# CAD: Curriculum-Enhanced Agent Distillation via Trajectory Refinement and Difficulty-Aware Scheduling

Xiang Yu[1], Yujia Huo[1,2*], Yongbin Qin[1], Fujian Feng[1], Gan Liu[1], Dawen Xia[3]
[1]School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China
[2]Guizhou Provincial Key Laboratory of Applied Mathematics and Computing Power & Algorithms, Guizhou University, Guiyang, China
[3]College of Microelectronics and Artificial Intelligence, College of Big Data Engineering, Kaili University, Kaili, China
{xiangyu, huo.yujia, ybqin, fujian_feng, liugan}@gzmu.edu.cn, dawenxia@kluniv.edu.cn
*Corresponding author.

***Abstract*—Large Language Model (LLM) agents exhibit robust problem-solving capabilities through tool usage; however,their deployment is constrained by prohibitive computational costs. Knowledge distillation offers a compression pathway; however, transferring dynamic agent behaviors remains challenging due to the complexity of interaction trajectories and the risk of training instability. We present Curriculum-enhanced Agent Distillation (CAD), a framework designed to progressively transfer agent capabilities to lightweight student models. Unlike static chain-of-thought distillation, CAD orchestrates a dynamic, easy-to-hard learning curriculum governed by a novel multi-dimensional metric quantifying tool frequency, code complexity, and interaction depth. Furthermore, we introduce trajectory optimization techniques, including a "First-thought Prefix" mechanism, to refine teacher supervision. Extensive evaluations across eight benchmarks (e.g., mathematical reasoning, factual QA) demonstrate that CAD-trained students (1.5B) significantly outperform standard distillation baselines and achieve parity with 3B-scale models, exhibiting superior generalization in open-ended tasks.**

***Index Terms*—Large Language Models, Agent Distillation, Curriculum Learning, Chain-of-Thought.**

## I. Introduction

The emergence of the Think–Act–Observe paradigm has substantially expanded the functional scope of Large Language Models (LLMs), enabling them to operate as autonomous agents that interact with external environments through tool invocation and iterative feedback [1]–[4]. While such agentic LLMs demonstrate strong capabilities in reasoning, retrieval, and planning, their practical deployment remains severely constrained by prohibitive computational costs and large parameter sizes. Consequently, effectively transferring agent capabilities to lightweight models has become a critical challenge for enabling scalable and on-device intelligence [5]–[7].

Knowledge distillation offers a promising pathway for model compression; however, its application to agent-based scenarios is far from straightforward. Unlike static reasoning tasks, agent learning involves long-horizon Thought–Action–Observation trajectories that are highly dynamic and structurally complex. On the one hand, raw trajectories generated by teacher agents often contain redundant steps, trial-and-error behaviors, or locally suboptimal decisions, which can mislead small student models and cause training instability. On the other hand, agent tasks vary drastically in intrinsic difficulty across multiple dimensions, such as tool usage frequency, reasoning depth, and interaction length. Naively mixing trajectories of heterogeneous difficulty levels during training frequently leads to imbalanced learning, where student models overfit simple behaviors while failing to acquire complex decision-making skills [8], [9]. Motivated by these challenges, we argue that student agents should not be forced to learn full-complexity behaviors in a single stage. Instead, they should follow a progressive learning process—mastering basic tool usage and simple reasoning patterns before gradually advancing toward complex, multi-step decision-making. This perspective naturally motivates the integration of curriculum learning into the agent distillation framework, providing structured guidance that aligns with the learning capacity of lightweight models.

To address these issues, we introduce Curriculum-enhanced Agent Distillation (CAD). Our core insight is that a student model should master fundamental tool usage and basic logic before attempting complex, multi-step trajectories. Specifically, we propose:

1) **A Multi-dimensional Curriculum:** We quantify trajectory difficulty through three orthogonal metrics: TCF, CEC, and EIF. This allows CAD to schedule a training sequence that aligns with human cognitive patterns.
2) **Trajectory Optimization:** To ensure high-fidelity supervision, we introduce the "First-thought Prefix" to guide the teacher's reasoning and utilize MCAG to filter out noisy or erroneous execution paths.

This work was supported by the National Natural Science Foundation of China (Grant No. 62266013), Guizhou Provincial Basic Research Program (Grant No. MS[2026]276), Guizhou Provincial Key Laboratory of Applied Mathematics and Computing Power & Algorithms (Qiankehe Platform ZSYS[2025]039), Guizhou Provincial Science and Technology Major Project (Qiankehe Major[2025]034)

3) **Empirical Validation:** Our results indicate that CAD not only accelerates the learning process but also yields a student model with superior generalization capabilities across unseen tasks.

This section will detail the design principles, algorithmic framework, and implementation details of the CAD method. ly, through extensive experiments on eight benchmark datasets covering factual question-answering and mathematical reasoning, the effectiveness of the CAD method is systematically verified and compared in depth against various baseline methods. Experimental results indicate that the CAD method not only significantly improves the performance of small-scale agent models—enabling them to match the levels of CoT distillation models with 2–4 times the parameter count on certain tasks—but also demonstrates significant advantages in generalization capabilities for complex tasks, providing a new, feasible pathway for constructing practical and efficient lightweight language agents.

## II. Related Work

In recent years, Large Language Models (LLMs) have achieved breakthrough progress in tasks such as natural language understanding, reasoning, and generation [10]–[13]. However, their massive parameter scale and high inference costs severely constrain practical deployment in resource-limited scenarios. Focusing on model compression, inference capability migration, and the enhancement of complex task execution, academia has produced a substantial body of research. This study is primarily associated with four key directions: Knowledge Distillation methods, Chain-of-Thought Distillation, Agent Learning and Distillation, and the application of Curriculum Learning in complex model training.

### *A. Knowledge Distillation and Large Model Compression*

Knowledge Distillation (KD), first proposed by Hinton et al., is centered on the concept of transferring the "dark knowledge" inherent in a large model to a smaller model by minimizing the divergence between the output distributions of the student and teacher models [14]. Early work primarily focused on classification tasks, achieving performance retention through soft targets or intermediate feature alignment. With the development of pre-trained language models, distillation research has gradually expanded to the field of Natural Language Processing (NLP); methods such as DistilBERT and TinyBERT achieve effective compression of BERT-like models by distilling hidden layer representations, attention matrices, or logits [15], [16].

In the context of Large Language Models, researchers have begun to address the distillation of instruction-tuned models. One category of methods enables student models to maintain robust instruction-following capabilities with significantly reduced parameters by directly distilling the teacher model's final output on instruction data [17]–[19]. However, such approaches typically prioritize "result-level" imitation, neglecting intermediate behavioral information during complex reasoning or decision-making processes, thereby limiting generalization capabilities on high-difficulty tasks.

### *B. Chain-of-Thought (CoT) Distillation*

To enhance model reasoning capabilities, Chain-of-Thought (CoT) prompting was proposed to explicitly guide models in generating intermediate reasoning steps. Subsequently, multiple studies have attempted to integrate CoT as a supervision signal within the knowledge distillation framework—specifically, by distilling the reasoning text generated by the teacher model to help the student model learn step-by-step reasoning patterns. Experiments demonstrate that CoT distillation significantly outperforms methods that only distill the final answer in tasks such as mathematical and symbolic reasoning.

Although CoT distillation alleviates the issue of "small models being unable to reason" to a certain extent, it remains essentially a form of static text imitation: the student model learns the reasoning trajectory already completed by the teacher, rather than how to make dynamic decisions in a real-world environment [20], [21]. When tasks require invoking external tools, performing information retrieval, or executing code, relying solely on CoT text can lead to hallucinations or reasoning distortions. Furthermore, in multi-task scenarios, the length and complexity of CoT vary significantly across tasks; direct mixed distillation can easily lead to overfitting on simple tasks while resulting in underfitting on complex ones.

### *C. Agent Learning and Agent Distillation*

Recently, LLM-based Agents have emerged as a research hotspot. These methods typically adopt a "Thought–Action–Observation" interaction paradigm, enabling the model to call external tools (e.g., search engines, code executors) to complete complex tasks [22]–[24]. Representative works include ReAct, Toolformer, and various reasoning frameworks based on tool invocation. These methods have significantly improved model performance in tasks such as open-domain QA, multi-hop reasoning, and mathematical calculation.

Building on this, some studies have begun to explore Agent Distillation, which involves transferring the complete interaction trajectory of a teacher agent to a student model [25]. Compared to CoT distillation, agent distillation covers not only reasoning text but also tool-calling decisions and environmental feedback processing, thus theoretically possessing greater generalization potential. However, existing agent distillation methods generally face two prevalent issues: first, the trajectories generated by teacher agents are often lengthy and contain trial-and-error behaviors, making them difficult for small models to learn directly; second, in multi-task scenarios, different tasks impose vastly different requirements on tool usage and decision depth. The lack of effective training scheduling mechanisms often leads to training instability.

### *D. Curriculum Learning in Complex Model Training*

Inspired by human learning processes, Curriculum Learning advocates organizing training samples in an order ranging

from easy to difficult [26]–[28]. This concept has been widely applied in fields such as computer vision, speech recognition, and reinforcement learning. In recent years, studies have also attempted to introduce curriculum learning into the training and fine-tuning processes of language models, for instance, by dynamically adjusting the training order based on sample length, perplexity, or loss values to improve model convergence speed and stability. In distillation scenarios, some works have explored "shallow-to-deep" strategies, such as distilling simple samples first before gradually introducing complex ones [29]–[32]. However, these methods mostly rely on single difficulty metrics and primarily target static task outputs. Regarding agent distillation—a highly structured and intensely interactive learning process—systematic research on how to construct a reasonable task difficulty metric system and deeply integrate curriculum learning with trajectory distillation remains lacking.

In summary, while existing research has made significant progress in knowledge distillation, chain-of-thought reasoning, and agent learning, distinct shortcomings persist in the scenario of multi-task agent distillation: static CoT distillation struggles to cover dynamic decision-making processes, direct agent distillation imposes an excessive learning burden on small models, and existing curriculum learning methods have not fully accounted for the multi-dimensional complexity of agent tasks.

Addressing these issues, this paper proposes a Curriculum-enhanced Agent Distillation (CAD) method. By leveraging multi-dimensional task difficulty quantification, dynamic curriculum scheduling, and teacher trajectory optimization, this method effectively reduces the difficulty for student models to learn complex agent behaviors and achieves stable, efficient capability transfer in multi-task scenarios. In terms of design motivation and technical pathway, this method forms a clear distinction from existing work and demonstrates significant advantages in experiments, offering a new solution for constructing lightweight language agents.

## III. Curriculum-enhanced Agent Distillation

### A. Method Overview

The Curriculum-enhanced Agent Distillation (CAD) method proposed in this study aims to integrate a curriculum learning mechanism into the agent distillation framework, thereby addressing the difficulty student models face in learning complex agent behaviors. The overall framework is conceptually illustrated in Fig 1. It comprises four core modules: (1) **Teacher Agent Trajectory Extraction and Optimization**, responsible for generating high-quality instructional data; (2) **Task Difficulty Quantification**, which provides the quantitative basis for curriculum design; (3) **Dynamic Curriculum Scheduling**, which plans learning paths ranging from easy to difficult; and (4) **Curriculum-based Agent Distillation**, which executes the final knowledge transfer.

Specifically, the method employs Large Language Models (LLMs) with robust tool-use capabilities (e.g., Qwen2.5-32B-Instruct) as teacher models to generate and optimize "Think–Act–Observe" trajectories across various tasks. Subsequently, tasks are quantified and scored via an innovative three-dimensional difficulty metric to design dynamic curricula. Under the guidance of these curricula, student models (e.g., Qwen2.5-1.5B/3B) progressively transition from mastering basic single-tool usage to acquiring complex multi-tool collaboration and multi-step decision-making capabilities.

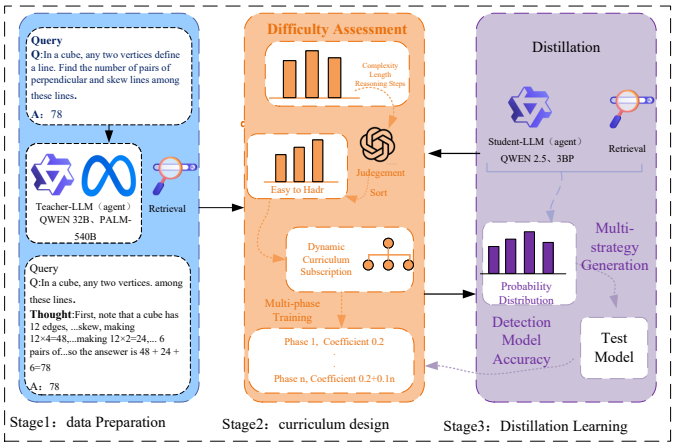



Fig. 1. The framework of Curriculum-enhanced Agent Distillation (CAD). It illustrates the process of generating optimized teacher trajectories, quantifying task difficulty, and training the student model through a progressive curriculum.

### B. Agent Trajectory Extraction and Optimization

High-quality teacher trajectories serve as the foundation for successful agent distillation. However, trajectories generated by standard LLM agents often contain redundancy, inefficiency, or errors. To address this, we design a two-stage optimization strategy comprising *trajectory compression* and *quality enhancement*.

To reduce the learning burden on the student model, we perform structured compression on the raw trajectories generated by the teacher, removing redundant information while retaining core logic. The specific steps include:

1) **Redundant Step Removal:** Identifying and eliminating repetitive tool calls (e.g., repeated searches for the same query) and invalid code debugging processes.
2) **Tool Template Encapsulation:** Abstracting common code execution logic into parameterized templates. For instance, encapsulating slope calculation logic '(y2-y1)/(x2-x1)' with varying numerical values into a 'calculate_slope(p1, p2)' tool template. This allows the student model to focus on learning *when* and *how* to invoke tools rather than memorizing specific code implementations.

Through this compression technique, the average trajectory length is reduced by 30%–40% while retaining over 90% of the core reasoning information. To further enhance trajectory quality and the student model's test-time performance, we introduce two effective techniques:

*1) First-thought Prefix (FTP):* Existing research indicates that while instruction tuning improves LLM adaptability, the

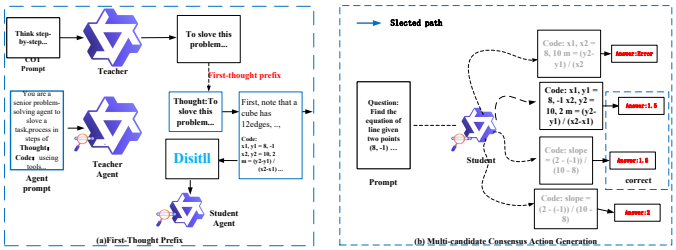



Fig. 2. (A) First-thought Prefix: We prompt the teacher with CoT for step-by-step reasoning. Using the first reasoning step as a prefix, we generate agent trajectories and distill them into the student agent for CoT-style reasoning initialization. (B) Multi-candidate Consensus Action Generation: The agent generates multiple candidate actions and selects the one with consistent results.

initial thinking phase in complex agent tasks often lacks the hierarchical structure and progressive logic inherent in Chain-of-Thought (CoT) reasoning. To mitigate this, we propose the First-thought Prefix mechanism.

First, we constrain the teacher model to the CoT reasoning paradigm to perform pre-reasoning on the input query, locking the first reasoning step as the "First-thought Prefix" with structured guidance attributes. Second, when the teacher agent initiates task reasoning, this prefix is injected into the model input as a pre-constraint. This mechanism forces the teacher agent to unfold its reasoning process following normative logic, generating high-quality trajectories with superior planning and completeness. When used as supervision signals, these trajectories significantly improve the student model's learning efficiency. This process can be formalized as:

$$\begin{aligned} y_1 &\sim P_T(\cdot|x, I_{CoT}), \\ \tau &= \{(r'_1, a_1, o_1), \ldots, (r'_{L_\tau}, a_{L_\tau}, o_{L_\tau})\} \\ &\sim P_T(\cdot|x, y_1, I_{agent}) \end{aligned} \tag{1}$$

where $x$ is the input, $I_{CoT}$ and $I_{agent}$ denote the prompts for CoT and agent modes respectively, $y_1$ is the first thought step, and $\tau$ represents the generated trajectory.

*2) Multi-candidate Consensus Action Generation (MCAG):* To address the issue of student models generating invalid or erroneous actions (e.g., non-standard code formats) during inference, we propose the MCAG strategy for the testing phase. (see Fig 2)

Instead of using a single decoding method, the student model employs high-temperature nucleus sampling to generate $N$ groups of complete "Think–Act" candidate sequences. We then introduce a lightweight code interpreter to perform an initial screening, filtering out sequences with parsing failures or runtime errors. Subsequently, we verify the consistency of the execution results among the valid sequences. Using a majority voting rule, the action with the highest result overlap is selected as the final output. This strategy effectively mitigates the risk of invalid actions, enhancing stability and robustness in real-world interactions.

### C. Task Difficulty Quantification and Curriculum Design

To implement effective curriculum learning, precise quantification of agent task difficulty is essential. We discard subjective standards in favor of a three-dimensional objective evaluation system:

1) **Tool Call Frequency (TCF):** Measures the task's dependence on tools, defined as the number of tool calls required to complete the task.
2) **Code Execution Complexity (CEC):** Comprehensively evaluates the length of required code, control flow complexity (e.g., loops, conditionals), and dependence on external libraries.
3) **Environment Interaction Frequency (EIF):** Measures the number of multi-step decision-making rounds and environmental feedback processing required.

Each dimension is normalized to the $[0, 1]$ interval, and the final sample difficulty score $D_{score}$ is calculated via weighted summation:

$$D_{score} = 0.3 \times \text{TCF} + 0.4 \times \text{CEC} + 0.3 \times \text{EIF} \tag{2}$$

The weights are empirically set to 0.3, 0.4, and 0.3, emphasizing the core role of code execution in complex tasks. Based on these scores, we design a three-stage dynamic curriculum:

**1. Basic Ability Phase (Phase 1):** In the early stage, only easy samples are used to enable the model to master the basic logic of single tools (e.g., simple retrieval, single-step calculation).

**2. Complex Collaboration Phase (Phase 2):** Medium-difficulty samples are introduced to train the model in multi-tool collaboration and multi-step reasoning. Hard samples are added dynamically; the proportion $\rho_p$ in the $p$-th stage is determined by:

$$\rho_p = \min\left(1, \sqrt{\frac{p}{P}(1-\lambda) + \lambda}\right) \tag{3}$$

where $\lambda$ is the curriculum difficulty ratio (set to 0.2 in this experiment), and $P$ is the total number of curriculum phases.

**3. Generalization Adaptation Phase (Phase 3):** High-difficulty samples and cross-scenario variant tasks are introduced, focusing on strengthening the model's generalization ability and robustness in complex, dynamic environments.

## IV. Experiments

### A. Datasets

To comprehensively evaluate the effectiveness of the proposed CAD method, we conducted experiments on eight public benchmark datasets covering two core challenges: factual reasoning and mathematical reasoning. Detailed statistics are provided in Table I.

For factual reasoning, we selected: HotpotQA, a multi-hop QA dataset requiring the correlation of information across multiple documents; Bamboogle, which tests the ability to locate key facts amidst interfering information through adversarial retrieval queries; MuSiQue, which evaluates stability in handling long contexts and deep logical reasoning; and 2WikiMultiHopQA, which emphasizes joint reasoning over textual information and knowledge graph relationships.

For mathematical reasoning, we used: MATH (in-domain), covering algebra and geometry from high school to competition levels; GSM-Hard, an extension requiring large-number arithmetic and rigorous planning; and AIME and OlymMath, representing Olympiad-level problems that test creative problem-solving and deep analysis capabilities in extreme reasoning scenarios.

These eight datasets constitute a multi-dimensional evaluation system, comprehensively validating the agent model's performance in tool usage, multi-hop reasoning, noise resistance, and cross-domain generalization.

TABLE I
OVERVIEW OF DATASETS USED IN EXPERIMENTS

| Task Type | Domain | Dataset | Description | Size |
|---|---|---|---|---|
| Factual Reasoning | In-domain | HotpotQA [33] | 2-hop question-answering | 400 |
| | Out-of-domain | Bamboogle [34] | Adversarial retrieval QA | 400 |
| | Out-of-domain | MuSiQue [35] | Complex connected reasoning | 400 |
| | Out-of-domain | 2WikiMultiHop [36] | Structured knowledge reasoning | 400 |
| Math Reasoning | In-domain | Math [37] | College-level math problems | 400 |
| | Out-of-domain | GSM-Hard [38] | Large number arithmetic | 400 |
| | Out-of-domain | AIME [39] | Olympiad-level problems | 90 |
| | Out-of-domain | OlymMath [40] | Advanced competition math | 200 |

### B. Experimental Configuration

**Models:** We employed Qwen2.5-32B-Instruct as the teacher model due to its robust instruction-following capabilities. The student models were selected from the Qwen2.5-Instruct series, covering three parameter scales: 1.5B, 3B, and 7B.

**Baselines:** We compared CAD against three primary baselines:

1) **CoT Distillation:** Transfers the teacher's static Chain-of-Thought reasoning.
2) **CoT Distillation + RAG:** Augments CoT distillation with retrieval capabilities.
3) **Agent Distillation (Naive):** Standard agent distillation without curriculum learning or trajectory optimization.

**Implementation:** Parameter-efficient fine-tuning was performed using LoRA (rank=64). Training was conducted for 2 epochs with a batch size of 8 and a learning rate of $2e^{-4}$ on 4 NVIDIA H20 96GB GPUs. During inference, the maximum interaction steps were set to 5. For the MCAG strategy, we set the sampling number $N = 5$ and temperature to 0.4. The initial curriculum learning parameter was set to 0.1. Evaluation metrics included Exact Match (EM) for math tasks and an LLM-as-a-judge approach (using GPT-4o-mini) for factual QA tasks.

### C. Overall Performance Comparison

Experimental results demonstrate that the Curriculum-enhanced Agent Distillation (CAD) method yields significant advantages across multiple tasks. As shown in TableII, CAD consistently outperforms both traditional agent distillation and CoT distillation baselines across all student model scales.

A notable phenomenon is the "Efficiency Leap": small-parameter models trained with CAD perform comparably to traditional distillation models with significantly larger parameters. For instance, the 1.5B parameter CAD model achieves an average score of 30.55%, surpassing the 3B parameter standard CoT distillation model (27.72%) and approaching the 7B baseline. This empirically confirms that the curriculum mechanism and trajectory optimization effectively reduce the learning burden, enabling high-density knowledge transfer.

## V. ANALYSIS

### A. Task Orthogonality and Curriculum Necessity

To elucidate the necessity of our Curriculum-enhanced Agent Distillation (CAD) framework, we visualize the functional distribution of the eight benchmarks within the *TCF, CEC, EIF* metric space. As illustrated in the ternary distribution in Fig 4, the agentic tasks exhibit a prominent tripolar polarization. Specifically, competitive reasoning datasets such as AIME and Olym are densely clustered at the *CEC* vertex, indicating a near-exclusive reliance on deep symbolic logic. In stark contrast, multi-hop retrieval tasks like MuSiQue and HotpotQA gravitate toward the *TCF* pole, emphasizing high-frequency tool invocation. This spatial divergence provides empirical evidence that "retrieval expertise" and "logical reasoning depth" constitute orthogonal optimization objectives. From an optimization perspective, the minimal overlap between these functional domains justifies our curriculum design. A vanilla mixed-training baseline forces the student model to reconcile mutually exclusive gradient directions in a single batch—for instance, the iterative environment feedback required by *EIF*-heavy tasks versus the static logical consistency required by *CEC*-heavy tasks. Our CAD framework mitigates this gradient interference by progressively expanding the training manifold, allowing the model to stabilize on foundational patterns (e.g., Bamboogle or GSM-H) before advancing toward the specialized expertise at the triangular extrema.

### B. Sensitivity Analysis and Training Dynamics

To further investigate the underlying mechanism of our framework, we conduct an exhaustive sensitivity analysis on the difficulty threshold coefficient $\lambda$, which serves as the core hyperparameter governing the curriculum transition. As illustrated in Fig 5, the performance across four distinct benchmarks—GSM-H, HotpotQA, AIME, and MusiQue—consistently exhibits a **characteristic bell-shaped trajectory**. Specifically, we identify an "Optimal Regime" within the narrow interval of $\lambda \in [0.25, 0.35]$.

The empirical results suggest that a lower threshold ($\lambda < 0.25$) introduces high-complexity trajectories too aggressively, overwhelming the student model's initial learning capacity and leading to suboptimal convergence. Conversely, an overly conservative schedule ($\lambda > 0.35$) confines the model to simplistic patterns for an excessive duration, which impedes the acquisition of sophisticated reasoning expertise. The peak performance at $\lambda \approx 0.3$ strikes a critical balance between foundational consolidation and progressive challenge.

Furthermore, the convergence analysis depicted in Fig 4 provides robust evidence for the **stability-enhancing properties** of our approach. While both the baseline and our

TABLE II
MAIN EXPERIMENTAL RESULTS. CAD OUTPERFORMS BASELINES ON MOST TASKS. ABBREVIATIONS: FTP (FIRST-THOUGHT PREFIX), CR (CURRICULUM LEARNING).

| Params | Method | Hotpot | MATH | MuSiQue | Bamboogle | 2Wiki | GSM-H | AIME | Olym | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| *Teacher: Qwen2.5-32B-Instruct* | | | | | | | | | | |
| 32B | COT Prompting | 36.8 | 79.2 | 12.2 | 60.8 | 33.4 | 74.6 | 13.3 | 6.0 | 39.54 |
| | Agent Prompting | 56.4 | 69.2 | 25.2 | 58.4 | 49.8 | 76.4 | 21.1 | 11.5 | 46.00 |
| *Student: Qwen2.5-Instruct* | | | | | | | | | | |
| 7B | COT Prompting | 29.2 | 71.8 | 5.8 | 43.2 | 29.2 | 66.6 | 12.2 | 7.5 | 33.19 |
| | Distill | 31.0 | 72.6 | 9.0 | 44.8 | 26.8 | 67.6 | 10.0 | 6.5 | 33.54 |
| | Distill+RAG | 42.8 | 68.0 | 6.6 | 40.0 | 27.6 | 60.6 | 6.7 | 5.0 | 32.16 |
| | Agent Prompting | 46.8 | 56.0 | 16.8 | 41.6 | 45.6 | 62.2 | 13.3 | 10.0 | 36.54 |
| | Distill | 50.2 | 61.2 | 18.6 | 51.0 | 43.2 | 71.3 | 12.1 | 6.5 | 39.26 |
| | +FTP | 54.0 | 65.6 | 17.3 | 54.0 | 42.9 | 71.1 | 13.4 | 14.2 | 41.56 |
| | **+FTPCR (CAD)** | **54.4** | **67.8** | **19.4** | **55.2** | **45.2** | **72.4** | **15.6** | 11.5 | **42.68** |
| 3B | COT Prompting | 38.6 | 62.8 | 6.2 | 33.6 | 21.6 | 60.2 | 6.7 | 4.5 | 29.27 |
| | Distill | 26.8 | 61.8 | 6.4 | 34.4 | 25.0 | 56.8 | 5.6 | 5.0 | 27.72 |
| | Agent Prompting | 38.6 | 30.5 | 8.8 | 29.6 | 28.8 | 25.8 | 4.4 | 3.0 | 21.20 |
| | Distill | 47.3 | 53.1 | 12.6 | 35.3 | 35.7 | 63.6 | 6.3 | 7.8 | 32.71 |
| | +FTP | 46.6 | 52.4 | 12.8 | 42.7 | 40.6 | 64.4 | 6.7 | 5.5 | 33.96 |
| | **+FTPCR (CAD)** | **48.4** | **57.2** | **15.8** | 38.4 | 40.0 | **65.4** | **15.6** | 7.0 | **35.97** |
| 1.5B | COT Prompting | 17.8 | 47.6 | 3.0 | 21.6 | 19.0 | 49.0 | 1.1 | 3.5 | 20.33 |
| | Distill | 23.8 | 46.4 | 2.0 | 21.6 | 18.4 | 51.0 | 5.6 | 1.5 | 21.28 |
| | Agent Prompting | 8.6 | 22.2 | 1.6 | 10.4 | 10.6 | 9.0 | 1.1 | 0.0 | 7.94 |
| | Distill | 43.0 | 46.8 | 9.0 | 27.2 | 35.6 | 54.8 | 1.1 | 7.0 | 28.06 |
| | +FTP | 43.6 | 46.4 | 8.0 | 30.4 | 32.6 | 60.6 | 7.8 | 3.5 | 29.11 |
| | **+FTPCR (CAD)** | **45.6** | **50.6** | **9.2** | **33.6** | 33.6 | **60.6** | 6.7 | 4.5 | **30.55** |

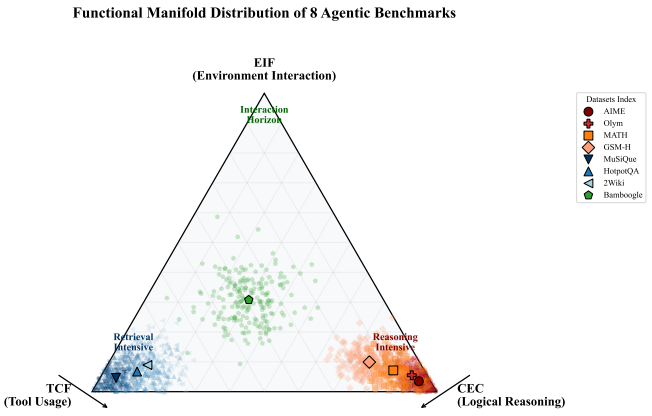


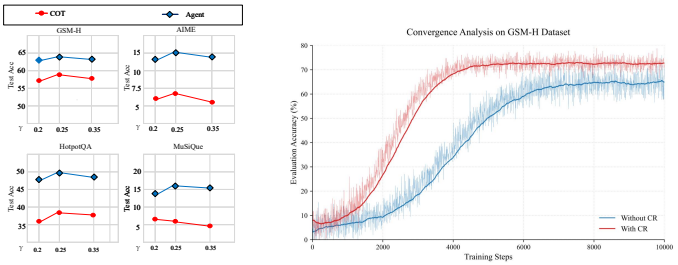

Fig. 3. Functional distribution of agentic benchmarks in the *TCF, CEC, EIF* metric manifold. Each data point represents the normalized difficulty profile of a task trajectory, and solid markers denote the geometric centroids of each dataset. The distribution reveals a pronounced tri-polar polarization: (i) Competitive Reasoning Zone: Datasets like AIME and Olym are densely clustered near the *CEC* vertex, emphasizing deep symbolic logic; (ii) Multi-hop Retrieval Zone: MuSiQue and HotpotQA gravitate toward the *TCF* pole, signifying high-frequency tool invocation; (iii) Balanced/Interaction Zone: Bamboogle and GSM-H occupy the transitional regions. The minimal spatial overlap between these specialized clusters provides empirical evidence for task orthogonality, justifying the necessity of our decoupled curriculum strategy to mitigate gradient interference during joint distillation.

proposed CAD framework exhibit comparable variance during the initial 2,000 steps, a significant bifurcation emerges as task complexity increases. The CAD-enhanced model maintains a remarkably tighter variance band in the late-stage training compared to the "Without CR" baseline. This suggests that the progressive difficulty scaling acts as an **implicit regularizer**, smoothing the optimization landscape and fostering "logical resilience" in the student model. Consequently, our method not only reaches a superior performance ceiling but also demonstrates a more robust and data-efficient learning trajectory.

### *C. Analysis of Retrieval Frequency and Model Performance*

As illustrated in Fig 5, a comparison between the baseline (w/o CR) and our curriculum-refined method (w/ CR) reveals a significant positive correlation between the frequency of tool invocation and final task accuracy. Across the HotpotQA, Bamboogle, and MuSiQue datasets, the CAD method not only induces more frequent external retrievals (indicated by the increased bar heights) but also achieves a synchronous improvement in accuracy (the red trend line consistently outperforms the grey line). This phenomenon strongly evidences that the performance gain is derived not from overfitting, but from the effective utilization of external information, justifying the trade-off of increased computational cost (retrieval steps) for superior reasoning quality.

In-depth analysis suggests that the curriculum learning mechanism effectively mitigates the "retrieval deficiency"




(a) Sensitivity to difficulty coefficient of $\gamma$

(b) Convergence and optimization stability on GSM-H

Fig. 4. Quantitative analysis of curriculum-enhanced distillation on agent benchmarks. (a) Sensitivity to difficulty coefficient $\lambda$: We evaluate the performance across four diverse benchmarks, observing a consistent bell-shaped trajectory. The "Optimal Regime" is identified within the interval of $\lambda \in [0.25, 0.35]$, where $\lambda \approx 0.3$ strikes a critical balance between foundational logic consolidation and progressive reasoning challenge. (b) Convergence and optimization stability on GSM-H: Solid lines denote the smoothed evaluation accuracy, while shaded areas represent raw fluctuations. Compared to the Without CR baseline (blue), our CAD framework (red) exhibits significantly tighter variance and a more robust convergence path in the later stages of training, demonstrating its ability to mitigate gradient instability in complex agent trajectories.

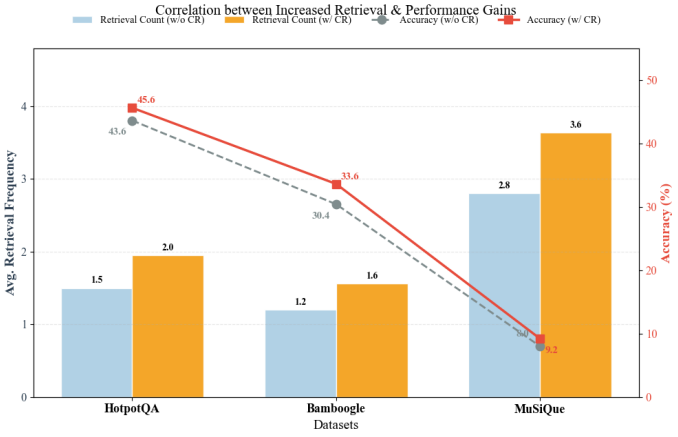



Fig. 5. We compare the average retrieval frequency (represented by bars) and accuracy (represented by lines) of models without curriculum and trajectory optimization (w/o CR) versus those adopting this strategy (w/ CR) across three multi-hop QA datasets: HotpotQA, Bamboogle, and MuSiQue.

and hallucination tendencies common in lightweight student agents. Without progressive guidance, baseline models tend to rely on limited parametric knowledge, leading to errors in long-tail scenarios. In contrast, CAD forces the model to establish robust tool-usage habits during the early stages of the easy-to-hard curriculum. This strategy successfully shifts the model's behavior from passive guessing to active verification, enabling it to autonomously initiate queries when internal confidence is low, thereby correcting behavioral defects in complex settings.

The advantage of our method is particularly pronounced in multi-hop QA tasks (e.g., MuSiQue) that demand high context dependency. The data shows that in these challenging tasks, the increase in retrieval frequency is most substantial and directly translates into accuracy gains. This indicates that the CAD-trained model possesses enhanced multi-step planning capabilities, allowing it to construct complete evidence chains. Rather than settling for partial information from a single retrieval, the model engages in an iterative "retrieve-read-reretrieve" loop to bridge information gaps, demonstrating superior robustness in handling open-domain, high-complexity dynamic tasks.

## VI. Conclusion

In this chapter, we introduced **Curriculum-enhanced Agent Distillation (CAD)**, a novel framework designed to bridge the gap between large language models and lightweight agents. By integrating a curriculum learning mechanism into the distillation process, CAD orchestrates a progressive, easy-to-hard training trajectory that guides student models in mastering complex tool usage and multi-step decision-making.

Extensive empirical evaluations across eight benchmarks—spanning mathematical reasoning (e.g., MATH, AIME) and multi-hop question answering (e.g., HotpotQA)—demonstrate the efficacy of our approach. CAD not only achieves a significant performance margin of 4%–8% over traditional agent distillation baselines in complex tasks but also enables a 1.5B parameter student model to rival the performance of 3B parameter models trained via standard methods.

Our analysis confirms that the synergy between dynamic difficulty scheduling, "First-thought Prefix" guidance, and trajectory compression is pivotal. These components collectively mitigate the optimization instability often associated with learning long-horizon agent trajectories, ensuring efficient and stable knowledge transfer.

In summary, CAD establishes a systematic and viable pathway for developing efficient, robust language agents. By effectively democratizing the reasoning and tool-use capabilities of LLMs into resource-constrained models, this work paves the way for the broader deployment of intelligent agents in real-world edge environments.

## References


[1] Shreyas Basavatia, Keerthiram Murugesan, and Shivam Ratnakar. STARLING: Self-supervised training of text-based reinforcement learning agent with large language models, 2024.

[2] Yizhao Jin, Gregory Slabaugh, and Simon Lucas. ADAPTER-RL: Adaptation of any agent using reinforcement learning, 2024.

[3] Xiaohan Huang, Dongjie Wang, Zhiyuan Ning, Ziyue Qiao, Qingqing Long, Haowei Zhu, Yi Du, Min Wu, Yuanchun Zhou, and Meng Xiao. Collaborative multi-agent reinforcement learning for automated feature transformation with graph-driven path optimization. *arXiv preprint arXiv:2504.17355*, 2025.

[4] Subash Neupane, Sudip Mittal, and Shahram Rahimi. Towards a hipaa compliant agentic ai system in healthcare. *arXiv preprint arXiv:2504.17669*, 2025.

[5] Zhuofeng Wu, Richard He Bai, Aonan Zhang, Jiatao Gu, VG Vinod Vydiswaran, Navdeep Jaitly, and Yizhe Zhang. Divide-or-conquer? which part should you distill your llm? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2572–2585, 2024.

[6] Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, et al. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*, 2024.

[7] Lei Li, Yongfeng Zhang, and Li Chen. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 1348–1357, 2023.

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

[9] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*, 2023.

[10] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[11] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[12] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[13] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

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

[16] Rajesh Upadhayaya, Manish Raj Osti, Zachary Smith, and Chritopher Kottmyer. Efficient llm context distillation. *arXiv preprint arXiv:2409.01930*, 2024.

[17] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.

[18] 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*, volume 34, pages 5191–5198, 2020.

[19] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

[20] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

[21] Jia Li, Ge Li, Yongmin Li, and Zhi Jin. Structured chain-of-thought prompting for code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–23, 2025.

[22] Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*, 2025.

[23] Edward Y Chang. *Multi-LLM Agent Collaborative Intelligence: The Path to Artificial General Intelligence*. ACM, 2025.

[24] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.

[25] Xiaofeng Zhou, He-Yan Huang, and Lizi Liao. Debate, reflect, and distill: Multi-agent feedback with tree-structured preference optimization for efficient language model enhancement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9122–9137, 2025.

[26] Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.

[27] Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *arXiv preprint arXiv:2503.12759*, 2025.

[28] Yang-Geng Fu, Xinlong Chen, Shuling Xu, Jin Li, Xi Yao, Ziyang Huang, and Ying-Ming Wang. Gsscl: A framework for graph self-supervised curriculum learning based on clustering label smoothing. *Neural Networks*, 181:106787, 2025.

[29] Zhiheng Ma, Anjia Cao, Funing Yang, Yihong Gong, and Xing Wei. Curriculum dataset distillation. *IEEE Transactions on Image Processing*, 2025.

[30] Lingyuan Liu and Mengxiang Zhang. Being strong progressively! enhancing knowledge distillation of large language models through a curriculum learning framework. *arXiv preprint arXiv:2506.05695*, 2025.

[31] Liang Chen, Bo Shen, and Jiale Hong. A multi-task deep reinforcement learning framework based on curriculum learning and policy distillation for quadruped robot motor skill training. *Systems Science & Control Engineering*, 13(1):2498914, 2025.

[32] Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. Fastcurl: Curriculum reinforcement learning with stage-wise context scaling for efficient training r1-like reasoning models. *arXiv preprint arXiv:2503.17287*, 2025.

[33] Sérgio S Mucciaccia, Thiago M Paixão, Filipe Mutz, Alberto F De Souza, Claudine S Badue, and Thiago Oliveira-Santos. Pt-hotpotqa: Evaluating multi-hop question answering on original and portuguese-translated datasets using llms. *Journal of the Brazilian Computer Society*, 31(1):871–883, 2025.

[34] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics.

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

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

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

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

[39] AI-Mo. Aime. *https://huggingface.co/datasets/AI-MO/aimo-validation-aime*, 2024.

[40] Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.