{T} \mathbf{1}\{m(t)=m\} I_m^{(t)} + \frac{\epsilon}{a}\sum_{t=1}^{T}\mathbf{1}\{m(t)\neq m\}\right), \end{aligned} \tag{15}$$

where the equality follows from the product rule for exponentials. This is exactly Equation 13. □

Justification of the cross-mode bound. Consider one optimizer update

$$\theta_{t+1} = \theta_t - \eta g_{m(t)}(\theta_t), \tag{16}$$

where

$$g_{m(t)}(\theta_t) = \nabla_\theta \mathcal{L}_{m(t)}(\theta_t) \tag{17}$$

is the gradient induced by the sampled training mode.

Let $\mathcal{L}_m^H$ denote the held-out loss associated with target mode m, and define

$$g_t^{H,m} = \nabla_\theta \mathcal{L}_m^H(\theta_t). \tag{18}$$

A local first-order approximation gives

$$\mathcal{L}_m^H(\theta_{t+1}) - \mathcal{L}_m^H(\theta_t) \approx -\eta \left\langle g_t^{H,m}, g_{m(t)}(\theta_t) \right\rangle. \tag{19}$$

Therefore,

$$\begin{aligned} \left|\mathcal{L}_m^H(\theta_{t+1}) - \mathcal{L}_m^H(\theta_t)\right| &\approx \eta \left| \left\langle g_t^{H,m}, g_{m(t)}(\theta_t) \right\rangle \right| \\ &\le \eta \left\| g_t^{H,m} \right\| \left\| g_{m(t)}(\theta_t) \right\|, \end{aligned} \tag{20}$$

where the inequality follows from Cauchy–Schwarz.

If the gradient norms are locally bounded by

$$\left\| g_t^{H,m} \right\| \le G, \qquad \left\| g_{m(t)}(\theta_t) \right\| \le G, \tag{21}$$

then

$$\left|\mathcal{L}_m^H(\theta_{t+1}) - \mathcal{L}_m^H(\theta_t)\right| \lesssim \eta G^2. \tag{22}$$

Thus, taking

$$\epsilon = \eta G^2 \tag{23}$$

gives the local perturbation bound

$$D_t^m \le D_{t-1}^m + \epsilon, \qquad m(t) \neq m. \tag{24}$$

Finally, whenever $D_{t-1}^m \ge a > 0$,

$$\begin{aligned} D_{t-1}^m + \epsilon &= D_{t-1}^m \left(1 + \frac{\epsilon}{D_{t-1}^m}\right) \\ &\le D_{t-1}^m \exp\left(\frac{\epsilon}{D_{t-1}^m}\right) \\ &\le D_{t-1}^m \exp\left(\frac{\epsilon}{a}\right), \end{aligned} \tag{25}$$

where the first inequality uses $1 + x \le e^x$. This recovers Equation 13.

B Training algorithm and implementation

B.1 Algorithms

Algorithm 1 Adaptive Mode Probability Update

Require: Held-out set $\mathcal{D}_h$, current probabilities $\rho_{\mathrm{SELF}}, \rho_{\mathrm{ON}}$, previous losses $D_{\mathrm{SELF}}^{\mathrm{prev}}, D_{\mathrm{ON}}^{\mathrm{prev}}$, threshold δ

function UPDATE-ROUTING($\mathcal{D}_h, \rho, D^{\mathrm{prev}}$)
  for $m \in \{\mathrm{SELF}, \mathrm{ON}\}$ **do**
    Evaluate current held-out loss D_m on $\mathcal{D}_h$
    $I_m \leftarrow \dfrac{D_m^{\mathrm{prev}} - D_m}{D_m^{\mathrm{prev}} + \epsilon_a}$
    if $I_m < \delta$ **then**
      $\rho_m \leftarrow \min(\rho_m + \eta_m, \rho_m^{\max})$
    end if
    $D_m^{\mathrm{prev}} \leftarrow D_m$
  end for
  $\rho_{\mathrm{OFF}} \leftarrow 1 - \rho_{\mathrm{SELF}} - \rho_{\mathrm{ON}}$
  return $\rho_{\mathrm{OFF}}, \rho_{\mathrm{SELF}}, \rho_{\mathrm{ON}}, D_{\mathrm{SELF}}^{\mathrm{prev}}, D_{\mathrm{ON}}^{\mathrm{prev}}$
end function

Algorithm 2 RIVAD Training Procedure

Require: Training set $\mathcal{D}$, held-out routing set $\mathcal{D}_h$, student p_θ, teacher p_T, routing interval K, mixing weight β

Initialize self-teacher $p_{\widetilde{T}} \leftarrow p_\theta$
Initialize mode probabilities $\boldsymbol{\rho} = (\rho_{\mathrm{OFF}}, \rho_{\mathrm{SELF}}, \rho_{\mathrm{ON}})$
for optimizer step $s \leftarrow 1, 2, \ldots$ **do**
    Sample $(x, y) \sim \mathcal{D}$
    Sample mode $m \sim \mathrm{Categorical}(\boldsymbol{\rho})$
    Construct $(r^m, q, \widetilde{x})$ for mode m as in Section 3.2
    Compute $\mathcal{L}^m_{\mathrm{KD}}$ using Equation 2
    Compute $\mathcal{L}^m_{\mathrm{CE}}$ using Equation 10
    $\mathcal{L} \leftarrow (1-\beta)\mathcal{L}^m_{\mathrm{CE}} + \beta\mathcal{L}^m_{\mathrm{KD}}$
    Update student parameters θ using $\mathcal{L}$
    if $s \bmod K = 0$ **then**
        Evaluate D_{SELF} and D_{ON} on $\mathcal{D}_h$
        Update ρ_{SELF} and ρ_{ON} using Equation 8
        $\rho_{\mathrm{OFF}} \leftarrow 1 - \rho_{\mathrm{SELF}} - \rho_{\mathrm{ON}}$
        Synchronize $p_{\widetilde{T}} \leftarrow p_\theta$
    end if
end for

Table 5: Additional defaults from v2 launch scripts. Record runtime overrides for every reported run.

Setting	Value
LoRA dropout	0.05
Weight decay / gradient clipping	0.01 / 1.0
Student total / prompt limit	1,024 / 512 tokens
Teacher total / prompt limit	2,560 / 1,536 tokens
Pooling / magnitude normalization	Mean / z-score
Geometry ϵ / scheduler ϵ_a	10^{-6} / 10^{-8}
SELF / ON probability increment	0.05 / 0.025
Minimum relative-improvement threshold	0.05
Training seed (default)	10

B.2 Current configuration

The project command selects four GPUs by default, giving a nominal effective batch of $8 \times 4 \times 4 = 128$ examples per update. Record actual GPU model, memory, software versions, and wall time separately. The default ON new-token budget is $1{,}024 - 512 = 512$. Context preparation allows 1,024 new tokens with input and total limits of 8,192 and 10,240 tokens.

B.3 Alignment and masking

Canonical boundaries are mapped to tokenizer offsets to handle tokens crossing punctuation and newlines. Prompt tokens are excluded from pooling. In SELF mode, the shared response budget is reserved before trimming teacher context, retaining identical supervised response token IDs for both models. Invalid spans and pairs are masked. Geometry averages over eligible trajectories, whereas output losses average over supervised tokens. The current runtime performs teacher forward passes during training; it does not cache geometry targets.

C Data and privileged-context construction

The preparation prompt asks the teacher to generate concise supporting facts, definitions, hints, formulas, or instruction guidelines using a question and its reference response. It requests supporting context rather than a replacement answer or full worked solution. This instruction does not guarantee

Table 6: Performance comparison on Gemma2-9B (Teacher) → Gemma2-2B (Student).

Method	GSM8K	GSM-Plus	MATH	MBPP	STEM	Avg.
Gemma2-9B (Teacher)	57.7	40.8	25.8	40.0	50.2	42.9
Gemma2-2B (Student)	53.7	36.2	16.1	39.2	25.5	34.1
Sequence-Level KD	53.4	33.8	17.7	39.2	25.9	34.0
Supervised-KD	53.6	36.8	17.9	39.0	25.5	34.6
DistiLLM	54.3	35.8	17.8	39.6	25.2	34.5
MiniLLM	54.4	36.8	17.5	**40.6**	25.9	35.1
GKD ($\beta = 0.9$)	54.1	36.9	17.2	40.0	26.2	34.9
TSD-KD	55.3	**37.9**	18.9	40.4	27.0	35.9
CSD	$47.2_{\pm 1.38}$	$32.0_{\pm 0.45}$	$19.6_{\pm 0.54}$	$31.6_{\pm 2.08}$	$47.4_{\pm 0.86}$	35.6
AMiD	$49.6_{\pm 1.48}$	$33.8_{\pm 0.26}$	$20.5_{\pm 0.15}$	$33.4_{\pm 2.06}$	$44.7_{\pm 0.79}$	36.4
RIVAD (Ours)	$\mathbf{54.8}_{\pm 1.17}$	$37.0_{\pm 0.37}$	$\mathbf{24.1}_{\pm 0.55}$	$38.0_{\pm 2.07}$	$\mathbf{46.2}_{\pm 0.81}$	**40.0**

that the output excludes answer information. The original response remains the supervised target. Context is inserted into the teacher's user turn before the assistant generation header.

To complete: dataset identifiers and licenses, source revisions, split sizes, deduplication, question-group split handling, response provenance, generation settings, context-length statistics, and benchmark-overlap checks. Record how often truncation removes context or response steps.

D Additional results and diagnostics

Efficiency and reliability. Report GPU-hours, peak memory, generated tokens, teacher tokens, and update counts, including context preparation and development rollouts. Use matched-update and matched-compute comparisons. Run at least three training seeds for central comparisons and report mean and variability. Tune on development data and evaluate final configurations under a fixed test protocol. Appendix D reserves space for detailed outcomes and qualitative traces.

Training efficiency. Table 7 compares the training cost of RIVAD with recent white-box distillation methods under the same 2× H200 setup. RIVAD requires 70.0 minutes per epoch, remaining close to CSD (66.3 min) and AMiD (65.1 min), while being substantially faster than TSD-KD (420.0 min). In terms of memory, RIVAD reaches a peak usage of 114.0 GB/GPU, lower than CSD (127.9 GB/GPU) and AMiD (125.5 GB/GPU), although TSD-KD has the lowest memory footprint at 90.0 GB/GPU. Overall, RIVAD introduces only a modest runtime overhead relative to CSD and AMiD while reducing their peak memory consumption, and is considerably more time-efficient than TSD-KD.

Table 7: Training efficiency comparison under the same hardware configuration: 2× NVIDIA H200 140GB GPUs with a batch size of 8 per device. Runtime is reported in minutes per epoch, and memory denotes peak GPU memory usage.

Method	Time / Epoch (min)	Peak Memory (GB/GPU)
TSD-KD	419.7	**92.4**
CSD	66.3	127.9
AMiD	**65.1**	125.5
RIVAD (Ours)	70.3	114.7

Self-distillation truncation strategy The results in Table 8 show that random truncation consistently provides the best overall performance, achieving an average score of 54.76, outperforming no truncation (54.42), fixed 25% truncation (54.32), and fixed 50% truncation (54.38). Notably,

Table 8: Ablation study on truncation strategies for self-distillation.

Truncation Strategy	GSM8K	GSM-Plus	MATH	MBPP	STEM	Avg.
No truncation	**70.1**	**50.4**	52.7	46.8	52.1	54.42
Fixed 25%	**70.1**	50.1	52.3	46.8	52.3	54.32
Fixed 50%	69.8	**50.4**	52.4	47.0	52.3	54.38
Random (Ours)	68.8	50.2	**53.0**	**49.2**	**52.6**	**54.76**

random truncation achieves the best results on MATH, MBPP, and STEM, while maintaining competitive performance on GSM8K and GSM-Plus. These results suggest that randomly removing a portion of the privileged context can preserve sufficient useful information for self-distillation while introducing beneficial diversity in the training contexts.

Moreover, random truncation offers an additional efficiency advantage over using the full context without truncation. By reducing the amount of context provided to the model, it lowers the context length and consequently reduces the computational and memory cost during training. Therefore, random truncation provides a favorable trade-off between performance and context efficiency, achieving the highest average accuracy while requiring less contextual information than the no-truncation setting.