TwT: Think without Tokens by Habitual Reasoning Distillation with Multi-Teachers' Guidance

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have made significant strides in problem-solving by incorporating reasoning processes. However, this enhanced reasoning capability results in an increased number of output tokens during inference, leading to higher computational costs. To address this challenge, we propose ***TwT*** (Thinking without Tokens), a method that reduces inference-time costs through habitual reasoning distillation with multi-teacher guidance, while maintaining high performance. Our approach introduces a Habitual Reasoning Distillation method, which internalizes explicit reasoning into the model's habitual behavior through a Teacher-Guided compression strategy inspired by human cognition. Additionally, we propose Dual-Criteria Rejection Sampling (DCRS), a technique that generates a high-quality and diverse distillation dataset using multiple teacher models, making our method suitable for unsupervised scenarios. Experimental results demonstrate that TwT effectively reduces inference costs while preserving superior performance, achieving up to a 13.6% improvement in accuracy with fewer output tokens compared to other distillation methods, offering a highly practical solution for efficient LLM deployment.

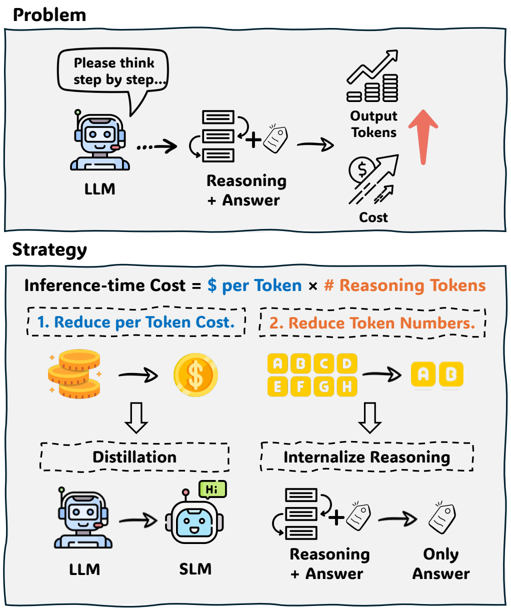

Figure 1: **Overview of the Proposed Method.** The upper part of the figure illustrates the background problem: while generating more reasoning steps improves the performance of LLMs, it also leads to significantly higher computational costs. To mitigate this, we propose targeted strategies, shown in the lower part of the figure. Our approach reduces the cost per token by distilling knowledge from large models into smaller ones and minimizes the total number of tokens by gradually shrinking intermediate reasoning paths.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable improvements in problem-solving by incorporating reasoning process (Brown et al., 2020; Wei et al., 2022; Chowdhery et al., 2023; Yao et al., 2024). It enhances the reasoning capability of LLMs by breaking down complex tasks into intermediate steps, leading to better performance. However, reasoning capability comes at a significant cost: the reasoning process substantially increases the number of output tokens during inference, resulting in higher inference-time computational costs(Snell et al., 2024; Wu et al., 2024). In practical deployments, computational resources and budgets are often constrained, making the reduction of inference-time computational costs a pressing issue that requires effective solutions (Zheng et al., 2022; Chung et al., 2024).

Recent research has primarily explored two key approaches to addressing this issue: reducing the cost per token and reducing the number of reasoning tokens as illustrated in Figure 7 (Wang et al., 2024a; Hsieh et al., 2023; Wang, 2024). A common strategy for reducing the cost per token is to replace large models with smaller, more efficient ones. However, directly using smaller models often results in a significant drop in performance, particularly on complex reasoning tasks. To mitigate this problem, knowledge distillation (Hinton, 2015; Park et al., 2019) has emerged as an effective solution, enabling student models to mimic the performance of larger teacher models (Tang et al., 2019; Hsieh et al., 2023). However, traditional knowledge distillation typically relies on expensive human-labeled data, limiting its practical applicability. Additionally, most distillation methods employ only a single teacher model, restricting the diversity of knowledge that could be leveraged from multiple teacher models. To overcome these limitations, we propose Dual-Criteria Rejection Sampling (DCRS), a method that first utilizes a multi-teacher strategy to generate pseudo-labels and then applies a two-stage selection process-Quality Selection and Diversity Selection-to construct a high-quality and diverse distillation dataset. This approach not only enhances the efficiency of knowledge transfer but also enables our model to adapt effectively to unsupervised settings.

For the other issue, the number of reasoning tokens can be reduced by shrinking intermediate reasoning paths (Wang et al., 2024a; Deng et al., 2024). While this effectively lowers token usage, it often comes at the cost of degraded model performance. Therefore, it is essential to develop a method that balances computational efficiency and model accuracy. Consider a real-world learning scenario where a teacher possesses a deep and comprehensive understanding of a concept, while a student may struggle to grasp the material fully. To bridge this gap, the teacher distills knowledge, extracting only the most essential and concise information to help the student learn more effectively. Over time, the student internalizes this reasoning process, enabling them to generate answers instantly upon encountering a question, without requiring explicit intermediate reasoning steps.

Inspired by this human cognitive process, we propose Habitual Reasoning Distillation (HaRD), a method that internalizes explicit reasoning into the model's habitual behavior through a multi-stage distillation process, thereby reducing the need for explicit reasoning during inference. HaRD follows a three-stage distillation strategy: (a) Full Reasoning Distillation, where the student learns reasoning patterns from complete reasoning paths generated by teacher models; (b) Reasoning-Compressed Distillation, where the reasoning process is progressively compressed, with teachers refining their outputs based on the student's responses to create reasoning paths aligned with the student's capabilities; and (c) Reasoning-Free Distillation, where the student is trained without explicit reasoning steps, relying only on final labels, allowing it to generate high-quality answers directly. This process shifts the computational burden from inference to training, enabling both high performance and low inference cost.

In this work, we propose ***TwT*** (Thinking without Tokens), a method that achieves an optimal balance between inference-time computational cost and performance. TwT follows a two-step process. First, DCRS utilizes multi-teacher LLMs to generate pseudo-labels, enabling the model to adapt to unsupervised settings. Then, HaRD applies a multi-stage distillation approach to progressively internalize explicit reasoning abilities into the student model as inherent capabilities. Our key contributions are summarized as follows:

- **Novel Distillation Framework:** We propose ***TwT***, a novel framework that aims to reduce inference-time computational cost through habitual reasoning distillation with multi-teachers' guidance while preserving high performance.
- **Unsupervised Sampling Strategy:** We propose Dual-Criteria Rejection Sampling (DCRS), a method that selects high-quality and diverse distillation data generated by multi-teacher LLMs, enabling adaptation to unsupervised settings.
- **Efficient Reasoning Distillation:** We design a Habitual Reasoning Distillation (HaRD) method that refines reasoning patterns through a teacher-guided compression strategy, ensuring better alignment with the student model's capabilities and ultimately integrates explicit reasoning into the model's inherent behavior.
- **Comprehensive Empirical Validation:** Experimental results demonstrate that our approach outperforms existing distillation techniques, achieving up to a 13.6% improvement in performance while generating fewer tokens.

2 Related Work

2.1 Knowledge Distillation for LLMs

Knowledge distillation (KD) (Hinton, 2015; Beyer et al., 2022; West et al., 2021) transfers capabilities

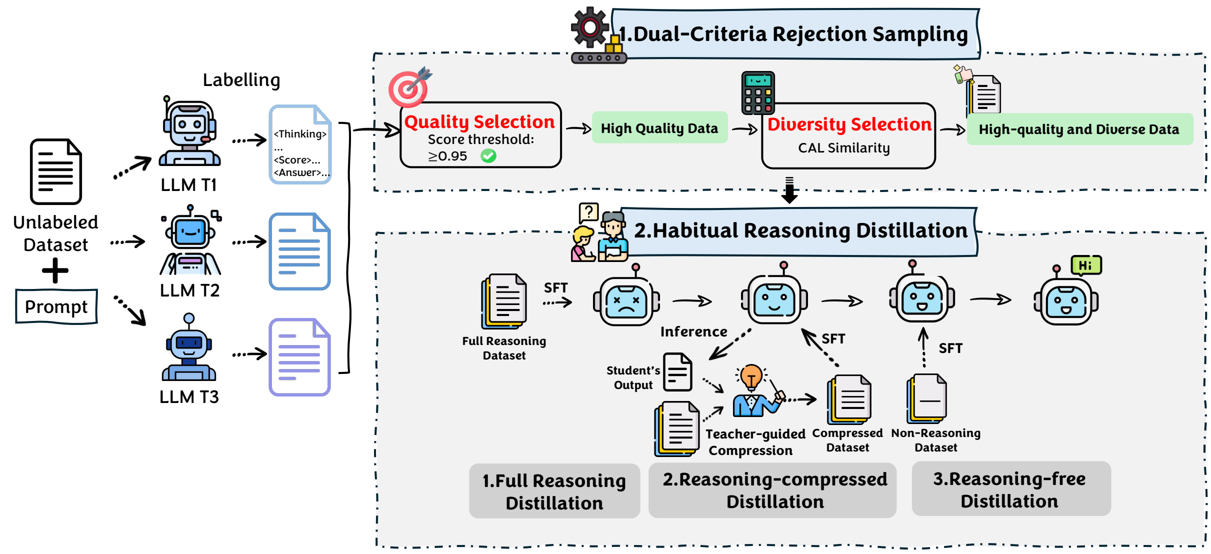

Figure 2: **Method Framework.** Our proposed ***TwT*** (Thinking without Tokens) framework consists of two stages: Dual-Criteria Rejection Sampling (**DCRS**) and Habitual Reasoning Distillation (**HaRD**). In the first stage, DCRS selects a high-quality and diverse reasoning distillation dataset generated by multiple teacher LLMs (e.g., T1, T2, T3). In the second stage, HaRD progressively internalizes reasoning ability into the student model through a three-stage distillation process.

from large LLMs to smaller ones using hard or soft labels from a teacher model. However, by relying solely on final outputs, standard KD provides limited information. Reasoning distillation (Hsieh et al., 2023) addresses this by training students to understand both the final answer and the underlying reasoning. Recent work (Chen et al., 2023; Liu et al., 2023) has also focused on generating multiple rationales per query to enhance prediction robustness. Yet, deriving rationales from a single teacher can introduce biases and reduce diversity. Multi-teacher strategies (Tian et al., 2024; Zhang et al., 2024) mitigate this by aggregating diverse reasoning paths, enriching the distillation dataset, and improving generalization. These methods select higher-quality distilled data by comparing predictions with ground truth, which, however, requires labeled datasets that are often hard to obtain. Our work introduces a multi-teacher approach to incorporate diverse reasoning data and proposes a Dual-Criteria Rejection Sampling strategy to obtain a high-quality and diverse distillation dataset from unlabeled data.

2.2 Reasoning and Inference-time Scaling

Recent work enhances output diversity by improving reasoning via structural methods (e.g., code parsing, problem decomposition (Gao et al., 2023; Zhou et al., 2022)) and by generating multiple reasoning paths through techniques like majority voting and reinforcement learning (Wei et al., 2022; Yao et al., 2023; Cao, 2024; Wang et al., 2022; Fu et al., 2022; Huang et al., 2023; Trung et al., 2024; Wang et al., 2024b). However, these supervised approaches rely on a single model and labeled data, limiting inherent diversity. Additionally, studies such as (Snell et al., 2024; Wu et al., 2024) show that adaptive inference-time strategies can significantly reduce compute costs, but model performance will be reduced. To address these issues, we propose a multi-teacher strategy, an unsupervised approach for generating diverse, high-quality samples with habitual reasoning distillation for efficient inference with explicit reasoning.

3 Method

In this section, we provide a detailed explanation of the implementation of our TwT as illustrated in Fig. 2. First, we propose a Dual-Criteria Rejection Sampling (DCRS) strategy to obtain high-quality and diverse distillation samples (Section 3.1). Then, we design a **H**abitual **R**easoning **D**istillation (HaRD) strategy that progressively internalizes the reasoning ability at each distillation stage, allowing the reasoning capabilities of the teacher models to be gradually internalized into the student model (Section 3.2).

3.1 Dual-Criteria Rejection Sampling

To provide the student model with high-quality and diverse reasoning paths, we propose a novel paradigm termed Dual-Criteria Rejection Sampling

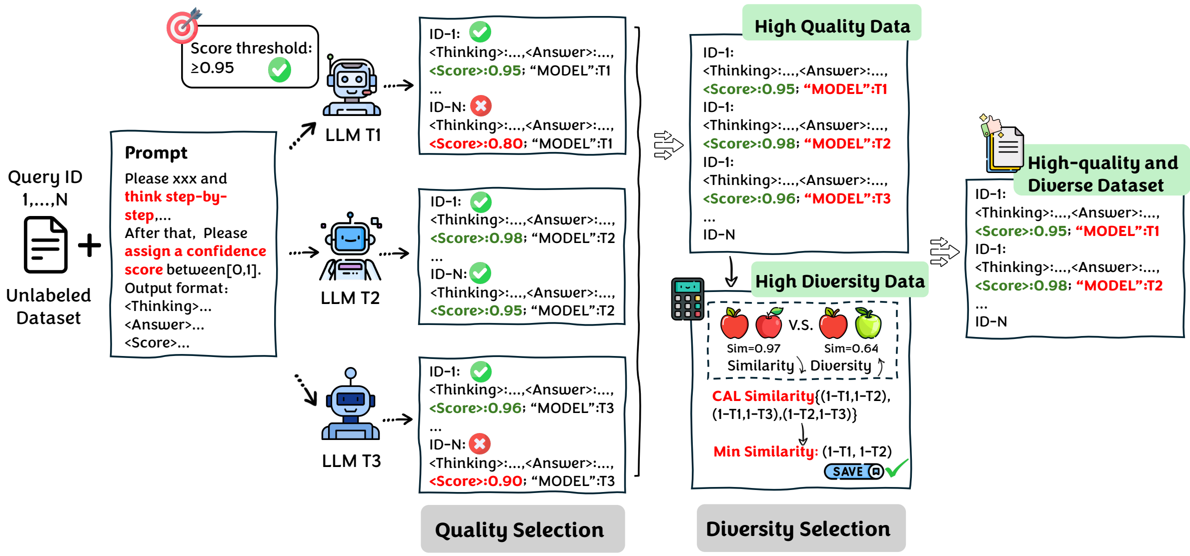

Figure 3: **Dual-Criteria Rejection Sampling Architecture.** Our proposed DCRS method comprises two stages: Quality Selection and Diversity Selection. The first stage filters samples using confidence scores, while the second stage enhances diversity by selecting samples based on similarity. This approach ensures a high-quality and diverse distillation dataset, enabling our method to effectively adapt to unsupervised scenarios.

(DCRS), which extends traditional rejection sampling(Gilks and Wild, 1992) by integrating two key selection metrics: confidence scores and similarity measures. Leveraging a multi-teacher strategy, we first prompt teacher LLMs to generate an initial pool of pseudo-labels. DCRS performs sample selection in two sequential steps: quality selection and diversity selection.

3.1.1 Quality Selection

As shown in Figure 3, we take use of CoT prompting(Wei et al., 2022) to generate and extract reasoning patterns from multi-teacher LLMs. Given an unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$, where each x_i is a query, we first design a prompt template p to clarify the task solution method. The prompt instructs the LLM to produce the output O_i in the form of a triplet (r_i, y_i, c_i), where y_i is the predicted label for task x_i, r_i is the rationale provided by the LLM, and c_i is the confidence score for the reasoning and predicted label, represented as a decimal in the range $[0, 1]$. To ensure that the confidence score is more reliable, we compute it using a weighted combination of multiple performance factors:

$$c_i = \sum_{j=1}^{n} w_j \cdot m_j \tag{1}$$

where w_j denotes the weight assigned to the j-th factor and m_j represents the corresponding factor value. For example, in a code generation task, we let the large model assign a weighted score based on factors such as the reasoning process, code readability, and robustness, yielding the final confidence score. For k teacher models, we can obtain a set of outputs $O_i = \{O_{i1}, O_{i2}, \ldots, O_{ik}\}$ for each query x_i. Then, we set a confidence score threshold s. For each output O_i, if its confidence score $c_i \geq s$, the output sample is considered high-quality and retained; otherwise, it is discarded. Subsequently, diversity selection is performed on the retained sample set $\mathcal{H} = \{(x_i, r_i, y_i, c_i)\}_{i=1}^{P}$, where P is the total number of retained samples.

3.1.2 Diversity Selection

For the high-quality sample set $\mathcal{H}$, if the outputs $O_i = \{O_{i1}, O_{i2}, \ldots, O_{ik}\}$ for the same query x_i come from three or more different teacher models (i.e., $k \geq 3$), we calculate the semantic similarity between the rationales provided by these teacher models. Specifically, we assume that r_{ij} denotes the rationale generated by the j-th teacher for query x_i. Each rationale r_{ij} is then mapped into a fixed-dimensional embedding $E(r_{ij})$ using a pre-trained sentence embedding model. The cosine similarity between any two rationales r_{ip} and r_{iq} is computed as:

$$\text{sim}(r_{ip}, r_{iq}) = \frac{E(r_{ip}) \cdot E(r_{iq})}{\|E(r_{ip})\| \, \|E(r_{iq})\|} \tag{2}$$

where r_{ip} and r_{iq} are rationales provided by two distinct teacher models for the same question, with $1 \leq p < q \leq k$. We calculate the cosine similarity for all unique pairs (r_{ip}, r_{iq}). To maximize diversity, we select the pair of rationales that yields the lowest similarity score. In our implementation, we directly select the pair of rationales that exhibits the minimum cosine similarity. For a given query x_i, we define:

$$(p^*, q^*) = \arg \min_{1 \le p < q \le k} \mathrm{sim}\big(r_{ip}, r_{iq}\big)) \quad (3)$$

where (p^*, q^*) are the indices corresponding to the pair of rationales with the smallest similarity, thereby ensuring that the final distillation dataset $\mathcal{G} = \{(x_i, r_i, y_i)\}_{i=1}^{Q}$ is both high-quality and diverse, where P is the total number of final distillation samples.

3.2 Habitual Reasoning Distillation

In this section, we propose a novel distillation strategy to address the trade-off between inference-time computational cost and model performance. Our approach decomposes the distillation process into three sequential stages. In the first stage, the student model is distilled using data with full reasoning, allowing it to fully absorb the teacher's comprehensive thought process. During the second stage, the model is provided with data containing compressed reasoning, encouraging it to solve problems using succinct and minimal explanations. Finally, in the third stage, distilling is conducted on data without any explicit reasoning, enabling the student to learn an end-to-end mapping from query to answer. Notably, in the second stage, we integrate a Teacher-Guided Compression method that tailors the complexity of the reasoning information to the student model's capacity.

Stage-1: Full Reasoning Distillation. In this stage, the student model is trained to learn the complete reasoning paths under the supervision of the teacher models' full reasoning. The goal is to help the weak student model understand the logical steps involved in the task and build a solid foundation for further distillation.

The teachers' reasoning ability can be transferred by fine-tuning the student model using the full demonstration $\mathcal{G}$ derived from high-quality and diversity sample selection. More specifically, the process of learning full reasoning paths through fine-tuning is defined as follows:

$$\mathcal{L}_1 = \mathbb{E}_{\mathcal{G}}[\log \mathrm{P}_f([x; r; y])] \quad (4)$$

where f indicates the student model, $\mathbb{E}_{\mathcal{G}}$ is the expectation over the distillation dataset $\mathcal{G}$, and $\mathrm{P}_f([x; r; \hat{y}])$ is the probability assigned by the student model f to the joint input $[x; r; \hat{y}]$.

Stage-2: Reasoning-Compressed Distillation. In this stage, we progressively simplify the reasoning paths of the teacher model, generating more concise one by compressing the original reasoning paths. We observed that for the same problem, the outputs of student models are often shorter and feature more concise reasoning steps compared to those of teacher models. Therefore, we adopt a Teacher-Guided Compression approach that ensures the reasoning paths provided by the teacher models are better aligned with the learning characteristics of the student model, thereby enhancing overall distillation performance.

Specifically, for a given query x_i, the teacher model generates the original reasoning r_i, while the student model produces the reasoning r_i^{s1}. We design a prompt $p\prime$ to guide the teacher model in refining its original reasoning r_i based on the characteristics of the student model's output (e.g., output length, complexity of understanding the problem). This process can be represented as $(p\prime, r_i, r_i^{s1}) \to r_i^T$. Subsequently, we replace the original reasoning r_i in the dataset $\mathcal{G} = \{(x_i, r_i, y_i)\}_{i=1}^{N}$ with the refined reasoning r_i^T, resulting in the second-stage distillation dataset $\mathcal{G}' = \{(x_i, r_i^T, y_i)\}_{i=1}^{N}$. The second stage fine-tuning can be defined as:

$$\mathcal{L}_2 = \mathbb{E}_{\mathcal{G}'}[\log \mathrm{P}_f([x; r^T; y])] \quad (5)$$

Stage-3: Reasoning-Free Distillation. Finally, we completely remove the reasoning paths and only retain the final answer as the supervision signal. The student model is trained to directly output the correct answer without relying on any reasoning chain. The goal of this stage is to enable the student model to form a "habitual" ability, allowing it to efficiently complete tasks without the need for complex reasoning. After removing r, the new dataset $\mathcal{G}'' = \{(x_i, y_i)\}_{i=1}^{N}$ only contains the original query x_i and the label y_i predicted by the teacher model. The third stage fine-tuning can be defined as:

$$\mathcal{L}_3 = \mathbb{E}_{\mathcal{G}''}[\log \mathrm{P}_f([x; y])] \quad (6)$$

In summary, Stage 1 and Stage 2 use full resoning distillation and reasoning-compressed distillation, respectively, to gradually familiarize the student model with the complete reasoning chain and establish a systematic reasoning pattern. This enables the model to master the key problem-solving

Method		MBPP		CQA		MetaMath	
		Pass@1	Token	Accuracy	Token	Accuracy	Token
GPT-4		77.68%	493	83.27%	312	86.31%	512
GPT-4o-mini		77.14%	451	81.76%	301	87.03%	529
Mistral-large		73.83%	365	80.15%	288	86.33%	463
Mistral-7B-v0.3	Vanilla Student	42.90%	209	60.19%	171	30.12%	255
	Standard KD	52.67%	53	63.63%	**5**	38.19%	**7**
	Distilling	53.39%	170	64.47%	129	42.20%	397
	Tinyllm	54.45%	175	67.39%	225	41.67%	226
	TwT	**57.11%** (↑ 4.88%)	**48**	**76.16%** (↑ 13.01%)	6	**47.94%** (↑ 13.60%)	**7**
Phi-3.5mini	Vanilla Student	54.71%	199	64.15%	150	73.36%	279
	Standard KD	61.71%	**98**	64.48%	**6**	78.16%	**10**
	Distilling	62.03%	202	65.74%	148	78.19%	370
	Tinyllm	64.44%	192	70.30%	144	78.23%	240
	TwT	**67.93%** (↑ 5.42%)	105	**76.42%** (↑ 8.70%)	10	**83.58%** (↑ 6.84%)	15

Table 1: **Quantitative results for baseline models.** The top three rows show the inference results of our teacher models, while "Distilling" is an abbreviation for "Distilling Step-by-Step." The best and the second best results are highlighted in bold and underlined respectively. The improvements of TwT over the second best results are shown in green with an upward arrow.

steps at both conceptual and operational levels, ultimately internalizing complex reasoning abilities and forming a stable reasoning mechanism. In Stage 3, we perform end-to-end training to further deepen the model's understanding of the relationship between problems and their answers. Since the explicit reasoning chain has been embedded as an inherent capability in the previous stages, Stage 3 no longer relies on explicit reasoning paths but instead directly reinforces answer generation. This progressive strategy allows the student model to output correct answers without explicit reasoning, thereby reducing inference-time computational cost while enhancing overall performance.

4 Experiment

4.1 Experiment Setup

Datasets. We evaluate our TwT on 3 benchmark datasets for 3 different NLP tasks: MBPP(Austin et al., 2021) for NL to python code generalization; CommonsenseQA (CQA)(Talmor et al., 2018) for commonsense question answering; MetaMathQA (MetaMath)(Yu et al., 2023) for mathematical reasoning, which is augmented from the training sets of GSM8K(Cobbe et al., 2021) and MATH(Hendrycks et al., 2021).

Models. We utilize GPT-4, GPT-4omini and Mistral-Large as our teacher models, which are accessed through OpenAI's API and MistralAI's API. For the student models, we use Mistral-7B-v0.3 and Phi-3.5mini. For the pre-trained sentence embedding model, we leverage all-mpnet-base-v2.

Baselines. For our baselines, we evaluate three types of methods: teacher model's performance, vanilla student model's performance, and knowledge distillation based methods that containing Standard-KD (Hinton, 2015), a general distillation method that fine-tunes the student model using the teacher model's generated labels as ground-truth; Distilling-Step-by-Step (Hsieh et al., 2023), which leverages LLM-generated rationales as additional supervision to train smaller models; TinyLLM (Tian et al., 2024), a paradigm that distills diverse reasoning paths from multiple teacher LLMs into a student model.

Implementation Details. In our experimental setup, we employed training with LoRA fine-tuning, setting the LoRA rank to 8, a learning rate of 1e-5, a batch size of 8, 4 training epochs, and a context window of 4096 tokens. During inference, we used a temperature of 0, max tokens set to 2048, and a top-p value of 0.95. For the sampling process, we selected a scoring threshold of $s = 0.95$. All experiments were conducted on four NVIDIA A100 Tensor Core GPUs, enabling large-scale training and efficient computation. The prompts for specific methods and a case study are provided in the appendix.

4.2 Baseline Comparison

Across three specific-tasks, TwT consistently outperforms other distillation methods as shown in Tab. 1. Compared with the best performing baseline, TwT achieves an improvement of up to 13.60% compared to the best-performing baseline, while reducing the token number by 98.2% (token numbers from 397 to 7 on MetaMath dataset), substantially

Method	MBPP				CQA				MetaMath			
	Mistral-7B-v0.3		Phi-3.5mini		Mistral-7B-v0.3		Phi-3.5mini		Mistral-7B-v0.3		Phi-3.5mini	
	Pass@1	Token	Pass@1	Token	Accuracy	Token	Accuracy	Token	Accuracy	Token	Accuracy	Token
TwT-stage1	54.83%	291	65.32%	310	73.31%	141	72.99%	138	46.48%	295	80.05%	313
TwT-stage2	56.48%	154	66.42%	184	74.89%	84	75.39%	73	47.02%	169	81.33%	196
TwT-stage3	**57.11%**	**48**	**67.93%**	**105**	**76.16%**	**8**	**76.42%**	**10**	**47.94%**	**12**	**83.58%**	**15**

Table 2: **Quantitative results for three distillation stages.** Accuracy and the number of output tokens were used to evaluate the model performance. The best results were highlighted in bold.

Method	MBPP				CQA				MetaMath			
	Mistral-7B-v0.3		Phi-3.5mini		Mistral-7B-v0.3		Phi-3.5mini		Mistral-7B-v0.3		Phi-3.5mini	
	Pass@1	Token	Pass@1	Token	Accuracy	Token	Accuracy	Token	Accuracy	Token	Accuracy	Token
TwT-stage1	54.83%	291	65.32%	310	73.31%	141	72.99%	138	46.48%	295	80.05%	313
TwT-stage2	56.48%	154	66.42%	184	74.89%	84	75.39%	73	47.02%	169	81.33%	196
TwT-stage3	56.94%	133	67.18%	161	75.44%	55	75.85%	64	47.41%	144	81.99%	172
TwT-stage4	**57.49%**	**42**	**68.21%**	**100**	**76.29%**	**8**	**76.88%**	**9**	**48.37%**	**12**	**83.61%**	**13**

Table 3: **Quantitative results for four distillation stages.** Accuracy and the number of output tokens were used to evaluate the model performance. The best results were highlighted in bold.

Samping Method	MBPP	CQA	MetaMath
Log probability-Based	74.83%	75.29%	72.19%
Hard Rejection Sampling-Based	72.29%	73.11%	70.26%
Confidence Score-Based	**83.49%**	**84.90%**	**81.14%**

Table 4: **Quantitative results for sampling methods.** Accuracy was used to evaluate the model performance. The best results were highlighted in bold.

lowering the inference cost. Typically, reducing inference tokens leads to a decline in performance; however, TwT overcomes this trade off by maintaining or even enhancing model performance while dramatically reducing token usage, thus achieving both high performance and low inference-time computational cost simultaneously. In addition, TwT effectively bridges the gap between the student and teacher models, significantly narrowing the performance disparity observed in vanilla student models.

4.3 Distillation Stage Analysis

In the Habitual Reasoning Distillation phase, we separately evaluated the student model's performance at each fine-tuning stage and tracked the number of output tokens during inference, as shown in Table 2. The results indicate that TwT steadily improves with each stage, while the inference token numbers gradually decrease. By leveraging our distillation strategy, the model successfully internalizes the reasoning process as part of its own capabilities.

Furthermore, we refined our three-stage procedure by extending it to four stages, introducing an additional step after the second stage in which the teacher model further compresses the reasoning process based on the student's output. As shown in Table 3, this refined approach yields a slight improvement but does not significantly outperform the three-stage method. Consequently, our three-stage process already achieves the intended effectiveness.

4.4 Sampling and Compression Analysis

We evaluated the effectiveness of our DCRS method's confidence score selection for distillation data and our Teacher-Guided Compression approach for enhancing student model's performance. For sampling, we compared DCRS with two alternatives—selection based on log probability and hard rejection sampling using Good, Medium, Poor categories from LLM outputs—across three datasets. As shown in Table 4, DCRS consistently outperforms the alternatives with nearly a 10% gain in accuracy, indicating that multi-dimensional confidence scoring more effectively filters high-quality examples. For compression, we compared our Teacher-Guided Compression with fixed-length compression and a compressor-based approach. As detailed in Table 5, our method better aligns the teacher's outputs with the student model's limited capacity, enhancing both training efficiency and overall performance.

Compression Method	MBPP		CQA		MetaMath	
	Mistral-7B-v0.3	Phi-3.5mini	Mistral-7B-v0.3	Phi-3.5mini	Mistral-7B-v0.3	Phi-3.5mini
Fixed-Length Compression	55.89%	64.94%	73.29%	72.92%	43.18%	80.62%
Compressor	56.01%	65.10%	73.61%	73.69%	45.22%	80.95%
Teacher-guided Compression	**56.48%**	**66.42%**	**74.89%**	**75.39%**	**47.02%**	**81.33%**

Table 5: **Quantitative Results for Compression Methods.** Accuracy was used to evaluate the model performance. The best results were highlighted in bold.

Methods	MBPP		CQA		MetaMath	
	Accuracy	Token	Accuracy	Token	Accuracy	Token
w/o Multi-Teacher Strategy (GPT-4)	55.38%	54	74.64%	7	46.82%	15
w/o DCRS	55.42%	48	74.79%	9	47.11%	14
w/o Compression Distillation Stage	54.49%	93	72.19%	21	46.38%	48
TwT	**57.11%**	**48**	**76.16%**	**8**	**47.94%**	**12**

Table 6: **Ablation Study for Model Components.** Accuracy and the number of output tokens were used to evaluate the model performance. The best results were highlighted in bold.

4.5 Ablation Study

We further analyze the impact of each component on TwT's performance through ablation studies. Specifically, w/o Multi-Teacher Strategy evaluates the effect of using a single teacher model, w/o DCRS assesses performance without filtering the distillation data, and w/o Compression Distillation Stage examines the impact of directly removing the reasoning step as shown in Tab. 6.

When comparing TwT with the multi-teacher strategy against the single-teacher approach, we observe an average improvement of 1.4% in the distillation performance. This result highlights the effectiveness of the multi-teacher strategy, as it offers a more diverse set of reasoning paths for the student model, thereby facilitating its learning process. Comparing TwT's DCRS strategy to directly organizing the raw pseudo-labeled data genereated by teachers, TwT achieves an average improvement of 1.5%, underscoring the importance of performing high-quality and diverse data sampling before distillation. This step helps avoid the interference of low-quality data and boosts overall distillation performance. Finally, when comparing TwT's multi-stage distillation approach with the strategy of simply removing the reasoning process (i.e., without any intermediate compression), TwT demonstrates an approximate 3.5% accuracy gain, while the uncompressed approach not only underperforms in accuracy but also increases the number of output tokens. This finding confirms the necessity of gradually internalizing the reasoning capability for optimal performance.

5 Conclusion and Future Work

We introduced TwT, a novel distillation framework that internalizes reasoning abilities into a student model under multi-teacher guidance. It incorporates a Dual-Criteria Rejection Sampling stage to obtain high-quality, diverse distillation datasets and a staged distillation phase to gradually integrate reasoning capabilities into the student model. TwT achieves high performance with low inference cost without relying on labeled data or an explicit reasoning process. In future work, we will continue to explore whether subdividing the distillation stages can bring further enhancements to our framework.

Limitations

Although our method has achieved excellent results, there are still some minor flaws here. One limitation of our approach is that it currently only works effectively on specific tasks and is not applicable to datasets containing mixed tasks. Additionally, the Dual-Criteria Rejection Sampling process could consist of noise. The impact of this potential noise on performance is still undetermined. A potential future direction is to investigate implicit natural language reasoning by utilizing more advanced training strategies. While current tasks are primarily focused on explicit reasoning, incorporating implicit reasoning mechanisms could improve the model's robustness and its ability to generalize across different tasks.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. 2022. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10925–10934.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lang Cao. 2024. GraphReason: Enhancing reasoning capabilities of large language models through a graph-based verification approach. In *Proceedings of the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ACL 2024)*, pages 1–12, Bangkok, Thailand. Association for Computational Linguistics.

Hongzhan Chen, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. 2023. Mcc-kd: Multi-cot consistent knowledge distillation. *arXiv preprint arXiv:2310.14747*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Walter R Gilks and Pascal Wild. 1992. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348.

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

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv: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. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. 2023. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models. *arXiv preprint arXiv:2311.09214*.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2024. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614.

Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. 2024a. Reasoning in token economies: Budget-aware evaluation of llm reasoning strategies. *arXiv preprint arXiv:2406.06461*.

Libo Wang. 2024. Reducing reasoning costs: The path of optimization for chain of thought via sparse attention mechanism. *arXiv preprint arXiv:2411.09111*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

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

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Yuzhe Zhang, Huan Liu, Yang Xiao, Mohammed Amoon, Dalin Zhang, Di Wang, Shusen Yang, and Chai Quek. 2024. Llm-enhanced multi-teacher knowledge distillation for modality-incomplete emotion recognition in daily healthcare. *IEEE Journal of Biomedical and Health Informatics*, pages 1–11.

Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P Xing, et al. 2022. Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 559–578.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Prompt

A.1 Prompt for Teachers to Generate Pseudo-Labels on the MBPP Dataset

Prompt for Teachers to Generate Pseudo-Labels on the MBPP Dataset

Implement a Python function based on the following guidelines. Ensure that it passes all three provided test cases which use assert statements for validation.\n
Instructions:\n

1. Function Name: Use the exact function name provided in the test cases.\n
2. Input and Output: The function should accept the same number and types of input arguments and return the same type of output as specified in the test cases.\n
3. Function Behavior: The function should pass all three provided test cases when executed.\n
4. Allowed Libraries: You may use any standard Python libraries.\n
5. **Confidence Score:** After completing the code, please assign a confidence score between 0.00 and 1.00 to your code based on the following criteria:
   (1)Correctness (50% weight):Whether the functionality is implemented correctly and all test cases are passed.\n
   (2)Readability (20% weight): Whether the code structure is clear, variable and function names are meaningful.\n (3)Execution Efficiency (20% weight): Whether the algorithm is efficient and there is unnecessary redundant code.\n
   (4)Test Coverage (10% weight): Whether possible edge cases and exceptions are considered.\n Please assign a score for each category based on the weight, then calculate the weighted total score. The final score should be in the format score: 0.85. Please provide a clear numerical value without any additional explanation.
6. **Response Format:** Your response must include three parts:
   (1)Thinking: A detailed step-by-step explanation of your approach.
   (2)Code: The Python code implementing the function, without additional explanations.
   (3)Score: Your confidence score as specified. \n Response Format Example:```Thinking:(Your detailed explanation here.)```\n ```python(Your Python code goes here)```\n```score: 0.85 （do not give any explanation） ```

Figure 4: Prompt for Teachers to Generate Pseudo-Labels on the MBPP Dataset.

A.2 Prompt for Teachers to Generate Pseudo-Labels on the MetaMath Dataset

Prompt for Teachers to Generate Pseudo-Labels on the MetaMath Dataset
You are a helpful assistant. Your task is to answer a math question. Please think step by step to analyze the question carefully and give your clear thinking step to solve the problem.\n Instructions:\n **1. Confidence Score:** After solving the problem, please assign a confidence score between 0.00 and 1.00 to your answer based on the following: (1)Correctness (50% weight): the answer is factually correct based on the information or reasoning provided.\n (2)Logical Reasoning (25% weight): Whether the analysis exhibits a clear, reasonable thought process.\n (3)Clarity (15% weight): The reasoning should be easy to follow, avoiding unnecessary complexity. \n (4)Completeness (10% weight): The analysis should cover all necessary aspects of the question. Please assign a score for each category based on the weight, then calculate the weighted total score. The final score should be in the format score: 0.85. **2. Response Format:** Your response must include three parts: **(1)Thinking**: A detailed step-by-step explanation of your approach. **(2)Answer**: The number for the problem, without additional explanations. **(3)Score**: Your confidence score as specified. \n Response Format Example:```Thinking:[your thinking process] ```; ```The answer is: [an exact number]```; ```score: [an exact number, for example:0.85]```

Figure 5: Prompt for Teachers to Generate Pseudo-Labels on the MetaMath Dataset.

A.3 Prompt for Teachers to Generate Pseudo-Labels on the CQA Dataset

Prompt for Teachers to Generate Pseudo-Labels on the CQA Dataset

You are a helpful assistant. Your task is to answer questions based on provided options. Each input consists of a question and five answer choices.\n
Instructions:\n

1. **Confidence Score:** After giving the answer, please assign a confidence score between 0.00 and 1.00 to your answer based on the following:
   (1)Correctness (50%): The answer is factually correct.\n
   (2)Logical Reasoning (25%): The explanation demonstrates clear and reasonable reasoning.\n
   (3)Clarity (15%): The response is easy to follow. \n
   (4)Completeness (10%): All necessary aspects of the question are addressed.
2. **Response Format:** Your response must include three parts:
   (1)Thinking: A detailed step-by-step explanation of your approach.
   (2)Answer: The answer choice without additional explanations.
   (3)Score: Your confidence score as specified. \n Response Format Example:```Thinking:[your thinking process] ```; ```The answer is: [answer choice]```; ```score: [an exact number, for example:0.85]```

Figure 6: Prompt for Teachers to Generate Pseudo-Labels on the CQA Dataset.

B Case Study

B.1 Query

Q: A bad person places little value on being honest, acting without pretense or being what? Choices are: a: excellent, b: upright, c: premium, d: competent, e: sincere.

B.2 Teacher's Original Reasoning (Stage-1's dataset)

Reasoning Path: The question describes a person who lacks honesty and genuineness. The phrase "being what" suggests we need a quality that embodies authenticity. Among the options, "sincere" (option e) best fits this requirement, as it represents being genuine and truthful—qualities that the described person would likely disregard. The other options do not directly capture this concept. Therefore, the answer is e. (437 words)

B.3 Stage2 Student's inference

Reasoning Path: The question asks for an antonym for a good or virtuous quality. In this case, the opposite of being honest, acting without pretense or being sincere would be insincere, dishonest, or deceitful. Among the given options, the most suitable antonym is "sincere". The answer is e. (279 words)

B.4 Teacher's refinement prompt

B.5 Refined Reasoning (Stage-2's dataset)

Reasoning Path: The question asks for an antonym of a good or virtuous quality. A bad person would lack honesty or authenticity. The phrase "being what" points to a quality opposite to these virtues. Among the options, "sincere" fits best as it contrasts with honesty and acting without pretense.The answer is e. (279 words)

B.6 Stage-3's dataset

Only contains predicted labels.

Prompt for Teacher's Refinement On CQA dataset

-Please compare your previous output on this task with the student's output in detail.\n
-After identifying any differences in style, tone, structure, and length, modify your original output to better align with the student's preferences. \n
-Ensure that your revised output maintains a similar length and depth of content as the student's version, while preserving all essential information.\n
-The goal is to harmonize your output with the student's approach, creating consistency in both presentation and quality.\n

Figure 7: Prompt for teacher's refinement.