CoKD: Generating Student-Reachable Targets through Teacher–Student Collaborative Decoding

Anonymous authors
Paper under double-blind review

Abstract

Knowledge distillation (KD) is a standard approach for compressing large language models, yet its effectiveness critically depends on the quality of the targets used to supervise the student. Teacher-generated targets can be informative but often lie outside what a small student can reliably learn, whereas student-only rollouts remain reachable but may encode suboptimal decisions. We propose *Collaborative Knowledge Distillation* (CoKD), which generates distillation targets through teacher–student collaborative decoding. At each decoding step, CoKD samples from a tempered distribution that combines teacher preference with student support, favoring tokens that are endorsed by both teacher and student. A task-aware sequence-level quality filter further retains only collaborative responses that provide useful training targets. Across arithmetic reasoning, instruction following, and dialogue summarization benchmarks with Qwen2.5 and Gemma teacher–student pairs, CoKD consistently improves over competitive KD baselines and remains stable across varying teacher scales. These results indicate that effective distillation requires targets that are not only teacher-preferred but also student-reachable.

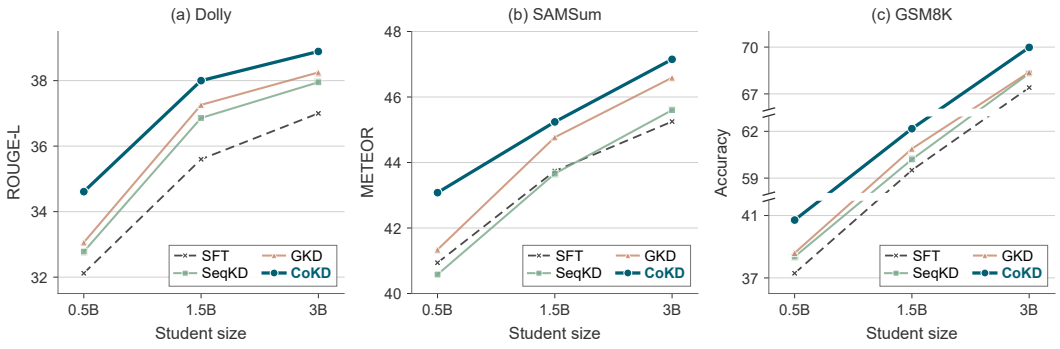

Figure 1: Student performance vs. student size with a fixed 7B teacher. The teacher and students are all from the Qwen2.5 model family (Qwen, 2025). All KD methods start from the corresponding student checkpoints obtained through supervised fine-tuning (SFT), and the teacher has likewise undergone SFT. CoKD consistently outperforms SFT and the KD baselines across all three student sizes and tasks: **(a)** Dolly for instruction following, **(b)** SAMSum for dialogue summarization, and **(c)** GSM8K for arithmetic reasoning.

1 Introduction

In recent years, large language models (LLMs) have achieved remarkable success across a wide range of natural language tasks (OpenAI, 2023; Qwen, 2025; Guo et al., 2025). They can generalize well across diverse domains and produce natural, high-quality outputs with minimal task-specific fine-tuning. However, deploying LLMs in practical scenarios remains expensive, as inference with large models often requires substantial computation and incurs high latency. Knowledge distillation (KD) is a widely used method for model compression by transferring the capabilities of a large teacher model to a smaller student model, enabling more efficient inference (Hinton et al., 2015).

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.

Existing KD methods typically train the student to match the teacher's full distribution on gold reference responses, through objectives like forward KL divergence, reverse KL divergence, or f-divergence (Ko et al., 2024; Wen et al., 2023; Hu et al., 2023). However, under a large teacher–student capacity gap, the teacher's distribution may encode signals too difficult for the student to absorb, and uniform token-level imitation can dilute the most instructive supervisory targets (Wang et al., 2021; Tavor et al., 2026). This has motivated a line of work on *distillation target design*, which proposes to adjust *what* the student learns from the teacher through example selection, token filtering, or reweighting training signals (Mirzadeh et al., 2020; Panigrahi et al., 2026; Xie et al., 2026). The core idea is that a well-designed target provides more effective supervision than directly imitating the teacher's raw distribution (Zhang et al., 2025). Among them, a prominent direction focuses on constructing alternative *target sequences*, replacing gold references with newly generated responses on which the student imitates the teacher. Sequence-level knowledge distillation (SeqKD) pioneered this paradigm by replacing gold targets with teacher-generated outputs (Kim and Rush, 2016; Setiawan, 2024; Wang et al., 2024). More recently, GKD trains on student-generated trajectories to reduce train–inference mismatch (Agarwal et al., 2024), while SKD and SWITCH incorporate teacher guidance into student-led generation (Xu et al., 2025; Koo et al., 2025).

Despite these advances, a fundamental question remains underexplored: can we construct a distillation target that provides high-quality supervision while remaining easy for the student to learn? Prior KD methods train the student to imitate the teacher-derived targets, implicitly assuming that after distillation the student will reproduce teacher-like responses and thereby improve task performance (Mirzadeh et al., 2020; Liu et al., 2024). However, this assumption may not hold unless we adequately account for student learnability—specifically, whether the teacher's outputs can be reliably learned by a lower-capacity student, particularly when a substantial capacity gap exists. Therefore, we argue that an ideal distillation target should be not merely *teacher-preferred*, but also *student-reachable*, aligning with the student's capacity and remaining within what the student can effectively internalize.

In this work, we propose **CoKD** (*Collaborative Knowledge Distillation*), which constructs distillation targets through teacher–student *collaborative decoding*. We define a response as *teacher-preferred* and *student-reachable* when both models assign it high probability mass. At each decoding step, CoKD scores candidate tokens by combining the teacher's and student's predictions, explicitly favoring candidates endorsed by both models. After constructing candidate target responses, CoKD applies a task-aware sequence-level quality filter to retain only responses that satisfy a predefined quality criterion. The student is then trained on the accepted targets. We optimize the student with standard cross-entropy loss rather than the classic KL-based distribution-matching objectives used in many KD methods (see §3.4 for further details).

Experiments on different text-generation benchmarks show that CoKD consistently outperforms SFT and competitive KD baselines as student size scales from 0.5B to 3B (Figure 1). These gains hold across all tasks and student scales, with the largest improvements over baselines observed for the smallest student models. We summarize our key contributions as follows:

- We identify *student reachability* as a key yet underexplored factor in knowledge distillation, and quantify it with student's next-token probability over candidate tokens.
- We propose **CoKD**, a collaborative distillation framework that creates targets through teacher–student collaborative decoding and task-aware sequence-level quality control, balancing teacher preference with student reachability.
- We demonstrate consistent improvements over strong KD baselines across multiple generation benchmarks and teacher–student pairs, confirming the value of capacity-aligned target construction.

2 Related Work

Knowledge distillation for LLMs. Knowledge distillation (KD) is a central approach for transferring the capabilities of LLMs into smaller, deployable students. Recent surveys organize this space around white-box and black-box distillation, skill transfer, data generation, evaluation, and application settings (Xu et al., 2024; Yang et al., 2025). Existing LLM distillation methods differ in what is transferred, including teacher rationales or reasoning traces (Hsieh et al., 2023), token distributions and divergence objectives (Gu et al., 2024), adversarially generated instructions from proprietary

teachers (Jiang et al., 2023), decomposed reasoning capabilities (Wu et al., 2024), domain-specific knowledge (Liu et al., 2024; Feng et al., 2026), OOD-guided synthetic data (Gholami et al., 2024), draft-model alignment for speculative decoding (Zhou et al., 2024), and selective token weighting (Huang et al., 2026). These works show that LLM distillation is not a single objective, but a family of design choices over targets, data, and losses.

Sequence- and token-level target distillation. Sequence-level knowledge distillation (SeqKD) trains the student on teacher-generated targets and was first shown to simplify neural machine translation targets (Kim and Rush, 2016). Subsequent work extends it to non-autoregressive translation (Zhou et al., 2020), selects student-beneficial targets (Wang et al., 2021; Liu et al., 2023), reranks teacher n-best hypotheses (Setiawan, 2024), and reuses richer teacher-generated candidates (Wang et al., 2024). In parallel, token-level distillation transfers supervision at each decoding step, typically by matching the teacher's full next-token distribution through KL or related divergence objectives (Zhang et al., 2023; Ko et al., 2025). Recent studies explore sentence- versus token-level trade-offs (Wei et al., 2024), bidirectional token supervision from pretrained models (Zhuang and Tu, 2023), token-adaptive reweighting for LLM distillation (Xie et al., 2026), and generalized objectives based on f-divergence minimization (Wen et al., 2023). These works show that teacher targets can simplify learning, but also that the most useful target depends on student capacity and decoding difficulty. CoKD follows this target-construction perspective, but explicitly couples token-level teacher preference with student support and then filters the resulting sequence by task utility.

On-policy distillation. On-policy distillation addresses the train–inference mismatch of autoregressive models by training on states induced by the student itself, echoing ideas from imitation learning and data aggregation (Ross et al., 2011; Song and Zheng, 2026). ImitKD and GKD apply this principle to language generation by evaluating teacher feedback on student-generated trajectories (Lin et al., 2020; Agarwal et al., 2024), while MiniLLM and DistillSpec further highlight the importance of reverse-KL-style objectives and on-policy draft-model data (Gu et al., 2024; Zhou et al., 2024). Recent work extends on-policy distillation to context distillation (Ye et al., 2026), self-distillation for reasoning (Zhao et al., 2026), entropy-aware objectives (Jin et al., 2026), autonomous-driving planning (Afsharrad et al., 2026), and constrained-MDP formulations (Zimmer et al., 2025). Among the most closely related methods, SWITCH and SKD introduce teacher–student interaction during target generation by selectively replacing tokens in student-generated sequences (Koo et al., 2025; Xu et al., 2025), while Veto reformulates the on-policy objective with a product-of-experts KL target on student rollouts (Jang et al., 2026). In contrast,CoKD constructs targets through collaborative decoding that explicitly balances teacher preference with student reachability, filters them with task-aware sequence filtering, and trains with cross-entropy rather than KL matching.

3 METHOD

3.1 PROBLEM SETUP AND METHOD OVERVIEW

Given a dataset $\mathcal{D} = \{(x_i, y_i^\star)\}_{i=1}^N$, where x is the input prompt and $y^\star$ is the gold reference, our goal is to construct an alternative target response for each prompt that is not only informative under the teacher, but also reachable under the student. We assume access to a teacher model p_T and a student model p_S, where the teacher is stronger but may produce targets that are difficult for the lower-capacity student to learn directly. Existing target construction strategies often rely on teacher-centered sequences (Kim and Rush, 2016; Wang et al., 2021; Zhou et al., 2020), or use student-generated trajectories as proposals for teacher-guided target construction (Lin et al., 2020; Agarwal et al., 2024; Xu et al., 2025). In contrast, CoKD combines teacher and student signals during decoding, constructing targets that balance teacher informativeness with student reachability.

A central difficulty in sequence-level distillation is that teacher-preferred targets are not always learnable for a small student, while student-centered generations remain reachable but may preserve suboptimal decisions. We address this mismatch with a two-stage procedure. First, given an input prompt x, CoKD constructs a collaborative response $y^{\text{co}} = (y_1^{\text{co}}, \ldots, y_L^{\text{co}})$ autoregressively, where each token is sampled from a distribution jointly induced by the teacher and the student. Second, we apply a *task-aware sequence quality control* rule that retains the collaboratively generated response only when it yields a higher-quality sequence than the student's own rollout or reaches a predefined quality threshold. An overview of CoKD is shown in Figure 2.

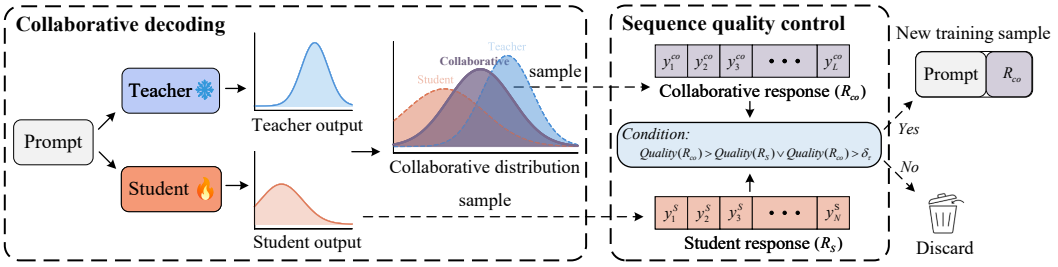

Figure 2: Overview of **CoKD** target construction and training. Given prompt x, token-level collaborative decoding combines teacher preference and student support at each step to sample a collaborative target y^{co}. A parallel student rollout y^{S} enables task-aware sequence quality control using reference $y^{\star}$. Retained collaborative targets are used for standard supervised fine-tuning of the student.

3.2 Token-Level Collaborative Decoding

Collaborative token scoring. Consider autoregressive generation at step t under the collaborative prefix $h_t = (x, y^{\text{co}}_{<t})$, where $y^{\text{co}}_{<t}$ denotes the previously generated tokens. Conditioned on h_t, the teacher and student induce the next token over the vocabulary $\mathcal{V}$. For each candidate token $v \in \mathcal{V}$, we define a collaborative score:

$$s_t(v) = \alpha \log p_T(v \mid h_t) + (1-\alpha) \log p_S(v \mid h_t), \tag{1}$$

where $p_T(\cdot \mid h_t)$ and $p_S(\cdot \mid h_t)$ denote the teacher and student next-token distributions, respectively, and $\alpha \in [0, 1]$ controls the relative influence of teacher preference and student reachability.

Equation (1) has a simple interpretation, namely, a token receives a high score only when it is simultaneously assigned non-trivial probability mass by both models. Conversely, tokens that are strongly disfavored by either model are down-weighted. As a result, the score favors tokens that lie in the intersection of *teacher preference* and *student support*. This stands in contrast to teacher-only target construction, which may produce targets beyond the student's capacity, and student-centered generation, which may remain reachable but less informative.

Reachability-constrained collaborative sampling. Instead of greedily selecting the token with the maximum score in Eq. (1), we convert the scores into a collaborative distribution and then sample from it. However, under a large teacher–student capacity gap, the teacher and student can place their probability mass on substantially different regions of the vocabulary $\mathcal{V}$, leaving no tokens effectively supported by both models. To handle this scenario and ensure that the generated targets remain within the student's distributional support, we explicitly restrict the sampling space to an admissible token set:

$$\mathcal{A}_t = \{v \in \mathcal{V} \mid p_S(v \mid h_t) \geq \rho\}, \tag{2}$$

where $\rho \in [0, 1)$ is a reachability threshold (see its effect in Table 3). We then define the constrained collaborative distribution over $\mathcal{A}_t$ as:

$$q_{\text{co}}(v \mid h_t) = \frac{\exp(s_t(v)/T)}{\sum_{u \in \mathcal{A}_t} \exp(s_t(u)/T)}, \tag{3}$$

where $T > 0$ is a temperature parameter that can vary based on task type. The next token is drawn via categorical sampling, $y^{\text{co}}_t \sim q_{\text{co}}(\cdot \mid h_t)$, yielding the final collaboratively generated sequence $y^{\text{co}} = (y^{\text{co}}_1, \ldots, y^{\text{co}}_L)$. This formulation favors teacher-preferred tokens within the student's learnable region, while maintaining sampling diversity instead of collapsing to a single deterministic trajectory.

3.3 Task-Aware Sequence Quality Control

Conventional KD methods typically treat teacher-generated outputs as pseudo-gold targets that the student directly imitates (Kim and Rush, 2016; Hinton et al., 2015). By contrast, CoKD uses token-level collaborative decoding to construct student-reachable targets, with the student playing a substantial role in target construction. Although this design improves target learnability, the resulting collaborative responses may have lower task-level quality than the ones used in conventional KD. To

address this issue, we introduce a sequence-level quality control mechanism that filters unhelpful collaborative targets before student training.

Specifically, for the same input x, we generate a student-only response $y^S = (y_1^S, \ldots, y_{L'}^S)$ by sampling $y_t^S \sim p_S(\cdot \mid x, y_{<t}^S)$ at each step. We then define a task-aware utility function $U_\tau(x, y, y^\star)$, where τ denotes the task type and $y^\star$ is the reference response for input x. The utility is defined to quantify the sequence-level similarity between a generated response and the reference response, thereby providing a reference-grounded measure of sequence-level target quality. We instantiate U_τ using the natural supervised evaluation criterion of each task (see Appendix G.3). Then we define a shared sequence quality control rule across tasks:

$$g(x) = \mathbb{I}\big[\big(U_\tau(x, y^{\text{co}}, y^\star) > U_\tau(x, y^S, y^\star)\big) \vee \big(U_\tau(x, y^{\text{co}}, y^\star) \geq \delta_\tau\big)\big], \tag{4}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function and δ_τ is a predefined quality threshold for task type τ. That is, we retain a collaboratively generated target when it either improves over the student-only rollout or already reaches a sufficiently high task-level quality. This design ensures that training focuses on high-utility targets rather than treating all collaborative sequences as equally useful supervision.

3.4 Training Objective

CoKD interleaves target construction and student optimization. For each training example, the student generates the student-only rollout and participates in collaborative decoding. If the collaborative target satisfies Eq. (4), we then update the student using the autoregressive cross-entropy loss:

$$\mathcal{L}_{\text{CoKD}} = -\sum_{t=1}^{|y^{\text{co}}|} \log p_S(y_t^{\text{co}} \mid x, y_{<t}^{\text{co}}). \tag{5}$$

Notably, CoKD does not adopt standard distribution-matching objectives, such as KL or reverse-KL divergence, which require the student to approximate the teacher's full next-token distribution. Under a large teacher–student capacity gap, we argue that matching the full teacher distribution can be difficult and may yield suboptimal distillation signals. Therefore, CoKD uses collaborative decoding to build reachable targets and then trains the student with standard cross-entropy (see Table 4 for comparison of training objectives). We summarize the overall procedure of CoKD in Algorithm E.1.

4 Experiments

4.1 Experimental Setup

Student and teacher models. We conduct experiments on two instruction-tuned model families, Qwen2.5 (Qwen, 2025) and Gemma 3 (Gemma Team, 2025). For the main comparison (Table 1), we use Qwen2.5-7B-Instruct and Gemma-3-4B-IT as teachers and Qwen2.5-0.5B-Instruct and Gemma-3-1B-IT as the corresponding students. For teacher-scaling analysis (§4.3), we fix the student to Qwen2.5-0.5B-Instruct and vary the teacher size among Qwen2.5-Instruct checkpoints at 1.5B, 3B, 7B, and 14B parameters ($3\times$–$28\times$ larger than the student).

Datasets and evaluation metrics. We evaluate on three generation benchmarks: GSM8K (Cobbe et al., 2021) for mathematical reasoning, Databricks-Dolly-15k (Conover et al., 2023) for instruction-following, and SAMSum (Gliwa et al., 2019) for dialogue summarization. Each dataset provides paired prompts and reference responses. For Databricks-Dolly-15k, we randomly select 14K, 500, and 500 samples for training, validation, and testing, respectively, following (Gu et al., 2024; Koo et al., 2025). For GSM8K, we use the official training and test splits provided in the dataset repository. For SAMSum, we evaluate KD in a low-resource setting by sampling 2K examples from the original training set for training, while keeping the original 819-example validation set for validation and the 819-example test set for evaluation. Following prior work, we report ROUGE-L (R-L) for Databricks-Dolly-15k (Lin, 2004), METEOR (MET) for SAMSum (Banerjee and Lavie, 2005), and exact-match accuracy (Acc.) for GSM8K.

Baselines. We compare against standard SFT, logits-based knowledge distillation (Logits KD), sequence-level KD (SeqKD) (Kim and Rush, 2016), generalized knowledge distillation (GKD) (Agarwal et al., 2024), speculative knowledge distillation (SKD) (Xu et al., 2025), and SWITCH (Koo et al., 2025). The implementation details of each baseline are summarized in Appendix F.

Training details. Following GKD (Agarwal et al., 2024), we first train both teacher and student on the task training split with supervised fine-tuning to obtain a task-adapted initialization. All distillation methods then start from the SFT checkpoints and are trained with the same optimization recipe (learning rate, batch size, number of epochs, maximum sequence length), so that performance differences primarily reflect each method's effectiveness. CoKD uses collaborative decoding with interpolation weight α, reachability threshold ρ, and sampling temperature T, followed by sequence-level quality control with threshold δ_τ. In the main experiments, we set $\alpha = 0.5$ and $\rho = 0.05$ across all tasks. For task-dependent parameters, we use $T = 1.0$ on Dolly and SAMSum, and $T = 0.8$ on GSM8K to encourage more deterministic token choices for mathematical reasoning. The quality-control threshold δ_τ is set to 0.6, 0.8, and 1.0 for Dolly, SAMSum, and GSM8K, respectively. These default parameters are used in the main experiments, and their effects are further examined in the ablation studies. We summarize the full implementation details of CoKD in Appendix G.

4.2 Main Results

	Qwen2.5-7B-Inst → Qwen2.5-0.5B-Inst			Gemma-3-4B-It to Gemma-3-1B-It		
Method	Dolly (R-L)	SAMSum (MET)	GSM8K (Acc.)	Dolly (R-L)	SAMSum (MET)	GSM8K (Acc.)
Teacher	38.75	46.68	75.82	35.89	44.48	57.39
SFT	32.12	40.94	37.30	33.17	38.67	33.81
Logits KD	33.69	41.54	37.83	33.31	40.04	36.39
SeqKD	32.78	40.58	38.36	32.34	38.17	34.87
GKD	33.06	41.34	38.67	33.47	38.76	**37.13**
SKD	33.77	41.71	39.42	32.80	39.31	35.63
SWITCH	33.53	41.08	38.51	33.10	38.94	36.03
CoKD	**34.61**	**43.08**	**40.71**	**34.88**	**41.07**	36.70

Table 1: Main results on three benchmarks: GSM8K for arithmetic reasoning, Dolly for instruction following, and SAMSum for dialogue summarization. Teacher denotes the teacher model performance, while all other rows report the student performance.

As shown in Table 1, we evaluate CoKD using two teacher–student pairs, with Qwen2.5-7B-Instruct and Gemma-3-4B-IT as teachers and Qwen2.5-0.5B-Instruct and Gemma-3-1B-IT as the corresponding students. CoKD consistently improves over KD baselines across most tasks and model pairs, indicating its effectiveness in constructing student-reachable and teacher-preferred targets. The gains are especially pronounced on SAMSum, where we use a low-resource setting with only 2K training examples. By constructing new collaborative target responses, CoKD enriches the available supervision beyond the original references while ensuring that the targets are easy for the student to imitate. Moreover, CoKD effectively improves the student even when the teacher–student gap is narrow, as on Dolly with the Gemma-3 pair. By contrast, other KD baselines provide little or no additional gain in this setting, underscoring the robustness of CoKD. This motivates the teacher-scaling study in §4.3, where we fix the student and compare how CoKD behaves relative to other KD baselines as teacher capacity varies.

4.3 Teacher Scaling under Reachability-Aware Distillation

Teacher capacity is a critical factor in knowledge distillation, as the choice of teacher size can substantially influence the quality of the distilled student (Panigrahi et al., 2026; Cho and Hariharan, 2019). A teacher that is too small may provide noisy or weak supervision, whereas an overly large teacher may produce signals that are difficult for the student to effectively learn (Zhang et al., 2025). In this experiment, we examine the effect of teacher scale using the Qwen2.5 model family (Qwen, 2025), which provides a dense ladder of checkpoints at multiple scales. Specifically, we fix the student to Qwen2.5-0.5B-Instruct and scale the teacher from 1.5B to 14B. Figure 3 summarizes student performance vs. teacher size across tasks.

Overall, CoKD consistently improves student performance across teachers of different sizes, suggesting that it is less sensitive to variations in teacher capacity and can mitigate the performance instability commonly observed in KD baselines. When the teacher becomes substantially larger than the student (e.g., 14B parameters, 28× the student), several baselines exhibit degraded student

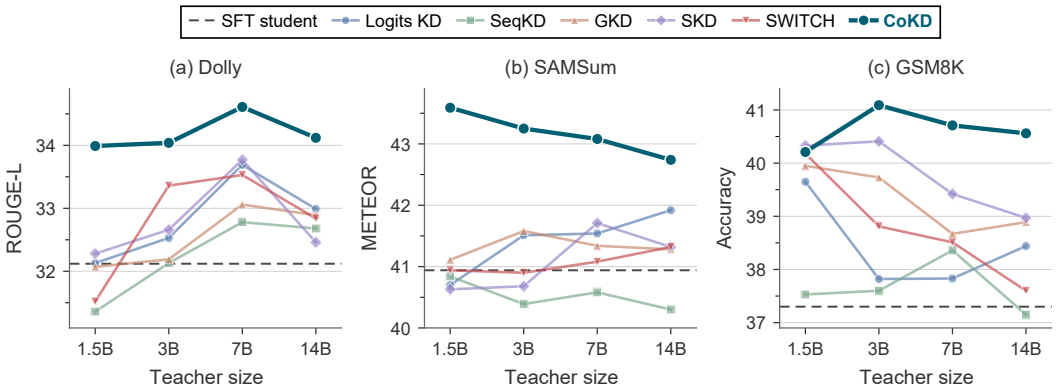

Figure 3: Student performance vs. teacher size with a fixed Qwen2.5-0.5B-Instruct student. **(a)** Dolly (ROUGE-L), **(b)** SAMSum (METEOR), and **(c)** GSM8K (Accuracy). The dashed line denotes the SFT student baseline. CoKD is the best or among the strongest distillation methods on all three tasks, while student performance does not improve consistently as the teacher grows.

performance. This trend is consistent with prior findings that an excessive teacher–student capacity gap can make teacher supervision difficult for the student to internalize (Zhang et al., 2025). CoKD, however, maintains appreciable improvements in this setting, suggesting that its reachability-aware design helps keep the teacher supervision within the student's learnable range. Moreover, on Dolly and SAMSum, when the teacher–student performance gap is relatively small (see Appendix H for teacher performance across different sizes), CoKD still achieves considerable gains while other KD baselines provide little or no additional improvement. This result suggests that the task-aware quality control mechanism (§3.3) of CoKD can effectively filter targets that are unhelpful, allowing student performance to remain stable even when smaller teachers provide weaker supervision.

4.4 Ablation Studies

We ablate the components that directly control target construction and student optimization in CoKD: the collaboration weight α, the reachability threshold ρ, the collaborative decoding temperature T, and the training objective. Additionally, we evaluate the effect of sequence-level quality control in Appendix I, while Appendix J ablates its threshold δ_τ. Unless noted otherwise, each ablation sweep is conducted with Qwen2.5-7B-Instruct as the teacher and Qwen2.5-0.5B-Instruct as the student, using the same training recipe as in the main experiments and fixing all non-swept parameters to their default values.

Effect of the collaboration weight α. In this ablation, we vary α and report four quantities: downstream task performance, student reachability $\bar{p}_S$, teacher preference $\bar{p}_T$, and the accepted-target ratio after sequence-level filtering. The reachability and preference statistics are computed as the average student and teacher probabilities assigned to tokens in the constructed collaborative responses.

We report the α ablation on Dolly in Figure 4. Varying α reveals a trade-off between teacher preference and student reachability in the constructed targets. As α increases, collaborative decoding becomes more teacher-dominated, where teacher preference rises and student reachability falls. Downstream performance, however, is not monotonic and peaks at $\alpha = 0.7$, outperforming both student-heavy and teacher-heavy extremes. Figure 4 (b) further shows that the accepted-target ratio increases with α as teacher becomes more dominant, yielding more training samples. However, downstream performance does not improve monotonically with this ratio. This indicates that more accepted targets do not necessarily lead to better distillation, because reduced student reachability can offset the benefit of additional supervision. Therefore, it is better to maintain a moderate α that balances teacher preference and student reachability. The GSM8K and SAMSum α ablations are provided in Appendix K.

Effect of collaborative decoding temperature T. Temperature T in Eq. (3) controls how sharply collaborative decoding concentrates probability mass on high-scoring tokens, trading off between

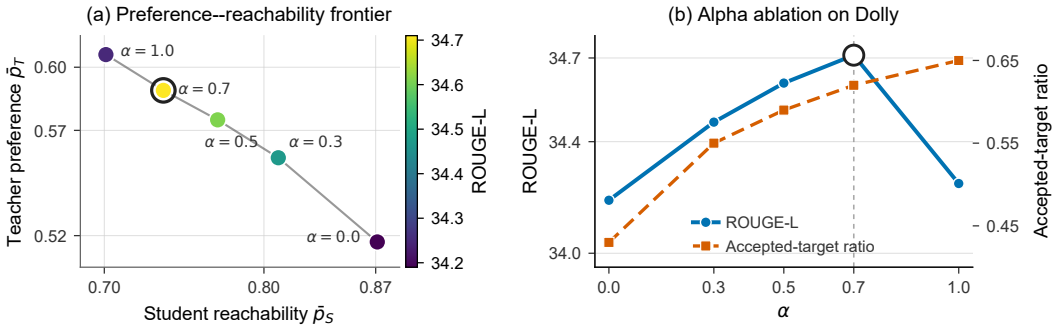

Figure 4: Ablation of the collaboration weight α. **(a)** Preference–reachability frontier induced by changing α; color denotes Dolly ROUGE-L and the ring marks the best setting. The frontier is relatively shifted toward student reachability because collaborative sampling is restricted to the admissible token set in Eq. (2). **(b)** Dolly ROUGE-L and accepted-target ratio under the same α sweep. Larger α increases teacher preference and accepted-target ratio but reduces student reachability.

greedy decoding and stochastic sampling. We compare greedy decoding with temperature-based sampling across $T \in [0.5, 2.0]$ and report student performance in Table 2.

Table 2 shows that temperature-based sampling consistently outperforms greedy decoding across all three tasks, suggesting that introducing controlled exploration during decoding leads to more effective target construction. Greedy decoding deterministically selects the highest-scoring token at each step, which may produce overly narrow trajectories and limit the diversity of useful distillation targets. In contrast, moderate temperatures allow collaborative decoding to sample from high-scoring alternatives, leading to stronger downstream performance. The optimal temperature is mildly task-dependent: Dolly performs best at $T = 1.0$, SAMSum at $T = 1.5$, and GSM8K at $T = 0.8$. This pattern suggests that open-ended generation tasks can benefit from more exploratory target construction, while mathematical reasoning favors a lower temperature that preserves more deterministic token choices. Overall, CoKD benefits from controlled stochasticity, while overly deterministic decoding is less effective for constructing distillation targets.

Table 2: Ablation over collaborative decoding temperature T with Qwen2.5-7B-Instruct → Qwen2.5-0.5B-Instruct. Best score in each column is in bold.

T	Dolly (R-L)	SAMSum (MET)	GSM8K (Acc.)
0.5	33.36	42.96	40.41
0.8	34.13	42.69	**40.71**
1.0	**34.61**	43.08	39.50
1.5	34.36	**43.56**	39.57
2.0	34.45	42.28	39.20
Greedy	32.85	39.59	36.27

Effect of the reachability threshold ρ. The threshold ρ in Eq. (2) controls the set of tokens eligible for collaborative sampling at each decoding step. To distinguish the effect of student-side reachability from teacher-side preference, we also evaluate a teacher-preference filtering baseline that selects admissible tokens according to teacher probability instead of student probability.

Table 3 shows that introducing a moderate student-reachability constraint improves performance over the unfiltered setting on all three tasks, with the best threshold falling between $\rho = 0.05$ and $\rho = 0.10$. This supports the role of the reachability threshold, which removes tokens with weak student support and makes the constructed targets easier for the student to absorb. However, setting ρ too large consistently hurts performance, suggesting that overly restrictive filtering narrows the candidate space and reduces the diversity of target trajectories. This is consistent with the weaker performance of greedy decoding in Table 2. Figure 5 provides a mechanistic view of this trade-off. For Dolly and SAMSum, increasing ρ steadily reduces the effective candidate count, while performance first improves and then drops, indicating that a moderate candidate set is preferable to either full-vocabulary sampling or near-deterministic filtering. Interestingly, GSM8K shows a different pattern: applying a small reachability floor increases the effective candidate count compared with no filtering. This is likely because the teacher–student performance gap is much larger on GSM8K, where the teacher is highly confident at many decoding steps. As a result, the collaborative distribution sharply concentrated on a small number of teacher-preferred tokens, while reachability filtering and renormalization expose more student-supported alternatives.

Overall, the teacher-preference filter performs worse than the reachability-based filters, even it if retains a comparable or larger number of effective candidates. This indicates that the benefit of CoKD does not come merely from the size of the admissible set and teacher preference alone is insufficient for constructing effective distillation targets, rather, student-reachability provides a more reliable criterion for balancing target quality and learnability during target construction.

Table 3: Ablation over token-filtering strategies. The left block sweeps the student-reachability threshold ρ in Eq. (2), where $\rho = 0.00$ means no reachability filtering (the full vocabulary is admissible). The right column reports an analogous teacher-preference filter, which retains candidate tokens with $p_T(v \mid h_t) > 0.05$. Rows report downstream task scores (ROUGE-L, METEOR, and accuracy).

	Student-reachability ρ				Teacher pref.
Task	0.00	0.05	0.10	0.20	$p_T = 0.05$
Dolly	33.73	34.61	34.18	33.63	33.62
SAMSum	42.27	43.08	43.30	41.78	41.70
GSM8K	40.41	40.71	39.58	38.97	37.60

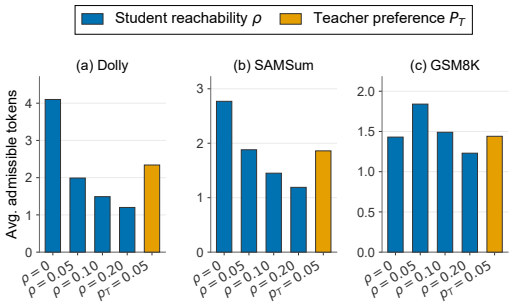

Figure 5: Average number of admissible tokens under different filtering strategies. For $\rho = 0$, the admissible token set is the full vocabulary, but here we report only tokens with non-negligible sampling probability $q_{\text{co}}(v \mid h_t) > 0.01$ under Eq. (3).

Effect of the training objective. After constructing target responses via teacher–student collaborative decoding, the student can be optimized with different training objectives. We compare standard cross-entropy (CE) against forward KL (FKL), reverse KL (RKL), Jensen–Shannon divergence (JSD) with $\beta = 0.5$, and a combined CE+FKL objective on the same accepted target responses, with all other CoKD settings held fixed.

Table 4 shows that CE consistently achieves the best student performance across all three tasks. Unlike KL-based objectives, which force the student to match the teacher's full next-token distribution at every position, CE is easier for the student to optimize because it only requires to predict the target tokens. This distinction is particularly important for CoKD: because collaborative decoding selects tokens that are both teacher-preferred and student-supported, the student already assigns non-negligible probability mass to the selected tokens. The resulting targets are therefore inherently reachable for the student, making CE a simple and effective objective for absorbing the teacher supervision embedded in the collaborative responses. Notably, CE+FKL performs worse than CE alone, indicating that imposing additional distribution-level supervision can make the learning target less student-reachable and even hurt the final student performance. This supports our design choice of using CE as training objective for CoKD. We provide an additional analysis of loss choices in Appendix B.

Table 4: Ablation over the training objective on accepted CoKD targets (Qwen2.5-7B-Instruct → Qwen2.5-0.5B-Instruct).

Objective	Dolly (R-L)	SAMSum (MET)	GSM8K (Acc.)
CE	**34.61**	**43.08**	**40.71**
FKL	33.19	42.09	39.34
RKL	33.67	42.07	40.09
JSD	33.27	41.48	39.95
CE + FKL	33.38	42.96	39.12

5 Conclusion

We presented **CoKD**, a collaborative knowledge distillation framework that constructs training targets through teacher–student collaborative decoding. By balancing *teacher preference* with *student reachability*, CoKD produces targets that are both informative and learnable for the student. Across arithmetic reasoning, instruction following, and dialogue summarization, CoKD consistently outperforms strong KD baselines, with particularly clear gains on open-ended generation tasks, and remains effective under larger teacher–student capacity gaps. These results indicate that effective LLM distillation requires not only stronger teachers, objectives, or sampling strategies, but also constructing targets that the student can reliably internalize.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript formatting, figure visualization improvement, and literature search. The proposed method and its core mechanism were designed by the authors and were implemented with AI used as a coding assistant for debugging and code refinement. Generative AI was also used to improve the grammar and readability of the manuscript, assisting with presenting experimental results, and searching for potentially overlapping prior work.

All AI-assisted outputs described above were reviewed and verified by the authors. We take full responsibility for all claims, interpretations, and conclusions in this work.

REPRODUCIBILITY STATEMENT

We describe the CoKD method in Section 3 and report all datasets, models, baselines, and evaluation metrics in Section 4.1. Full hyperparameters, training configurations, and baseline implementation details are provided in Appendix F. We provide additional analyses, implementation details, and ablations in Appendices A–K. Source code to reproduce the CoKD framework and experimental results is anonymously available at https://anonymous.4open.science/r/Collaborative-Knowledge-Distillation-B855.

ETHICS STATEMENT

This work studies knowledge distillation methods using publicly available datasets (GSM8K, Databricks-Dolly-15k, and SAMSum) and open-weight language models. It does not involve human-subjects experimentation, collection of personal data, or deployment in high-stakes settings. We do not identify additional ethics concerns beyond the standard responsible use of pretrained language models.

REFERENCES

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.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4794–4802, 2019.

Mike Conover, Matt Hayes, Ankit Mathur, Jianxing Xie, Jun Wan, Sam Shah, Ali Ghodsi, Haoyang Wu, and Matei Zaharia. Free dolly: Introducing the world's first truly open instruction-tuned LLM. Technical report, Databricks, 2023.

Bogdan Gliwa, Ireneusz Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, 2019.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. In *Findings of ACL*, pages 8003–8017, 2023.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, 2024.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *International Conference on Machine Learning*, 2025.

Jahyun Koo, Yerin Hwang, Yongil Kim, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. SWITCH: Studying with teacher for knowledge distillation of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3733–3746, 2025.

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*, pages 6121–6133, 2020.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. Selective knowledge distillation for neural machine translation. In *Proceedings of ACL-IJCNLP*, pages 6456–6466, 2021.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57905–57923, 2024.

Jaehyeok Lee, Keisuke Sakaguchi, and JinYeong Bak. Self-training meets consistency: Improving LLMs' reasoning with consistency-driven rationale evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10519–10539, 2025. https://doi.org/10.18653/v1/2025.naacl-long.528.

Hendra Setiawan. Accurate knowledge distillation via n-best reranking. In *Proceedings of NAACL*, pages 1330–1345, 2024.

Jun Wang, Eleftheria Briakou, Hamid Dadkhahi, Rishabh Agarwal, Colin Cherry, and Trevor Cohn. Don't throw away data: Better sequence knowledge distillation. In *arXiv preprint arXiv:2407.10456*, 2024.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. In *arXiv preprint arXiv:2402.13116*, 2024.

Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, and Xue Liu. Teacher-student architecture for knowledge distillation: A survey. In *arXiv preprint arXiv:2308.04268*, 2023.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of AISTATS*, pages 627–635, 2011.

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.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, 2025.

Chuanpeng Yang, Wang Lu, Yao Zhu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 16(6):1–27, 2025.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of EMNLP*, pages 3134–3154, 2023.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Zhuofeng Wu, He Bai, Aonan Zhang, Jiatao Gu, V. G. Vinod Vydiswaran, Navdeep Jaitly, and Yizhe Zhang. Divide-or-conquer? Which part should you distill your LLM? In *Findings of EMNLP*, 2024.

Jiaheng Liu, Chenchen Zhang, Jinyang Guo, Yuanxing Zhang, Haoran Que, Ken Deng, Zhiqi Bai, Jie Liu, Ge Zhang, Jiakai Wang, Yanan Wu, Congnan Liu, Jiamang Wang, Lin Qu, Wenbo Su, and Bo Zheng. DDK: Distilling domain knowledge for efficient large language models. In *Advances in Neural Information Processing Systems*, 2024.

Cheng Feng, Chaoliang Zhong, Jun Sun, and Yusuke Oishi. Following the teacher's footsteps: Scheduled checkpoint distillation for domain-specific LLMs. In *arXiv preprint arXiv:2601.10114*, 2026.

Mohsen Gholami, Mohammad Akbari, Cindy Hu, Vaden Masrani, Z. Jane Wang, and Yong Zhang. GOLD: Generalized knowledge distillation via out-of-distribution-guided language data generation. In *Findings of NAACL*, 2024.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19287–19297, 2026.

Chunting Zhou, Graham Neubig, and Jiatao Gu. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*, 2020.

Min Liu, Yu Bao, Chengqi Zhao, and Shujian Huang. Selective knowledge distillation for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13246–13254, 2023.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. In *Advances in Neural Information Processing Systems*, 2025.

Songming Zhang, Yunlong Liang, Shuaibo Wang, Wenjuan Han, Jian Liu, Jinan Xu, and Yufeng Chen. Towards understanding and improving knowledge distillation for neural machine translation. In *Proceedings of ACL*, pages 8062–8079, 2023.

Jingxuan Wei, Linzhuang Sun, Yichong Leng, Xu Tan, Bihui Yu, and Ruifeng Guo. Sentence-level or token-level? A comprehensive study on knowledge distillation. In *Proceedings of IJCAI*, pages 6531–6540, 2024.

Yimeng Zhuang and Mei Tu. Pretrained bidirectional distillation for machine translation. In *Proceedings of ACL*, pages 1132–1145, 2023.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-Divergence minimization for sequence-level knowledge distillation. In *Proceedings of ACL*, pages 10817–10834, 2023.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. In *arXiv preprint arXiv:2602.12275*, 2026.

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. In *arXiv preprint arXiv:2601.18734*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *arXiv preprint arXiv:2603.07079*, 2026.

Amirhossein Afsharrad, Amirhesam Abedsoltan, Ahmadreza Moradipari, and Sanjay Lall. On-policy distillation of language models for autonomous vehicle motion planning. In *arXiv preprint arXiv:2604.07944*, 2026.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. Technical report, 2026. https://openai.com/index/gpt-5-6/.

Anthropic. Introducing Claude Opus 5. Technical report, 2026. https://www.anthropic.com/news/claude-opus-5.

Matthieu Zimmer, Xiaotong Ji, Tu Nguyen, and Haitham Bou Ammar. Rethinking large language model distillation: A constrained Markov decision process perspective. In *arXiv preprint arXiv:2509.22921*, 2025.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. In *arXiv preprint arXiv:2604.00626*, 2026.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020.

Abhishek Panigrahi, Bingbin Liu, Sadhika Malladi, Sham Kakade, and Surbhi Goel. In good GRACEs: Principled teacher selection for knowledge distillation. In *International Conference on Learning Representations*, 2026.

Xurong Xie, Zhucun Xue, Jiafu Wu, Jian Li, Yabiao Wang, Xiaobin Hu, Yong Liu, and Jiangning Zhang. LLM-oriented token-adaptive knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 34070–34078, 2026.

Chen Zhang, Qiuchi Li, Dawei Song, Zheyu Ye, Yan Gao, and Yao Hu. Towards the law of capacity gap in distilling language models. In *Proceedings of the ACL*, pages 22504–22528, 2025.

Almog Tavor, Itay Ebenspanger, Neil Cnaan, and Mor Geva. Rethinking selective knowledge distillation. *arXiv preprint arXiv:2602.01395*, 2026.

A Analysis of CoKD's Effectiveness

A central design in CoKD is to construct distillation targets from teacher–student *collaborative decoding* rather than from either model alone. To understand why such targets are effective, we conduct a controlled test-set analysis of response quality under different decoding modes. We use the Qwen2.5-7B-Instruct → Qwen2.5-0.5B-Instruct pair on Dolly with the same SFT teacher and student checkpoints as in the main experiments. Specifically, we compare (i) student-only generation, (ii) teacher-only generation, and (iii) collaborative generation with $\alpha \in \{0.3, 0.5, 0.7\}$. For collaborative generation, we follow the standard test-time evaluation protocol, except that each next token is selected greedily according to the collaborative score in Eq. (1), which combines teacher and student predictions.

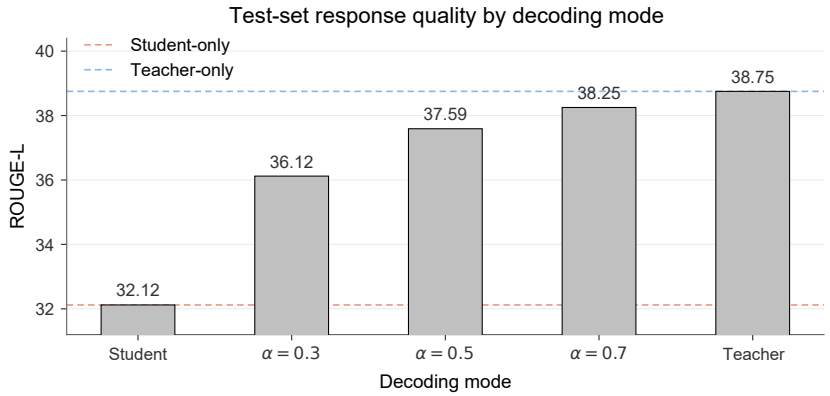

Figure 6: Test-set response quality on Databricks-Dolly-15k under different decoding modes with Qwen2.5-7B-Instruct → Qwen2.5-0.5B-Instruct pair. The dashed lines mark the student-only and teacher-only baselines. Collaborative decoding yields responses that are substantially stronger than student-only generation and progressively approaches teacher quality as α increases.

Figure 6 summarizes the resulting test-set ROUGE-L scores. Collaborative decoding yields consistent gains even under the student-heavy setting ($\alpha = 0.3$). As α increases, quality improves monotonically and progressively approaches teacher-only performance.These results explain the effectiveness of CoKD's target construction. During distillation, we train the student on collaboratively generated responses, which encourages the student to reproduce similar responses at inference time, thereby improving performance. Moreover, collaborative responses are sampled from the intersection of teacher preference and student support under Eq. (1), so they remain within the student's learnable range and are easy to internalize.

B Cross-Entropy vs. KL Divergence as the Training Objective

CoKD trains the student with standard autoregressive cross-entropy (CE) on accepted collaborative targets (Eq. (5)), rather than token-level KL or reverse-KL objectives commonly used in knowledge distillation (Hinton et al., 2015; Gu et al., 2024; Ko et al., 2024). We summarizes our rationale and provides an empirical comparison against distribution-matching losses on the same accepted targets.

Design rationale. Our motivation is threefold. First, the goal of LLM distillation is primarily behavioral imitation: given the same prompt, the student should generate the same response as the teacher. This goal does not require the student to match the teacher's full next-token distribution, since different distributions can still lead to the same decoded token and hence the same generated sequence. CE with a hard target therefore provides a direct supervision signal for reproducing the target token. Second, CoKD already incorporates teacher knowledge during target construction (§3.2). Given such targets, standard cross-entropy provides a simple, stable, and effective objective for enabling the student to absorb this knowledge. Third, KL-based objectives encourage the student to match the teacher's full next-token distribution at every position. Under a large teacher–student capacity gap, this requirement is difficult to satisfy and may dilute the learning with low-probability teacher mass that the student cannot reliably reproduce (Gu et al., 2024; Wen et al., 2023). Figure 7

illustrates this distinction at a single decoding step. Panel (a) shows that forward KL requires the student distribution p_S to match the teacher distribution p_T over the full vocabulary: every token probability contributes to the loss, including low-mass tail tokens that are difficult for a small student to reproduce. Panel (b) shows that CE with a hard target y^* only requires the student to assign the largest probability mass to the target token and does not require the student to reproduce the teacher's relative probabilities over non-target tokens. This aligns with the behavior-level goal of LLM generation, where producing the same token does not require identical predictive distributions. Thus, CE serves as a less restrictive and more student-reachable objective.

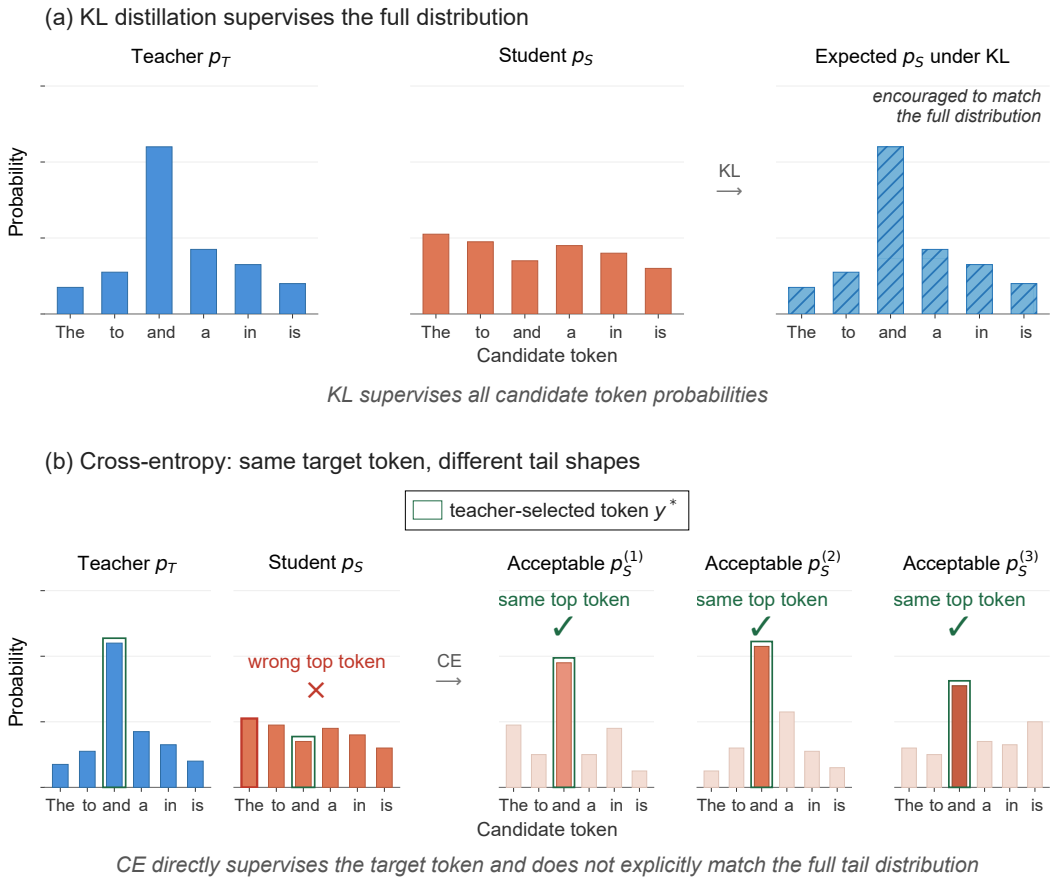

Figure 7: Comparison between KL-based and CE-based token-level distillation. **(a)** KL encourages the student to match the teacher's full next-token distribution, thereby penalizing mismatch across all candidate tokens. **(b)** Cross-entropy with a hard teacher target directly supervises only the selected token. As a result, multiple student distributions can assign the highest probability to the same target token while differing in their non-target probabilities.

C Teacher Preference versus Student Reachability of Distillation Targets

To connect target construction with downstream distillation quality, we analyze the teacher preference and student reachability of sequences produced by SeqKD, SKD, and CoKD. During distillation training, we measure, for each token in the constructed targets, its rank under the teacher and student next-token distributions, where rank 1 denotes each model's top prediction. Then we average these ranks over all such tokens to obtain $\bar{r}_T$ and $\bar{r}_S$ (Figure 8). We exclude tokens with probability > 0.99 under both the teacher and the student when computing $\bar{r}_T$ and $\bar{r}_S$, since autoregressive sequences contain a lot of near-deterministic tokens that contribute little to distillation. Excluding them makes rank differences among distillation methods clearer. Lower ranks indicate stronger teacher preference or student reachability, so the ideal target lies near $(\bar{r}_T, \bar{r}_S) = (1, 1)$. We evaluate on GSM8K, Dolly,

and SAMSum with Qwen2.5 teachers at 3B, 7B, and 14B and a fixed Qwen2.5-0.5B-Instruct student, as shown in Figure 8.

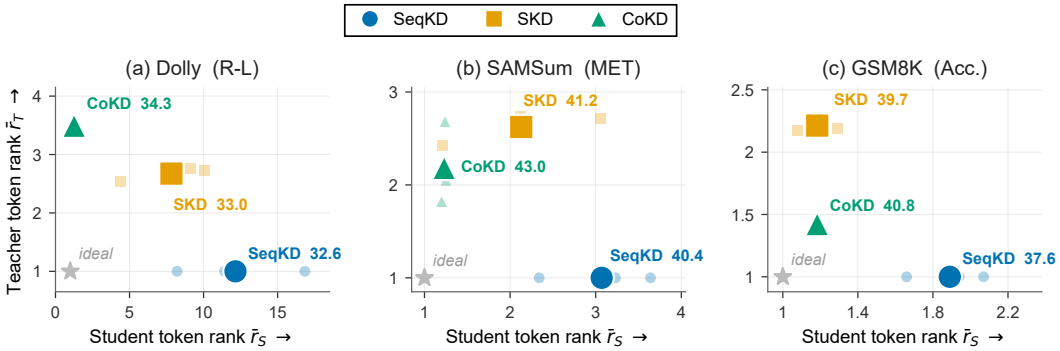

Figure 8: Teacher preference ($\bar{r}_T$) versus student reachability ($\bar{r}_S$) of constructed distillation targets. Lower is better in both axes; the star marks the ideal $(\bar{r}_T, \bar{r}_S) = (1, 1)$. Faded markers report individual teacher sizes and bold markers average over sizes, annotated with the corresponding mean student task score.

Figure 8 reveals a consistent gap between teacher preference and student reachability across baselines. SeqKD constructs targets through teacher greedy decoding, so each selected token is the teacher's top-1 prediction and $\bar{r}_T$=1 by construction. Yet these teacher-greedy trajectories can be poorly aligned with the student's distribution. On Dolly, $\bar{r}_S$ increases from approximately 8 to 17 as the teacher size grows from 3B to 14B, showing that higher teacher preference may come at the cost of lower student reachability. As a result, it may undercut distillation effectiveness, which aligns with the weak SeqKD performance in Table 1. SKD improves reachability through a student-proposes, teacher-verifies pipeline, but its targets remain suboptimal on both teacher preference and student reachability, with both $\bar{r}_T$ and $\bar{r}_S$ well above the ideal corner. CoKD targets remain student-reachable by design because sampling is confined to the admissible set in Eq. (2), and they retain moderate teacher preference because tokens are drawn from the collaborative distribution induced by the scores in Eq. (1). Accordingly, CoKD lies closer to the ideal region on all three tasks ($\bar{r}_S \approx 1.2$; $\bar{r}_T \approx 1.4$–3.6 across tasks), consistent with its strongest downstream performance in downstream task tasks. Overall, these results support our central claim that effective distillation requires targets that are both teacher-preferred and student-reachable, rather than teacher-optimal sequences that the student cannot reliably internalize.

D Limitations and Future Directions

While CoKD has demonstrated promising performance across different tasks and teacher–student configurations, it has two main limitations. First, CoKD incurs additional training-time computational cost during target construction. Unlike SeqKD and GKD (Kim and Rush, 2016; Agarwal et al., 2024), collaborative decoding requires evaluating both the teacher and student autoregressively at each generation step, resulting in higher target-generation latency. Such teacher–student computation during distillation is also common in recent on-policy or interactive KD methods, including MiniLLM, SKD, and SWITCH (Gu et al., 2024; Xu et al., 2025; Koo et al., 2025). The main incremental expense of CoKD is the sequence-level quality-control pass, which requires an additional student rollout for each sample. To mitigate the computational cost, a promising future direction is to progressively reduce teacher involvement and shift target generation toward the student itself. In particular, the quality-control mechanism could provide feedback for a self-training or reinforcement-learning loop in which the student generates and evaluates upon its own trajectories, requiring teacher guidance only when necessary. Recent self-training and self-rewarding approaches have shown that LLMs can improve from their own generated responses and feedback (Yuan et al., 2024; Lee et al., 2025).

Second, the current quality-control mechanism of CoKD relies on a reference-grounded utility function and therefore assumes that a gold reference is available for each training example. This restricts the direct applicability of CoKD to unlabeled prompts, for which methods such as SeqKD or GKD can still construct distillation signals without reference responses. One possible solution is to generate synthetic reference responses using strong frontier models, such as GPT-5.6 Sol (OpenAI,

2026) or Claude Opus 5 (Anthropic, 2026), before applying CoKD. The use of LLM-generated supervision has been shown to be effective in prior instruction-tuning and distillation studies (Wang et al., 2023; Hsieh et al., 2023). However, such synthetic references may themselves contain errors or biases. In addition, although sequence-level quality control is consistently beneficial within CoKD, applying the same procedure to other KD methods may not yield uniform improvements (Appendix I), indicating that an effective filter must be aligned with how each method constructs targets rather than serving as a generic post-processing step. This observation is consistent with broader evidence that students benefit from selectively absorbing teacher supervision rather than imitating it uniformly (Huang et al., 2026). Developing general quality-control mechanisms—whether reference-free (e.g., via learned verifiers or reward models) or transferable across distillation pipelines—that can selectively retain student-beneficial teacher signals remains an important direction for future work.

E CoKD Algorithm

Algorithm E.1 Collaborative Knowledge Distillation (CoKD)

Require: Dataset $\mathcal{D} = \{(x_i, y_i^\star)\}_{i=1}^N$, teacher p_T, student p_S, collaboration weight α, reachability threshold ρ, temperature T, quality control threshold δ_τ

for each $(x, y^\star) \in \mathcal{D}$ **do**
  Generate collaborative response y^{co} autoregressively:
  for $t = 1, 2, \ldots$ until EOS **do**
    $h_t \leftarrow (x, y^{\text{co}}_{<t})$
    $\mathcal{A}_t \leftarrow \{v \in \mathcal{V} \mid p_S(v \mid h_t) \geq \rho\}$
    $s_t(v) \leftarrow \alpha \log p_T(v \mid h_t) + (1-\alpha) \log p_S(v \mid h_t)$ for all $v \in \mathcal{A}_t$
    $q_{\text{co}}(v \mid h_t) \propto \exp(s_t(v)/T)$ for all $v \in \mathcal{A}_t$
    Sample $y_t^{\text{co}} \sim q_{\text{co}}(\cdot \mid h_t)$
  end for
  Generate student-only response $y^S \sim p_S(\cdot \mid x)$
  if $U_\tau(x, y^{\text{co}}, y^\star) > U_\tau(x, y^S, y^\star)$ or $U_\tau(x, y^{\text{co}}, y^\star) \geq \delta_\tau$ **then**
    Update p_S with supervised fine-tuning on y^{co}
  end if
end for

F Training Details and Baseline Implementation

F.1 Training Details and Hyperparameters

Before applying distillation, we first fine-tune both the teacher and the student on the training set of each task for 2 epochs with a learning rate of 2×10^{-5}, obtaining task-adapted teacher and student initializations. All distillation methods are then initialized from the SFT student and use the corresponding SFT teacher as the teacher model. Unless otherwise specified by method-specific settings, all distillation methods use the same optimization hyperparameters, summarized in Table 5. Task-specific training configurations are reported in Table 6.

Table 5: Task-agnostic hyperparameters used for distillation.

Hyperparameter	Value
Learning rate	1×10^{-5}
Epoch	1
Warmup ratio	0.1
Learning rate scheduler	Cosine
Data type	bfloat16
Optimizer	AdamW
Weight decay	0.1

Table 6: Task-specific training configurations used for distillation.

Task	Training steps	Save steps	Max length
Dolly	14K	2500	768
GSM8K	7.5K	2000	768
SAMSum	2K	500	1024

F.2 Baseline Implementation

We provide the implementation details of the KD baselines used in our experiments, including logits-based KD, SeqKD, GKD, SKD and SWITCH.

Logits-based KD. Logits-based KD performs token-level distribution matching on the gold reference responses. At each decoding step, both the teacher and the student are conditioned on the gold prefix, and the student is optimized to match the teacher's next-token logits. We use reverse KL divergence as the training objective, since it consistently performs better than forward KL in our preliminary experiments.

SeqKD. For SeqKD, we follow the standard sequence-level distillation pipeline. For each training input, the teacher greedily generates a response, which is then used as the distillation target. The student is fine-tuned on these teacher-generated responses with the standard cross-entropy loss.

GKD. GKD performs on-policy distribution matching by training the student against teacher feedback along student-generated trajectories. We implement GKD in the on-policy regime by setting the student-data fraction to $\lambda = 1.0$, so training uses only generation-conditioned trajectories from student rollouts rather than the mixed pipeline ($\lambda = 0.5$) that combines reference- and generation-conditioned supervision. In our experiments, on-policy GKD consistently outperformed the mixed setting. Following the mode-seeking configuration in Agarwal et al. (2024), we use generalized Jensen–Shannon divergence as the distillation loss and set $\beta = 0.9$.

SKD. SKD constructs distillation trajectories through speculative decoding, in which the student drafts candidate tokens and the teacher verifies or replaces them. We use the on-policy setting with $\lambda = 1$: the student drafts candidate continuations with temperature $T = 0.2$ and top-$p = 0.5$, and the teacher verifies the drafted tokens with top-$k = 25$. The student is then trained on the verified trajectories using reverse KL divergence.

SWITCH. SWITCH builds distillation trajectories through token-level teacher intervention. At each decoding step, it compares the teacher and student next-token distributions via normalized Jensen–Shannon divergence and lets the student sample when the divergence is below a position-dependent threshold, otherwise the teacher samples. We use the exponentially decaying threshold $\tau_t = \tau_0 e^{-\rho t}$ with $\tau_0 = 1$ and decay rate $\rho = 0.1$, and sample from the active model's distribution at temperature $T = 1.0$. Following the paper's default training schedule, we set the generation-conditioned fraction to $\lambda = 0.5$, so half of the updates use SWITCH-generated trajectories and the remainder use reference supervision. The student is trained on the resulting sequences with skewed reverse KL, $\mathrm{KL}(q_{\text{student}} \,\|\, (1-\alpha)p_{\text{teacher}} + \alpha q_{\text{student}})$, using $\alpha = 0.1$.

G Details of CoKD implementation

Below, we describe additional implementation details of CoKD: sampling student-only rollouts, handling an empty admissible set during collaborative sampling, defining the task-aware utility function U_τ, and filtering ground-truth-matching targets before distillation.

G.1 Student Rollout Sampling

For sequence-level quality control, we sample the student-only rollout y^S from the current student model with temperature $T_S = 0.9$ across all tasks. This temperature applies only to the student rollout and is independent of the collaborative-decoding temperature T.

G.2 Fallback for Reachability-constrained sampling

In the extreme case where the student's maximum next-token probability is strictly below the reachability threshold, i.e., $\max_{v \in \mathcal{V}} p_S(v \mid h_t) < \rho$, the admissible set in Eq. (2) is empty. We then set $\mathcal{A}_t = \{\hat{v}_t\}$ as fallback solution, where $\hat{v}_t = \arg\max_{v \in \mathcal{V}} p_S(v \mid h_t)$, thereby ensuring that the collaborative distribution in Eq. (3) remains well-defined.

G.3 Task-Specific Utility Definitions

We instantiate the task-aware utility function $U_\tau(x, y, y^\star)$ using the standard evaluation criterion of each task, measuring sequence-level similarity between a generated response and the reference to avoid training on targets substantially misaligned with the gold references. For arithmetic reasoning, we use exact-match accuracy between the generated answer and the reference answer. For instruction following and summarization, we use ROUGE-L between the generated output and the reference response. More generally, U_τ can be replaced by any task-appropriate supervised metric, provided that it can reliably compare the collaborative response and the student-only rollout for the same input.

G.4 Filtering Ground-Truth-Matching Sequences

Before filtering collaboratively generated targets with the sequence-level quality rule in Eq. (4), we discard those that exactly match the ground-truth response $y^\star$. Since the student is already initialized by SFT on the ground-truth references, these reference-identical targets provide limited additional supervision and may dilute the distillation signals during the distillation stage. Filtering them encourages the student to focus on collaborative targets that contain additional teacher-preferred yet student-reachable knowledge beyond the original references. We found in preliminary experiments that this filtering step improves downstream performance, indicating that removing redundant targets makes the distillation signal more effective.

H Teacher Performance across Model Sizes

Table 7 reports the performance of Qwen2.5-Instruct teachers at 1.5B, 3B, 7B, and 14B parameters, alongside the fixed SFT student (Qwen2.5-0.5B-Instruct). These scores characterize the teacher–student performance gap under each teacher size and help interpret the teacher-scaling results in Figure 3.

Table 7: Teacher performance across model sizes with a fixed Qwen2.5-0.5B-Instruct SFT student. Student scores are included here for reference.

Model	Dolly (ROUGE-L)	SAMSum (METEOR)	GSM8K (Acc.)
SFT student (0.5B)	32.12	40.94	37.30
Teacher (1.5B)	35.60	43.74	59.51
Teacher (3B)	37.00	45.25	67.40
Teacher (7B)	38.75	46.68	75.82
Teacher (14B)	39.86	48.16	84.99

On Dolly and SAMSum, the performance gap between teacher and student is relatively narrow, especially for the teacher–student pair of Qwen2.5-1.5B-Instruct and Qwen2.5-0.5B-Instruct.

I Effect of Sequence-Level Quality Filtering

To evaluate the contribution of sequence-level quality filtering (§3.3), we provide a controlled comparison in which all response-based KD methods have access to the same filtering mechanism. Specifically, we apply CoKD's quality-control procedure to SeqKD, SKD, and SWITCH, whose generate-then-train pipelines naturally support such post-generation filtering. For each baseline, we leave its response-generation and student-training processes unchanged, but filter its generated targets using the task-aware acceptance rule in Eq. (4).

Method	Dolly (R-L)	SAMSum (MET)	GSM8K (Acc.)
SeqKD	33.23 (↑ 0.45)	40.84 (↑ 0.26)	38.51 (↑ 0.15)
SKD	33.18 (↓ 0.59)	41.46 (↓ 0.25)	38.38 (↓ 1.04)
SWITCH	33.97 (↑ 0.44)	41.21 (↑ 0.13)	38.21 (↓ 0.30)
CoKD (w/o filter)	33.40	40.07	37.68
CoKD (full)	**34.61**	**43.08**	**40.71**

Table 8: Effect of applying CoKD's sequence-level quality filter to response-based KD baselines under Qwen2.5-7B-Instruct → Qwen2.5-0.5B-Instruct. SeqKD, SKD, and SWITCH retain only targets that satisfy the same reference-grounded quality-control procedure as CoKD. Parenthetical arrows report changes relative to the corresponding baselines in Table 1, where blue upward arrows denote improvements and orange downward arrows denote degradations.

Table 8 shows that applying CoKD's reference-grounded quality filter to other KD baselines has method- and task-dependent effects, ranging from slight improvements to performance degradation. This variability is unsurprising because these baselines follow distinct target-construction principles, and retrofitting an external filter changes their intended operation while potentially discarding substantial training data (e.g., approximately 40% for SKD). For these methods, target reachability may be more consequential than reference-grounded quality alone, especially when the teacher primarily drives target construction. In contrast, quality filtering is a natural complement to CoKD: because the student plays a central role in constructing reachable training targets, the resulting targets may include low-quality responses that the filter is designed to remove. Accordingly, removing sequence-level quality control consistently degrades CoKD, while full CoKD outperforms all filtered baselines, indicating that its reachability-aware construction and reference-grounded filtering operate jointly rather than as interchangeable components.

J Ablation over Sequence-Level Quality Threshold

The threshold δ_τ in Eq. (4) determines when a collaboratively generated response is retained even if it does not strictly improve over the student-only rollout; at $\delta_\tau = 1.0$, only targets that strictly outperform the student rollout are accepted. Figure 9 sweeps δ_τ on Dolly and SAMSum while fixing the remaining hyperparameters to their main-experiment settings. For each threshold, we report the downstream task score and the accepted-target ratio after sequence-level filtering.

This ablation exposes a trade-off between training-sample quality and quantity: a higher δ_τ retains higher-quality collaborative targets but accepts fewer of them. Tightening δ_τ therefore reduces the accepted-target ratio on both Dolly and SAMSum, as expected from the sequence-level quality rule in §3.3. Task performance is non-monotonic and peaks at an intermediate threshold rather than at the most permissive or strictest setting. Notably, the best δ_τ differs across tasks, indicating that this hyperparameter should be tuned on the validation split for each task type. Overall, the quality-control step filters collaboratively generated targets that are reachable yet unhelpful under the task objective, helping CoKD retain useful supervision without accepting every collaborative response.

K Additional Ablation over Collaboration Weight α

K.1 GSM8K

The GSM8K results presented in Figure 10 are broadly consistent with the Dolly ablation in Figure 4, where accuracy peaks at an intermediate α, indicating that purely teacher- or student-dominated

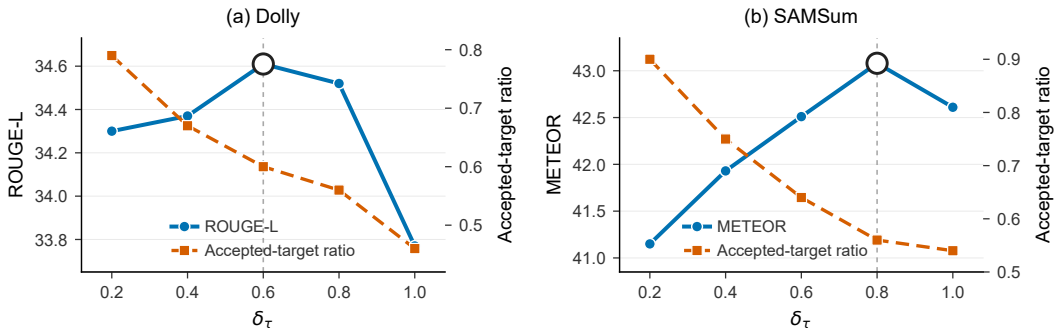

Figure 9: Ablation of the sequence-level quality threshold δ_τ. **(a)** Dolly ROUGE-L and accepted-target ratio. **(b)** SAMSum METEOR and accepted-target ratio. The ring marks the best task score in each panel; the vertical dashed line indicates the corresponding δ_τ. Lower thresholds retain more collaborative targets, whereas higher thresholds enforce stricter filtering.

collaborative targets are suboptimal. As α increases, the accepted-target ratio also rises because teacher-dominated trajectories produce higher-quality responses that more often satisfy the quality-control rule. However, more distillation samples does not necessarily improve distillation performance. This again suggests that effective target construction depends on whether the resulting targets remain reachable for the student.

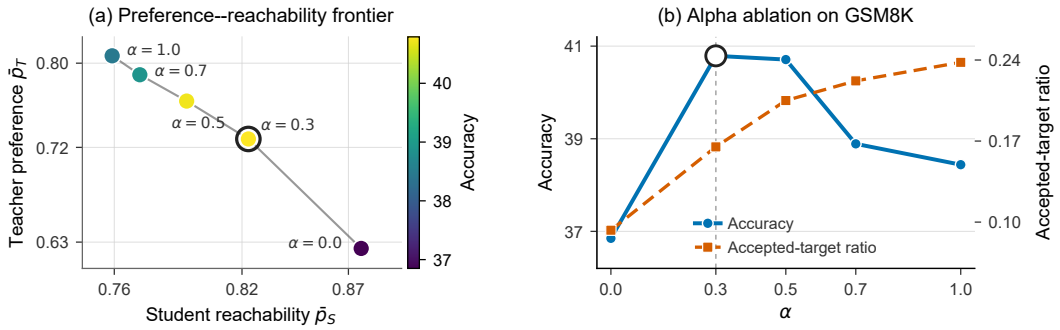

Figure 10: Ablation of the collaboration weight α on GSM8K. For preference–reachability frontier, the reported $\bar{p}_T$ and $\bar{p}_S$ values are relatively high because autoregressive sequences contain a large portion of near-deterministic tokens to which both models assign large probability mass, as also observed in Figures 4 and 11. Appendix K.2 further verifies this effect through binned target-token probability profiles. **(b)** GSM8K accuracy and accepted-target ratio under the same α sweep. The accepted-target ratio here is lower than those on Dolly and SAMSum because GSM8K uses exact-match accuracy as U_τ, which makes the acceptance rule in Eq. (4) stricter. For example, a collaborative trajectory is retained only when it predicts the correct answer while the student-only rollout does not.

K.2 SAMSum

For SAMSum, the effect of α differs from Dolly and GSM8K and emphasizes the role of student reachability. As shown in Figure 11, increasing α improves teacher preference and raises the accepted-target ratio, but reduces student reachability and leads to lower downstream METEOR. The best performance is obtained at the most student-dominated setting, where the student is trained on targets that remain close to its own rollouts and are retained by the task-aware quality control mechanism. One likely explanation is that dialogue summarization is comparatively easier than instruction following on Dolly and arithmetic reasoning on GSM8K (Table 1), so the SFT student can already produce high-quality rollouts that remain the most reachable targets, with task-aware quality control filtering out the remaining low-quality responses. This observation is consistent with the teacher-scaling results in Figure 3, where SAMSum performance consistently decreases as the teacher size grows and peaks with the 1.5B teacher, which has the narrowest teacher–student performance gap. By contrast, on Dolly and GSM8K the student benefits more from teacher participation to construct responses that are sufficiently high-quality for learning. These results suggest that, for

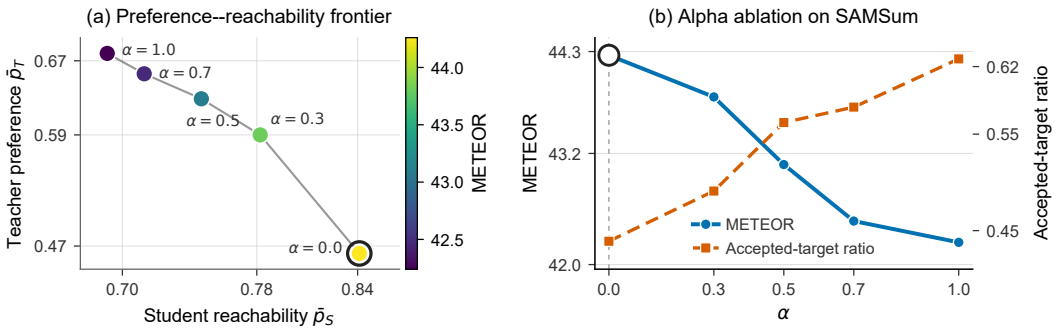

Figure 11: Ablation of the collaboration weight α on SAMSum. Unlike Dolly and GSM8K, SAMSum favors lower α, indicating that student reachability is especially important in the low-resource summarization setting.

SAMSum, stronger teacher signals are not necessarily more beneficial: when the teacher becomes too dominant, the resulting targets may be less aligned with the student's current capacity and harder to absorb.

K.3 Token Probability Profiles under Varying α

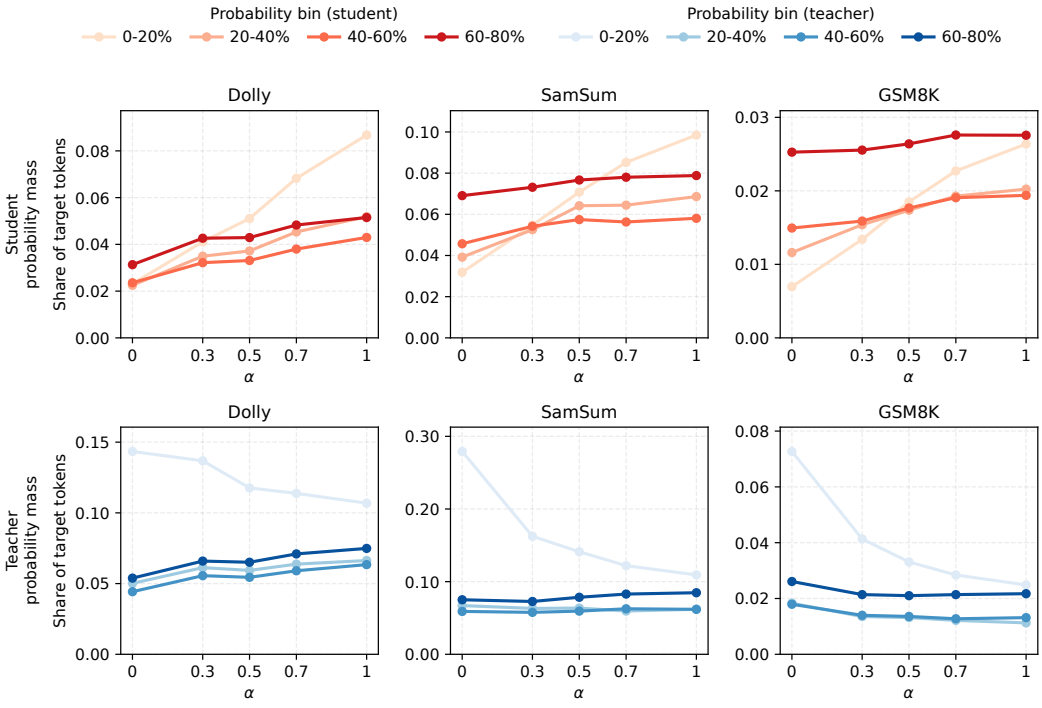

Figure 12: Probability profiles of collaborative-target tokens under different collaboration weights α. For each selected target token, we record its probability under the student and teacher distributions and report the share of target tokens falling into each probability bin. The top row shows student probabilities and the bottom row shows teacher probabilities across Dolly, SAMSum, and GSM8K. The 0.8–1.0 probability bin is omitted from the plot to better highlight changes among non-saturated tokens; its share can be inferred from the total mass of the displayed 0–0.8 bins.

Figure 12 provides a distribution-level view of the preference–reachability trade-off induced by α. Across tasks, the most visible change occurs in the lowest probability bin. As α increases, the share of selected tokens with low student probability increases, indicating that teacher-dominated decoding increasingly pulls targets toward regions that are harder for the student to reproduce. Conversely, the

share of selected tokens with low teacher probability decreases, showing that larger α suppresses teacher-disfavored candidates. The remaining probability bins are comparatively stable, suggesting that α mainly controls the amount of low-support mass on each side rather than uniformly reshaping the entire probability profile. Thus, the movement along the preference–reachability frontier in the α ablation is driven primarily by changes in low-support target tokens.

This observation complements the aggregate results in the α ablation. Larger α improves teacher alignment by reducing low-teacher-probability targets, but it also introduces more low-student-probability targets, which can make the constructed responses less reachable for the student. Therefore, the effect of α is not simply to improve or degrade target quality; instead, it shifts the composition of collaborative targets along the teacher-preference versus student-reachability axis. This helps explain why intermediate α usually yields better performance.