KALE: Enhancing Knowledge Manipulation in Large Language Models via Knowledge-aware Learning

Anonymous ACL submission

Abstract

Despite the impressive performance of large language models (LLMs) pretrained on vast knowledge corpora, advancing their knowledge manipulation—the ability to effectively **recall, reason, and transfer relevant knowledge**—remains challenging. Existing methods mainly leverage Supervised Fine-Tuning (SFT) on labeled datasets to enhance LLMs' knowledge manipulation ability. However, we observe that SFT models still exhibit the *known&incorrect* phenomenon, where they explicitly possess relevant knowledge for a given question but fail to leverage it for correct answers. To address this challenge, we propose KALE (**K**nowledge-**A**ware **LE**arning)—a post-training framework that leverages knowledge graphs (KGs) to generate high-quality rationales and enhance LLMs' knowledge manipulation ability. Specifically, KALE first introduces a **K**nowledge-**I**nduced (KI) data synthesis method that efficiently extracts multi-hop reasoning paths from KGs to generate high-quality rationales for question-answer pairs. Then, KALE employs a **K**nowledge-**A**ware (KA) fine-tuning paradigm that enhances knowledge manipulation by internalizing rationale-guided reasoning through minimizing the KL divergence between predictions with and without rationales. Extensive experiments on eight popular benchmarks across six different LLMs demonstrate the effectiveness of KALE, achieving accuracy improvements of up to 11.72% and an average of 4.18%.

1 Introduction

Standing out as versatile tools with vast knowledge repositories, large language models (LLMs), such as ChatGPT (OpenAI, 2023), Deepseek R1 (DeepSeek-AI, 2024), and Gemini 3 (Google DeepMind Team, 2023), demonstrate remarkable power across a wide range of domains (Zhao et al., 2021; Zhang et al., 2026). However, the most capable LLMs produce errors, even when the relevant knowledge is encoded within them, indicating struggles to flexibly manipulate relevant knowledge during inference (Allen-Zhu and Li, 2025).

Recently, extensive research efforts have focused on enhancing LLMs' knowledge manipulation abilities. Among these methods, Supervised Fine-Tuning (SFT) has been widely adopted as a standard post-training approach (Wei et al., 2022a). The key idea of SFT is to adapt pretrained LLMs to specific tasks by training on labeled datasets, thereby optimizing their parameters for task-specific performance (Zhang et al., 2023). Several studies have also explored variations of SFT. Dual-stage Mixed Fine-Tuning (DMT) (Dong et al., 2024) expands SFT datasets to achieve a balance between general and specialized abilities. KG-SFT (Chen et al., 2025a) utilizes knowledge graphs to filter SFT data to enhance LLMs' ability in knowledge-intensive tasks. Extensive studies demonstrate the effectiveness and versatility of SFT methods (Xie et al., 2024; Dong et al., 2024).

Despite the multiple benefits of SFT methods, LLMs fine-tuned via SFT still exhibit the *known&incorrect* phenomenon— **LLMs possess relevant knowledge but cannot effectively manipulate it to answer questions correctly.** This phenomenon mainly stems from two limitations in the SFT process: (i) the lack of training data with explicit reasoning rationales. Such high-quality reasoning data is scarce in many domains, and manual creation requires substantial effort, posing significant barriers to broader LLM applications (Li et al., 2025); and (ii) the insufficient ability to manipulate task-relevant knowledge. SFT methods fine-tune LLMs using labeled datasets to learn specific patterns through explicit input-output pairs. However, LLMs often overly rely on these explicit mappings, which restricts their ability to effectively manipulate task-relevant knowledge (Luo et al., 2024). As shown in Figure 1, although the LLM possesses the knowledge that the true color of the Sun is white, it

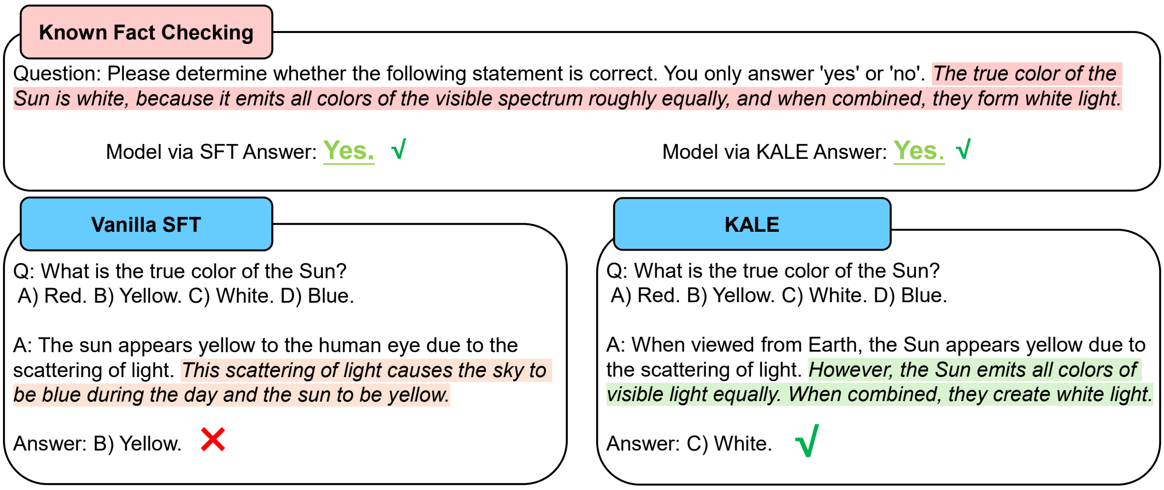

Figure 1: While both post-trained LLMs know relevant knowledge, the LLM via SFT still cannot recall the knowledge to answer. In contrast, KALE effectively recalls the knowledge and answers correctly. We use Mistral 7B (Jiang et al., 2023a) as an example, and more cases are in Appendix B.

fails to answer correctly after SFT. Therefore, even after sufficient SFT process, LLMs still struggle to effectively manipulate task-relevant knowledge to answer correctly (Allen-Zhu and Li, 2025).

To address these challenges, we propose KALE (**K**nowledge-**A**ware **LE**arning)—a novel post-training framework to boost LLMs' knowledge manipulation ability. KALE consists of two components: (i) knowledge-induced data synthesis (**KI**) that leverages knowledge graphs to generate high-quality rationales, and (ii) knowledge-aware fine-tuning (**KA**) that enhances knowledge manipulation by minimizing KL divergence between outputs with and without rationales. Specifically, for a given question-answer pair, KALE **first** identifies named entities and extracts reasoning paths from the question to answer using a multi-path A* algorithm over knowledge graphs. **Then**, KALE feeds the question-answer pair and reasoning paths to an LLM to generate rationales. **Finally**, rather than learning specific patterns through explicit supervised input-output pairs, KALE minimizes the KL divergence (Kullback and Leibler, 1951) between the output distributions of LLMs with and without rationales. This encourages the two distributions to be aligned, allowing LLMs to effectively manipulate task-relevant knowledge when rationales are absent during inference.

In summary, our key contributions include:

(i) **Identification of the known&incorrect phenomenon.** We identify a critical limitation of existing SFT methods: LLMs fine-tuned via SFT often possess relevant knowledge but fail to manipulate it to answer correctly, revealing fundamental challenges in knowledge manipulation during post-training.

(ii) **A novel knowledge-aware post-training method.** We propose KALE, a unified framework that addresses the known&incorrect phenomenon through: (i) knowledge-induced data synthesis to generate high-quality reasoning rationales via KGs, and (ii) knowledge-aware fine-tuning to enable effective knowledge manipulation by aligning output distributions with and without rationales.

(iii) **Significant improvement and versatility.** We conduct extensive experiments on eight benchmarks across six LLMs to demonstrate the effectiveness of KALE, yielding a maximum accuracy improvement of 11.72% and an average of 4.18% over SFT baselines.

2 Related Work

2.1 Text Data Augmentation Methods

With the advent of LLMs, data augmentation has undergone a significant transformation (Ding et al., 2024). LLMs have shown remarkable abilities in generating high-quality text, which provides advantages in data augmentation tasks (Deng et al., 2023; Fang et al., 2023). AugGPT (Dai et al., 2023) leverages the generative power of LLMs to rephrase questions in SFT data. GPT3Mix (Yoo et al., 2021) extends augmentation abilities of LLMs by using few-shot prompting to generate questions semantically similar to the SFT data. StaR (Zelikman et al., 2024) utilizes a self-taught mechanism to let LLMs provide internal thoughts. While existing augmentation methods primarily focus on expanding the data quantity of the original data, they lack the multi-hop logic rationales, KALE can effectively generate textual rationales underlying the Q&A pair.

2.2 KG Retrieval Generation Methods

Knowledge graphs provide structured, factual knowledge that complements the unstructured, text-based knowledge encoded in LLMs (Pan et al., 2024). Recent research has explored integrating KGs to enhance LLMs' reasoning ability through retrieval-augmented approaches (Zhao et al., 2024; Sun et al., 2023; Edge et al., 2024). Think-on-Graph (ToG) (Sun et al., 2023) employs iterative beam search over a KG to guide LLM reasoning. KGR (Guan et al., 2024) augments LLM responses with factual statements retrieved from KGs. KAPING (Baek et al., 2023) enhances zero-shot Q&A by appending retrieved facts to prompts. StructGPT (Jiang et al., 2023b) employs an iterative reading-then-reasoning framework over structured data. GraphRAG (Edge et al., 2024) integrates KG traversal to retrieve relationships from graph-indexed data. These retrieval-based methods require additional retrieval operations from knowledge bases during inference, introducing additional latency. In contrast, KALE internalizes knowledge-grounded reasoning paths during training, eliminating the need for retrieval during inference (please refer to Appendix H.2 for inference time comparison for each backbone and baseline).

2.3 SFT Variant Methods

Supervised Fine-Tuning (SFT) has become a standard approach to adapt pre-trained LLMs to specific downstream tasks (Ouyang et al., 2022). Recent work has explored various SFT strategies to improve model performance and generalization. Dual-stage Mixed fine-Tuning (DMT) (Dong et al., 2024) improves the general ability of LLMs by balancing task-specific and general knowledge during fine-tuning. Self-Distillation Fine-Tuning (SDFT) (Yang et al., 2024b) leverages a distilled dataset generated by the model itself to maintain the original distribution and reduce catastrophic forgetting (Kirkpatrick et al., 2016). KG-SFT (Chen et al., 2025a) uses KGs to filter high-quality SFT data for knowledge-intensive tasks. However, these SFT-based methods learn from explicit input-output mappings, which limits their flexibility in manipulating task-relevant knowledge when faced with varied input queries. In contrast, KALE employs a knowledge-aware fine-tuning paradigm that aligns LLM distributions with and without rationales, enabling more flexible and dynamic knowledge manipulation during inference.

3 Preliminaries

3.1 Notations

We denote $\mathbf{x}^{\text{ins}}$ as instructions for downstream tasks, $\mathbf{x}^{\text{que}}$ as queries, $\mathbf{x}^{\text{ans}}$ as answers, and $\mathbf{x}^{\text{rats}}$ as rationales. We define two types of input prompts for LLMs: one includes the rationale, represented as $(\mathbf{x}^{\text{ins}}, \mathbf{x}^{\text{que}}, \mathbf{x}^{\text{rats}})$, and the other excludes the rationale, represented as $(\mathbf{x}^{\text{ins}}, \mathbf{x}^{\text{que}})$. Let $\mathcal{E}_q = [\mathbf{e}_{q_1}, \mathbf{e}_{q_2}, \mathbf{e}_{q_3}, \ldots]$ denote the question entity list extracted from $\mathbf{x}^{\text{que}}$, and $\mathcal{E}_a = [\mathbf{e}_{a_1}, \mathbf{e}_{a_2}, \mathbf{e}_{a_3}, \ldots]$ denote the answer entity list extracted from $\mathbf{x}^{\text{ans}}$. Let $\mathcal{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \ldots]$ denote the reasoning path list connecting entities in $\mathcal{E}_q$ to entities in $\mathcal{E}_a$, where $\mathbf{p}_i$ represents the i-th reasoning path.

3.2 A* Algorithm

A* algorithm (Hart et al., 1968; Goldberg and Harrelson, 2005) is a heuristic search algorithm that extends traditional shortest path algorithms by incorporating a heuristic function to guide the search. Unlike uniform search strategies, A* prioritizes nodes with lower estimated total cost, reducing the search space:

$$f(\mathbf{e}) = g(\mathbf{e}) + h(\mathbf{e}) \tag{1}$$

where $g(\mathbf{e})$ is the cost of the current shortest path from start entity $\mathbf{e}_{start}$ to entity $\mathbf{e}$, and $h(\mathbf{e})$ is a heuristic function estimating the cost from $\mathbf{e}$ to the target entity $\mathbf{e}_{end}$. $f(\mathbf{e})$ represents the estimated total cost for reaching the target through $\mathbf{e}$.

4 Method

4.1 Knowledge-induced Data Synthesis

Answering a question often requires integrating multiple knowledge fragments. For instance, for the question *"What is the true color of the Sun?"* and answer *"White"*, it involves multiple pieces of knowledge such as: **(i)** "The Sun emits all colors of the visible spectrum," **(ii)** "The combination of all visible light produces white light," and **(iii)** "The balanced intensity distribution of sunlight integrates into white light." The fragmented nature of such knowledge in pre-training

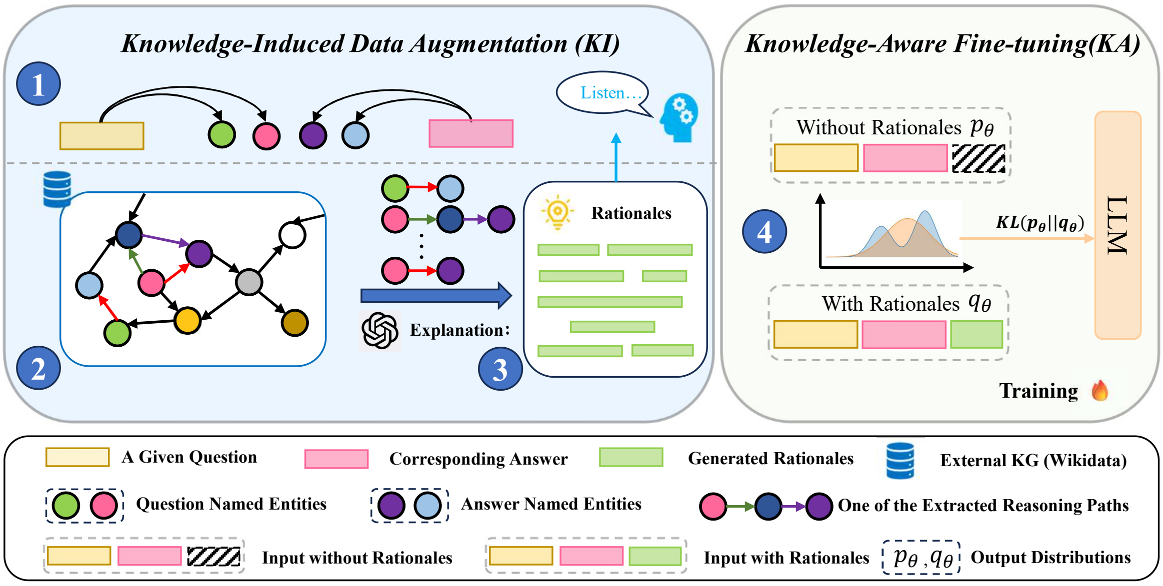

Figure 2: An overview of KALE. For a given Q&A pair, the workflow of KALE is as follows. **(1)** Perform named entity recognition to extract relevant question and answer entities. **(2)** Search for reasoning paths via the proposed multi-path A* algorithm. **(3)** Combine reasoning paths with the Q&A pair to generate the corresponding rationales via GPT-4o. **(4)** Align the LLM's output distributions with and without rationales via knowledge-aware fine-tuning.

data makes it challenging for LLMs to manipulate relevant knowledge. In contrast, KGs provide a way to organize fragmented knowledge into structured relationships. Specifically, this knowledge can be formalized into a reasoning path: "*the Sun–emits–>full-spectrum light–integrates_into–>white light*," which corresponds to a series of interconnected triples within a KG, including [the Sun, emits, full-spectrum light] and [full-spectrum light, integrates into, white light]. Building upon this, we propose knowledge-induced data synthesis (KI) to generate rationales.[1]

Specifically, KALE **first** performs named entity recognition separately on the question and the answer, resulting in the question entity list $\mathcal{E}_q = \{\text{the Sun}\}$ and the answer entity list $\mathcal{E}_a = \{\text{white}\}$. **Then**, KALE leverages these entities to search for reasoning paths in a KG. Conducting a full breadth-first search (BFS) from the question entities to the answer entities in a large KG (e.g., Wikidata[2]) is time-consuming. For instance, the extraction of reasoning paths from the AbsR training set (Xiong et al., 2024) **requires over one week**. Therefore, we propose an efficient multi-path A* algorithm to extract reasoning paths. It requires **less than 4 hours** to extract all reasoning paths on the same set. Specifically, we adopt a small set of *anchor entities*. For a given entity pair $\mathbf{e}_q$ and $\mathbf{e}_a$ in $\mathcal{E}_q$ and $\mathcal{E}_a$, we select k anchor entities by randomly sampling from the m-hop neighbors of $\mathbf{e}_a$, thereby extracting a local subgraph around the answer entity. For each anchor, we conduct a limited 3-step BFS to pre-compute partial distances, which serve as a lower bound for the remaining path cost in A*.

Formally, let $g(\mathbf{e})$ be the accumulated cost (the number of edges traversed) from the start entity to the current entity $\mathbf{e}$, and $h(\mathbf{e})$ be the heuristic function estimating the cost from $\mathbf{e}$ to the answer entity $\mathbf{e}_a$. We define the priority function as $f(\mathbf{e}) = g(\mathbf{e}) + h(\mathbf{e})$, where $f(\mathbf{e})$ is the priority value in A*. To ensure $h(\mathbf{e})$ does not overestimate the actual distance, we use the maximum of anchor-based lower bounds derived from the BFS. Let $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ be k anchor entities. We pre-compute $\text{dist}(\alpha_i, \mathbf{e})$ up to depth d; if $\mathbf{e}$ is not reachable within d steps, we set $\text{dist}(\alpha_i, \mathbf{e}) = \infty$. Likewise, we compute $\text{dist}(\alpha_i, \mathbf{e}_a)$ for each anchor. Then we let

$$h(\mathbf{e}) = \max_{1 \le i \le k} \Big[\text{dist}(\alpha_i, \mathbf{e}_a) - \text{dist}(\alpha_i, \mathbf{e})\Big]^+ \quad (2)$$

where $[x]^+ = \max(x, 0)$ ensures non-negative values. Intuitively, if $\mathbf{e}$ is already closer to the answer entity than α_i, this difference provides a nontrivial lower bound; otherwise, it contributes

[1]We only generate rationales during training. This example is for illustration; KALE does not introduce additional overhead during inference (please refer to Figure 1).

[2]We use Wikidata as the default external KG to extract all reasoning paths. To evaluate the robustness of KALE, we report results using alternative KGs in Appendix H.6.

zero and does not lead to overestimation. We prove the admissibility of our multi-path A* algorithm via the proposed heuristic function in Appendix E. This heuristic design is simple yet efficient for reasoning path retrieval in a large KG. We can also apply KG embedding-based methods (Rossi et al., 2021; Zhu et al., 2023) to incorporate semantic information, and we leave it as future work.

To retrieve multiple reasoning paths, we extend the standard A* algorithm by incorporating a priority queue $\mathcal{Q}$, which stores multiple paths leading to the same entity. Each entry in $\mathcal{Q}$ is a tuple $(f(\mathbf{e}), g(\mathbf{e}), \mathbf{e}, p^i_{\mathbf{e}})$, where $p^i_{\mathbf{e}}$ is the i-th path from the start entity $\mathbf{e}_q$ to the current entity $\mathbf{e}$. Algorithm 1 in Appendix F provides the pseudocode for the overall procedure. After obtaining $\mathcal{P}$, we combine the Q&A pair and $\mathcal{P}$ as input to GPT-4o, prompting it to generate the rationale $\mathbf{x}^{\text{rats}}$ underlying the Q&A pair (Appendix C provides prompt details). For example, given the extracted reasoning path "*the Sun–emits–>full-spectrum light–integrates_into–>white light*," the generated rationale is "*The Sun emits light that contains the entire visible spectrum. When these different colors of light are combined, they create white light.*" These rationales offer high-quality textual reasoning data from question to answer, enabling better understanding of the underlying logic and correlations. We include more examples of reasoning paths and rationales in Appendix D to provide a comprehensive understanding of KALE.

4.2 Knowledge-aware Fine-tuning Paradigm

When confronted with a question, humans typically retrieve relevant experiences and knowledge, reason based on them, and then provide a response (Buckner and Wheeler, 2001; Yadav et al., 2022). Motivated by this, we propose a simple yet effective learning paradigm called knowledge-aware fine-tuning, which encourages LLMs to recall relevant knowledge and reason with it before generating a response.

Formally, consider an LLM denoted by $\mathcal{M}$ with parameters θ and input $\mathbf{x}^{\text{inp}} = (\mathbf{x}^{\text{ins}}, \mathbf{x}^{\text{que}}, \mathbf{x}^{\text{ans}})$, where $\mathbf{x}^{\text{ins}}$ denotes instructions, and $\mathbf{x}^{\text{que}}$ and $\mathbf{x}^{\text{ans}}$ denote the Q&A pair. The LLM models the conditional probability of output $\mathbf{x}^{\text{out}}$. We consider two probabilities, which differ in whether rationales are included as input:

$$\mathcal{M}_{w/o} = -\sum_t \log p_\theta(\mathbf{x}^{\text{out}}_t \mid \mathbf{x}^{\text{inp}}, \mathbf{x}^{\text{out}}_{<t}) \tag{3a}$$

$$\mathcal{M}_{w} = -\sum_t \log q_\theta(\mathbf{x}^{\text{out}}_t \mid \mathbf{x}^{\text{inp}}, \mathbf{x}^{\text{rats}}, \mathbf{x}^{\text{out}}_{<t}) \tag{3b}$$

Equation (3a) represents the classical process of LLM generation, where a given instruction and query are provided as input, and the LLM produces an output. We aim for the LLM to manipulate learned knowledge and reason with it. In Equation (3b), we use the generated rationales $\mathbf{x}^{\text{rats}}$ as input to the LLM to enable better recall of knowledge fragments relevant to the question.

Therefore, our goal is to enable the LLM to implicitly leverage relevant knowledge based on the instruction and query before generating a response. To achieve this, we propose knowledge-aware fine-tuning to minimize the divergence between the two distributions in Equations (3a) and (3b) as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x}^{\text{inp}}, \mathbf{x}^{\text{out}}, \mathbf{x}^{\text{rats}})} \left[\text{KL} \left(p_\theta \| q_\theta \right) \right] \tag{4}$$

where $\text{KL}(\cdot\|\cdot)$ denotes the KL divergence. We use two distributions: p_θ (i.e., $p_\theta(\mathbf{x}^{\text{out}}_t \mid \mathbf{x}^{\text{inp}}, \mathbf{x}^{\text{out}}_{<t})$), which is updated during training, and q_θ (i.e., $q_\theta(\mathbf{x}^{\text{out}}_t \mid \mathbf{x}^{\text{inp}}, \mathbf{x}^{\text{rats}}, \mathbf{x}^{\text{out}}_{<t})$), which is fixed and serves as the alignment target. By minimizing the KL divergence in Equation (4), KALE does not require outputs without rationales to exactly match those produced with rationales. Instead, it encourages the two distributions to align, which enables the LLM to flexibly manipulate task-relevant knowledge when rationales are absent during inference.

5 Experiments

We aim to evaluate the effectiveness of KALE in enhancing LLMs' knowledge manipulation ability and its versatility across different settings. To achieve this, we organize the experiments into the following parts:

- To demonstrate the superiority, we conduct comparative experiments on eight benchmarks across six different LLMs.
- To investigate the contribution of each component, we conduct ablation studies.
- To provide more insights, we conduct case studies on the known&incorrect phenomenon and ratios of augmented rationales.

- To demonstrate the versatility, we evaluate KALE in knowledge-intensive domains across six different languages in Appendix H.1.
- To analyze KALE's real-world deployability, we evaluate its inference time and hyperparameter sensitivity in Appendix H.2 and H.3.
- To demonstrate the effectiveness of rationales:
  - (i) We employ different KGs to generate reasoning paths in Appendix H.6.
  - (ii) We use rationales of other LLMs to show robustness in Appendix H.10.
  - (iii) We generate irrelevant and contrastive rationales in Appendix H.11.
  - (iv) We evaluate the quality of the generated rationales in Appendix H.12.
- To provide an in-depth understanding, we conduct comparisons including (i) combining KALE with SFT, (ii) evaluations on open-ended settings, (ii) comparisons with thinking-style models, self-taught settings, and GRPO, (iv) prompt distillation, and (v) joint training settings in Appendix H.13 to H.19.

5.1 Experimental Setup

Models and Benchmarks. We use **six** open-source LLMs ranging from 7B to 32B parameters, including LlaMA3 8B (Team, 2024), Mistral 7B (Jiang et al., 2023a), Qwen2.5 32B (Yang et al., 2024a), Gemma2 9B (Riviere et al., 2024), OLMOE 7B (Muennighoff et al., 2024), and Orca2 7B (Mitra et al., 2023). Experiments are conducted on 8 NVIDIA SXM A100 80G GPUs for models under 32B parameters, and on 16 NVIDIA SXM A100 80G GPUs for 32B models. We evaluate on benchmarks for **logical reasoning**, including AbsR (Xiong et al., 2024), Commonsense (denoted as Common) (Xiong et al., 2023), and Big Bench Hard (BBH) (Suzgun et al., 2023); **reading comprehension**, including RACE-H and RACE-M (Lai et al., 2017); and **natural language understanding**, including MMLU (Hendrycks et al., 2021), ARC-c, and ARC-e (Bhakthavatsalam et al., 2021). We use **accuracy** as the evaluation metric. More details and baseline descriptions are provided in Appendix G.

5.2 Main Results

Table 1 presents results using three representative LLMs at different scales: LlaMA3 8B, Mistral 7B, and Qwen2.5 32B. Additional results for three other open-source LLMs are provided in Table 13, Figures 20, and 21 in Appendix H.5 to demonstrate the versatility of KALE. From Table 1, we observe that KALE outperforms other state-of-the-art baselines across all three LLMs by a substantial margin. **Notably, KALE achieves a maximum accuracy improvement of 11.72% on the AbsR benchmark with Qwen2.5 32B as the backbone.**

We also observe that traditional SFT-based and data augmentation methods yield marginal improvements on downstream tasks, particularly when applied to larger and more powerful LLMs (e.g., only a 1.39% improvement on the BBH benchmark when using Qwen2.5 32B). In contrast, KALE delivers consistent and significant improvements for larger LLMs. This indicates that as LLMs scale up and become more capable, SFT-based methods that focus on learning input-output patterns or data augmentation methods that merely increase data quantity are suboptimal for further enhancing LLMs. **In contrast, KALE improves LLMs' ability to manipulate knowledge, achieving significantly better results for larger LLMs.**

5.3 Ablation Study

To investigate the contribution of each component within KALE, we conduct ablation studies and present the results in Table 2. More results for the other three LLMs are provided in Table 13 in Appendix H.7.

Ablation on Rationale Generation We denote KALE without **K**nowledge-**I**nduced (KI) data synthesis as $\text{KALE}_{\text{w/o KI}}$. That is, we do not utilize our proposed multi-path A* algorithm to provide reasoning paths. Instead, we directly input the Q&A pair and prompt the LLM to generate rationales. As shown in Table 2, we observe that directly prompting LLMs to generate rationales without reasoning paths leads to performance degradation. Notably, when using Mistral 7B as the backbone, the degradation on the ARC-e dataset reaches 12.50%. This demonstrates that the extracted reasoning paths effectively capture the reasoning process, contributing to the generation of higher-quality rationales.

Ablation on KL Divergence We denote KALE without **K**nowledge-**A**ware (KA) fine-tuning as $\text{KALE}_{\text{w/o KA}}$. That is, we directly train the LLM with rationale data generated from the KG using cross-entropy loss as the objective function. We

Table 1: Results of our KALE using LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as backbone models (for more results of different backbone models, please see Appendix H.5). We **bold** the best results and underline the suboptimal results for each backbone model.

Backbone	Category	Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
LlaMA3 8B	Prompt-based	**Vanilla**	62.68	66.79	69.90	58.72	55.88	46.54	53.35	57.02
		CoT	63.15	71.67	69.34	54.67	56.83	48.55	54.31	57.02
	Retrieval-based	**TOG**	65.98	69.93	72.23	61.87	56.97	48.81	58.60	59.80
		StructGPT	65.35	70.50	73.34	62.32	58.87	49.58	60.03	60.86
		GraphRAG	75.83	74.83	75.76	61.51	57.28	55.83	60.89	69.57
	SFT-based	**SFT**	67.77	68.23	71.74	59.79	58.00	45.39	56.17	58.91
		SDFT	76.15	74.91	71.44	62.24	58.78	52.37	56.88	61.03
		DMT	74.57	70.82	72.84	61.43	59.11	50.14	61.46	60.64
		MeanLearn	71.09	72.53	74.53	63.39	58.79	50.61	60.03	61.84
		KG-SFT	78.20	73.12	79.55	63.09	58.79	53.68	64.98	62.47
	Augmented-based	**STaR**	69.95	71.50	70.99	58.20	53.41	50.07	61.21	64.32
		AugGPT	64.45	72.22	75.29	55.12	56.82	51.90	59.21	60.16
		GPT3Mix	68.27	70.57	74.24	61.33	57.79	53.92	61.03	62.67
		KALE (ours)	**83.62**	**81.23**	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**
Mistral 7B	Prompt-based	**Vanilla**	62.35	52.05	68.31	39.15	37.43	28.68	50.14	55.92
		CoT	67.18	58.45	66.08	36.94	43.57	31.60	55.15	58.98
	Retrieval-based	**TOG**	64.60	57.25	70.41	50.78	41.35	31.29	52.20	56.96
		StructGPT	65.17	57.94	69.28	46.11	44.94	32.98	55.69	60.10
		GraphRAG	68.26	57.76	71.93	48.24	45.53	35.12	57.15	62.60
	SFT-based	**SFT**	68.48	55.89	71.55	44.14	48.86	34.90	57.09	61.00
		SDFT	73.82	61.01	73.61	51.84	52.19	34.97	64.32	65.53
		DMT	73.22	57.00	72.85	49.71	50.49	35.89	61.64	64.42
		MeanLearn	70.97	64.42	71.55	47.83	50.95	35.58	61.06	64.42
		KG-SFT	72.39	65.96	72.94	54.55	52.10	34.20	61.15	63.37
	Augmented-based	**STaR**	70.02	57.85	74.53	49.80	41.02	35.89	55.09	59.12
		AugGPT	65.28	59.73	72.77	48.24	40.24	33.21	57.75	59.96
		GPT3Mix	59.72	61.69	71.93	53.97	39.84	36.04	56.75	60.10
		KALE (ours)	**76.90**	**71.59**	**77.95**	**59.05**	**54.21**	**39.26**	**67.98**	**70.06**
Qwen2.5 32B	Prompt-based	**Vanilla**	66.35	75.09	80.10	65.52	80.47	69.01	71.47	76.95
		CoT	68.72	76.79	82.07	66.34	81.65	69.79	73.58	77.64
	Retrieval-based	**TOG**	74.64	80.55	84.13	68.63	83.27	72.09	74.12	78.34
		StructGPT	74.17	82.43	83.29	71.58	83.41	71.93	75.56	77.92
		GraphRAG	75.24	80.20	84.18	69.00	84.85	73.20	75.84	77.37
	SFT-based	**SFT**	72.03	79.61	83.33	67.89	82.82	70.40	73.99	79.39
		SDFT	73.34	80.80	84.30	71.25	84.13	71.01	74.59	80.71
		DMT	75.24	81.48	86.07	70.43	85.17	73.62	75.72	80.01
		MeanLearn	71.09	76.37	84.18	69.12	83.61	72.85	74.64	81.82
		KG-SFT	78.91	78.41	84.13	69.62	84.26	72.39	74.80	80.43
	Augmented-based	**STaR**	72.99	83.87	84.60	69.21	85.24	73.16	76.30	80.43
		AugGPT	78.91	84.47	86.27	68.96	85.04	71.93	77.16	81.89
		GPT3Mix	80.10	85.23	87.33	69.53	85.69	73.47	76.21	80.77
		KALE (ours)	**91.82**	**89.93**	**94.90**	**75.02**	**88.59**	**77.91**	**81.76**	**86.70**

observe that training LLMs with cross-entropy loss to match outputs with rationales does not achieve satisfactory results. Specifically, when using Mistral 7B as the backbone on the ARC-e dataset, KALE$_{\text{w/o KA}}$ results in a 14.90% degradation. This demonstrates the effectiveness of the KL divergence for better knowledge manipulation.

5.4 Case Study

Known&incorrect Phenomenon As shown in Figure 3, LLMs trained via SFT still exhibit the known&incorrect phenomenon. We provide a detailed analysis of six different LLMs trained with SFT and KALE (see Appendix H.8 for results of other baselines). We use the known fact checking process in Figure 1 (see Appendix C for prompt details) to categorize LLMs' responses given that LLMs already possess relevant knowledge: (i) ***Known&correct***: LLMs possess the knowledge and correctly answer the question, indicating successful knowledge manipulation. (ii) ***Known&incorrect***: LLMs possess the knowledge

Table 2: Results of the ablation study of KALE, using LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as backbones (We provide more results for the other three backbones in Appendix H.7).

Backbone	Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
LlaMA3 8B	KALE$_{\text{w/o KI}}$	78.91$_{\downarrow 4.71}$	76.79$_{\downarrow 4.44}$	81.65$_{\downarrow 4.80}$	65.52$_{\downarrow 0.17}$	60.09$_{\downarrow 3.18}$	55.21$_{\downarrow 2.12}$	64.15$_{\downarrow 4.46}$	69.50$_{\downarrow 4.62}$
	KALE$_{\text{w/o KA}}$	73.93$_{\downarrow 9.69}$	75.26$_{\downarrow 5.97}$	78.70$_{\downarrow 7.75}$	63.06$_{\downarrow 2.63}$	60.74$_{\downarrow 2.53}$	53.68$_{\downarrow 3.65}$	60.03$_{\downarrow 8.58}$	64.76$_{\downarrow 9.36}$
	KALE	**83.62**	**81.23**	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**
Mistral 7B	KALE$_{\text{w/o KI}}$	71.09$_{\downarrow 5.81}$	66.30$_{\downarrow 5.29}$	65.45$_{\downarrow 12.50}$	57.58$_{\downarrow 1.47}$	52.58$_{\downarrow 1.63}$	36.81$_{\downarrow 2.45}$	64.95$_{\downarrow 3.03}$	66.85$_{\downarrow 3.21}$
	KALE$_{\text{w/o KA}}$	65.64$_{\downarrow 11.26}$	63.91$_{\downarrow 7.68}$	63.05$_{\downarrow 14.90}$	56.84$_{\downarrow 2.21}$	49.05$_{\downarrow 5.16}$	35.74$_{\downarrow 3.52}$	62.78$_{\downarrow 5.20}$	64.00$_{\downarrow 6.06}$
	KALE	**76.90**	**71.59**	**77.95**	**59.05**	**54.21**	**39.26**	**67.98**	**70.06**
Qwen2.5 32B	KALE$_{\text{w/o KI}}$	87.32$_{\downarrow 4.50}$	87.03$_{\downarrow 2.90}$	89.98$_{\downarrow 4.92}$	71.01$_{\downarrow 4.01}$	86.87$_{\downarrow 1.72}$	75.15$_{\downarrow 2.76}$	78.04$_{\downarrow 3.72}$	83.57$_{\downarrow 3.13}$
	KALE$_{\text{w/o KA}}$	82.94$_{\downarrow 8.88}$	85.32$_{\downarrow 4.61}$	88.38$_{\downarrow 6.52}$	70.43$_{\downarrow 4.59}$	84.91$_{\downarrow 3.68}$	76.69$_{\downarrow 1.22}$	77.82$_{\downarrow 3.94}$	82.94$_{\downarrow 3.76}$
	KALE	**91.82**	**89.93**	**94.90**	**75.02**	**88.59**	**77.91**	**81.76**	**86.70**

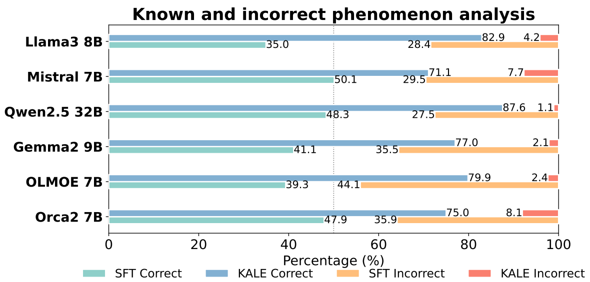

Figure 3: **Known&incorrect phenomenon analysis:** following the known fact checking in Figure 1, we collect cases where LLMs possess the knowledge to answer and analyze the ratios of correct and incorrect answers, denoted as known&correct and known&incorrect.

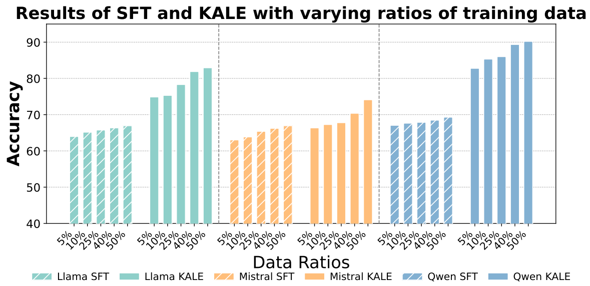

Figure 4: **Ratios of augmented rationales**: by setting the data augmentation ratio from 5% to 50%, we explore the differences between KALE and the SFT under varying data scales. We provide results of LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as the backbones as examples, with more results in Appendix H.9.

yet cannot correctly answer the question, indicating inflexible knowledge manipulation. As shown in Figure 3, we observe that SFT models often exhibit the *known&incorrect* phenomenon. More than 25% of the questions are cases where the LLM possesses the knowledge but cannot provide correct responses. For OLMOE 7B, it reaches 44.1%. In contrast, LLMs trained via KALE demonstrate excellent knowledge manipulation ability, with less than 10% *known&incorrect* issues across all LLMs. Notably, for the Qwen2.5 32B model, this proportion drops to as low as 1.1%. This indicates that KALE effectively enhances knowledge manipulation ability.

Ratios of Augmented Rationales In real-world applications, data acquisition in certain domains can be particularly challenging due to privacy concerns and security restrictions (Rodríguez-Mazahua et al., 2015). Therefore, we investigate KALE and SFT under limited training data scenarios. Taking the AbsR dataset as an example, by setting the training data ratio from 5% to 50%, we present the results in Figure 4. We find that KALE consistently outperforms SFT methods across all levels of augmented rationales. Moreover, this improvement becomes more significant for Qwen2.5 32B, which also demonstrates that KALE is highly effective for more powerful LLMs. **This highlights the significant potential of KALE for low-data, real-world applications.**

6 Conclusion

In this paper, we propose a novel **K**nowledge-**A**ware **LE**arning (KALE) framework to improve the knowledge manipulation ability of LLMs. Specifically, KALE consists of (i) a knowledge-induced data synthesis method to generate high-quality rationales for each Q&A pair through a structured knowledge graph, and (ii) a knowledge-aware fine-tuning paradigm to enhance the knowledge manipulation ability of LLMs. Extensive experiments on eight benchmarks and six open-source models across different scales, ranging from 7B to 32B, demonstrate the superiority of KALE, delivering significant, consistent, and generalizable improvements.[3]

[3]More discussions on KALE can be found in Appendix I.

7 Limitations

We consider a few limitations and future directions. (i) Structured Q&A dataset requirement. Current KALE relies on a structured Q&A dataset to facilitate knowledge-induced data synthesis. For cases where a Q&A dataset is not available, users can consider employing GPT-4o or other LLMs to transform a raw corpus into a structured Q&A format. We think applying KALE directly to raw data is a promising direction. (ii) Hard-match reasoning path generation. When generating reasoning paths, the multi-path A* algorithm is a hard-match approach. Obtaining vectorized embeddings for similarity-based matching is also an optimization direction. (iii) Anchor node selection. In multi-path A*, we empirically sample k anchor nodes for distance estimation. Finer entity-specific selection (e.g., a neural decision module) may yield better results. (iv) KG availability. Current KALE relies on an existing KG, which may constrain its applicability in domains where specialized KGs are scarce. Meanwhile, many areas—including the medical domain—already benefit from community-maintained Wikidata, whose ongoing expansion enhances its value for diverse applications. We are further encouraged by advances in the KG community that target automatic construction of domain-specific KGs: methods like SAC-KG (Chen et al., 2024) show promise in building high-quality KGs. Such approaches are pivotal for extending KALE to domains where mature KGs are not available.

8 Ethical Considerations

This work adheres to the ACL Code of Ethics. Our study does not involve human subjects or personally identifiable information, and we only use publicly available datasets under their respective licenses. We transparently report our methods and potential risks and do not recommend deployment in high-stakes settings without further safety assessments.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2025. Physics of language models: Part 3.2, knowledge manipulation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Anthropic Team. 2024. The Claude 3 model family: Opus, Sonnet, Haiku. Technical report, Anthropic.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Randy L. Buckner and Mark E. Wheeler. 2001. The cognitive neuroscience og remembering. *Nature Reviews Neuroscience*, 2:624–634.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Aaron Chan, Zhiyuan Zeng, Wyatt Lake, Brihi Joshi, Hanjie Chen, and Xiang Ren. 2023. KNIFE: Distilling reasoning knowledge from free-text rationales. *Preprint*, arXiv:2212.09721.

Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. 2024. SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4345–4360, Bangkok, Thailand. Association for Computational Linguistics.

Hanzhu Chen, Xu Shen, Jie Wang, Zehao Wang, Qitan Lv, Junjie He, Rong Wu, Feng Wu, and Jieping Ye. 2025a. Knowledge graph finetuning enhances knowledge manipulation in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. 2025b. Unveiling the key factors for distilling chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15094–15119, Vienna, Austria. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. PaLM: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging NLP cloud APIs. *CoRR*, abs/1812.04718.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. ChatAug: Leveraging chatgpt for text data augmentation. *CoRR*, abs/2302.13007.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *CoRR*, abs/2412.19437.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *CoRR*, abs/2311.04205.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-Verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3563–3578. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 1679–1705. Association for Computational Linguistics.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 177–198. Association for Computational Linguistics.

Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-CARE: a new dataset for exploring explainable causal reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 432–446, Dublin, Ireland. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130.

Luyang Fang, Gyeong-Geon Lee, and Xiaoming Zhai. 2023. Using GPT-4 to augment unbalanced data for automatic scoring. *CoRR*, abs/2310.18365.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628. ACM.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Andrew V. Goldberg and Chris Harrelson. 2005. Computing the shortest path: A search meets graph theory. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 156–165. SIAM.

Google DeepMind Team. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Xinyan Guan, Yanjiang Liu, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2024. Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18126–18134. AAAI Press.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.*, 4(2):100–107.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex 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. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 90–93. ACM.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7B. *ArXiv*, abs/2310.06825.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023b. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9237–9251. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jungo Kasai, Yuhei Kasai, Keisuke Sakaguchi, Yutaro Yamada, and Dragomir Radev. 2023. Evaluating GPT-4 and chatGPT on Japanese medical licensing examinations. *CoRR*, abs/2303.18027.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. 2025. Efficient knowledge injection in llms via self-distillation. *Trans. Mach. Learn. Res.*, 2025.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Yanis Labrak, Adrien Bazoge, Richard Dufour, Beatrice Daille, Pierre-Antoine Gourraud, Emmanuel Morin, and Mickael Rouvier. 2022. FrenchMedMCQA: A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 41–46, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. *CoRR*, abs/2502.12289.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2025. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *Trans. Mach. Learn. Res.*, 2025.

Tianyu Liu, Yun Li, Qitan Lv, Kai Liu, Jianchen Zhu, Winston Hu, and Xiao Sun. 2025. PEARL: parallel speculative decoding with adaptive draft length. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Junyu Luo, Xiao Luo, Kaize Ding, Jingyang Yuan, Zhiping Xiao, and Ming Zhang. 2024. RobustFT: Robust supervised fine-tuning for large language models under noisy response. *CoRR*, abs/2412.14922.

Qitan Lv, Jie Wang, Hanzhu Chen, Bin Li, Yongdong Zhang, and Feng Wu. 2024. Coarse-to-Fine highlighting: Reducing knowledge hallucination in large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yougang Lyu, Lingyong Yan, Shuaiqiang Wang, Haibo Shi, Dawei Yin, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2024. KnowTuning: Knowledge-aware fine-tuning for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14535–14556, Miami, Florida, USA. Association for Computational Linguistics.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. *CoRR*, abs/2210.07128.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *ArXiv*, abs/2311.11045.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Daniel Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2024. Olmoe: Open mixture-of-experts language models. *ArXiv*, abs/2409.02060.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7):3580–3599.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *CoRR*, abs/2402.13963.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training. *arXiv preprint*.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 176 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Lisbeth Rodríguez-Mazahua, Cristian Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes, Jair Cervantes, Jorge Luis García-Alcaráz, and Giner Alor-Hernández. 2015. A general perspective of big data: applications, tools, challenges and trends. *The Journal of Supercomputing*, 72:3073 – 3113.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, 15(2).

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Sai Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung yeung Shum, and Jian Guo. 2023. Think-on-Graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Llama Team. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

David Vilares and Carlos Gómez-Rodríguez. 2019. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy. Association for Computational Linguistics.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2024. Boosting language models reasoning with chain-of-Knowledge prompting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4958–4981. Association for Computational Linguistics.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023a. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Shiming Xie, Hong Chen, Fred Yu, Zeye Sun, and Xiuyu Wu. 2024. Minor SFT loss for LLM fine-tune to increase performance and reduce model deviation. *CoRR*, abs/2408.10642.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590, Singapore. Association for Computational Linguistics.

Kai Xiong, Xiao Ding, Ting Liu, Bing Qin, Dongliang Xu, Qing Yang, Hongtao Liu, and Yixin Cao. 2024. Meaningful learning: Enhancing abstract reasoning in large language models via generic fact guidance. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Nakul Yadav, Chelsea Noble, James E. Niemeyer, Andrea Terceros, Jonathan Victor, Conor Liston, and Priya Rajasethupathy. 2022. Prefrontal feature representations drive memory recall. *Nature*, 608:153 – 160.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024a. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024b. Self-Distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1028–1043. Association for Computational Linguistics.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woo-Myoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2225–2239. Association for Computational Linguistics.

Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Yongji Wang, and Jian-Guang Lou. 2023. Large language models meet NL2Code: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7443–7464. Association for Computational Linguistics.

Eric Zelikman, YH Wu, Jesse Mu, and Noah D Goodman. 2024. STaR: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, volume 1126.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *CoRR*, abs/2308.10792.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2026. A comprehensive survey on automatic text summarization with exploration of llm-based methods. *Neurocomputing*, 663:131928.

Yifei Zhang, Xintao Wang, Jiaqing Liang, Sirui Xia, Lida Chen, and Yanghua Xiao. 2024. Chain-of-Knowledge: Integrating knowledge reasoning into large language models by learning from knowledge graphs. *CoRR*, abs/2407.00653.

Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024. KG-CoT: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6642–6650. ijcai.org.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Zhaocheng Zhu, Xinyu Yuan, Michael Galkin, Louis-Pascal A. C. Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. 2023. A*net: A scalable path-based reasoning approach for knowledge graphs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A More Related Works

A.1 Large Language Models

The advent of pre-trained language models has fundamentally transformed the landscape of natural language processing (NLP), marking a significant paradigm shift in how language understanding and generation tasks are approached. The pioneering work of the GPT series (Radford et al., 2018; Brown et al., 2020) introduced the concept of unsupervised pre-training followed by task-specific fine-tuning, demonstrating the effectiveness of leveraging large-scale unlabeled text corpora. This approach was further refined by BERT (Devlin et al., 2019), which introduced bidirectional context encoding through the masked language modeling objective, achieving state-of-the-art results across a wide range of NLP benchmarks. Subsequent advancements, such as RoBERTa (Liu et al., 2019), optimized the pre-training process by removing the next sentence prediction objective and training with larger batches and more data, leading to improved performance. Megatron-LM then (Shoeybi et al., 2019) showcased the scalability of these models, leveraging model parallelism to train significantly larger architectures. More recently, the field has witnessed the emergence of LLMs that have pushed the boundaries of what is possible in NLP. Models such as LlaMA3 (Team, 2024; Touvron et al., 2023), GPT (OpenAI, 2023), PaLM (Chowdhery et al., 2023), Gemini (Google DeepMind Team, 2023), Claude3 (Anthropic Team, 2024), and Deepseek V3 (DeepSeek-AI, 2024) have demonstrated remarkable abilities in both few-shot and zero-shot learning scenarios (Madaan et al., 2022). These models, often comprising hundreds of billions of parameters, have been pre-trained on diverse and extensive benchmarks, enabling them to generalize across a wide array of tasks with minimal or no task-specific fine-tuning. The evolution from earlier models like GPT and BERT to the current generation of LLMs underscores the importance of scale and the effectiveness of pre-training on large corpora. These advancements have not only improved performance on traditional NLP tasks but have also enabled new applications and capabilities, such as conversational agents (Chen et al., 2024), code generation (Zan et al., 2023; Liu et al., 2025), and complex reasoning tasks (Lv et al., 2024). The continued development and refinement of these models promise to further enhance their utility and impact across various domains.

A.2 Classic Text Data Augmentation Methods

Data augmentation has long been a foundational research area in natural language processing (NLP), aimed at enhancing the quality and diversity of training data to improve model generalization and performance. Traditional data augmentation techniques have predominantly focused on character-level and word-level modifications. An example is Easy Data Augmentation (EDA) (Wei and Zou, 2019), which employs straightforward yet effective strategies such as random insertion, random swapping, random deletion, and synonym replacement to introduce variability into the benchmark (Belinkov and Bisk, 2018; Coulombe, 2018; Wang et al., 2023c). These methods, while computationally efficient, are often limited in their ability to generate semantically coherent and contextually rich variations, particularly at higher linguistic levels such as sentences or documents.

A.3 Chain-of-X Approaches in LLMs

The ability of LLMs to decompose complex problems into a series of intermediate steps and generate internal reasoning processes, known as Chain-of-Thought (CoT) prompting (Wei et al., 2022b), represents a significant advancement in enhancing their reasoning capabilities. The CoT approach emulates human problem-solving strategies by breaking down intricate problems into smaller, more manageable components. This step-by-step reasoning process allows LLMs to focus on each segment individually, reducing errors and improving logical coherence in their responses (Wang et al., 2023b). Moreover, CoT explicitly encourages models to articulate their thought processes, which not only facilitates debugging and refinement of the model's reasoning but also significantly enhances the interpretability of its outputs. As a result, responses generated through CoT are often more accurate, logically consistent, and contextually relevant compared to those produced by models that directly generate final answers without revealing intermediate cognitive steps. The success of CoT has inspired a series of follow-up works that extend its principles to other chain-of-X methods, further broadening its applicability and effectiveness. For instance, chain-of-explanation (Huang et al., 2023) focuses on generating detailed explanations to justify the reasoning process, while chain-of-knowledge (Wang et al., 2024) emphasizes the integration of external knowledge to enrich the model's responses. More recently, chain-of-verification (Dhuliawala et al., 2024) has been proposed to enhance the reliability of LLMs by prompting them to draft initial responses, plan verification questions, answer those questions, and generate a final verified response. This iterative verification process reduces the likelihood of misunderstandings or errors in the model's reasoning. Another notable extension is Chain-of-Knowledge (Zhang et al., 2024), which elicits LLMs to generate explicit pieces of knowledge evidence in the form of structured triples. This approach is inspired by human cognitive behaviors, where individuals often draw mind maps or knowledge maps as reasoning evidence before addressing complex questions. By structuring knowledge in this way, LLMs can better organize and utilize information, leading to more informed and accurate responses.

Algorithm 1 Pseudo code for Multi-path A*

Input: Start node $\mathbf{e}_q$, target node $\mathbf{e}_a$, maximum number of paths m and maximum search depth d

Initialize priority queue $\mathcal{Q}$ with $(f(\mathbf{e}), g(\mathbf{e}), \mathbf{e}, p^i_{\mathbf{e}})$
Initialize reasoning path and visited list $\mathcal{P}, \mathcal{V} \leftarrow [], []$
while $\mathcal{Q} \neq \emptyset$ and $|\mathcal{P}| < m$ **do**
    Dequeue the element with the smallest $f(\mathbf{e})$ from $\mathcal{Q}$
    Append $\mathbf{e}$ into $\mathcal{V}$
    if $\mathbf{e} = \mathbf{e}_a$ **then**
        Append $p^i_{\mathbf{e}}$ into $\mathcal{P}$
        continue ▷ Find Reasoning Path
    end if
    if $g(\mathbf{e}) > d$ **then**
        continue ▷ Path exceeds maximum search depth
    end if
    for each neighbor $\mathbf{n}$ of $\mathbf{e}$ **do**
        if $\mathbf{n} \in$ path **then**
            continue ▷ Avoid cycles
        end if
        Obtain $g(\mathbf{n}) \leftarrow g(\mathbf{e}) + 1$
        Compute $f(\mathbf{n})$ and $h(\mathbf{n})$ via Equations (1) and (2)
        Enqueue $(f(\mathbf{n}), g(\mathbf{n}), \mathbf{n}, p^i_{\mathbf{e}} + [\mathbf{n}])$ into $\mathcal{Q}$
    end for
end while

Output: Reasoning path list $\mathcal{P}$

B More Cases of the *Known&incorrect* Phenomenon

In Figure 1, we present a comparative analysis of the *known&incorrect* phenomenon of models fine-tuned after SFT and KALE, using Mistral-7B as the backbone model. In this section, we further extend the investigation by providing more *known&incorrect* phenomenon comparisons **across LlaMA3 8B, Qwen2.5 32B, Gemma2 9B, OLMOE 7B, and Orca2 7B** on various domains to comprehensively demonstrate the efficacy of our proposed KALE. **As illustrated in Figures 5, 6, 7, 8, and 9**, we still find that that models fine-tuned after SFT still exhibit the *known&incorrect* phenomenon, wherein the models cannot properly recall and apply acquired knowledge to answer correctly despite possessing the relevant knowledge. In contrast, LLMs fine-tuned after KALE demonstrate a better ability to effectively manipulate relevant knowledge to generate correct answers. These results also demonstrate that our KALE effectively strengthens LLMs' knowledge manipulation ability.

C Prompt Templates

We list the prompt templates for different tasks to offer more visually intuitive results for each task in Figures 10, 11, 12, and 13. More detailed prompt information for the best performance of each task and dataset can be seen within the code.

The placeholders `Known Fact`, `Question`, `Answer`, `Reasoning Path`, `Options`, `Generic Fact`, and `Rationales` will be filled with the corresponding terms in each example of corresponding benchmarks.

D Examples of generated reasoning paths and rationales

We present extracted reasoning paths alongside the generated rationales for some samples to provide a more intuitive and straightforward understanding of KALE. We select one sample each from the domains of **Science, Medicine, Common Knowledge, Computer Science, Economics, and Art** in Figures 14, 15, 16, 17, 18, and 19. For each sample, we provided the extracted reasoning paths and the generated rationales to support a more comprehensive understanding of KALE.

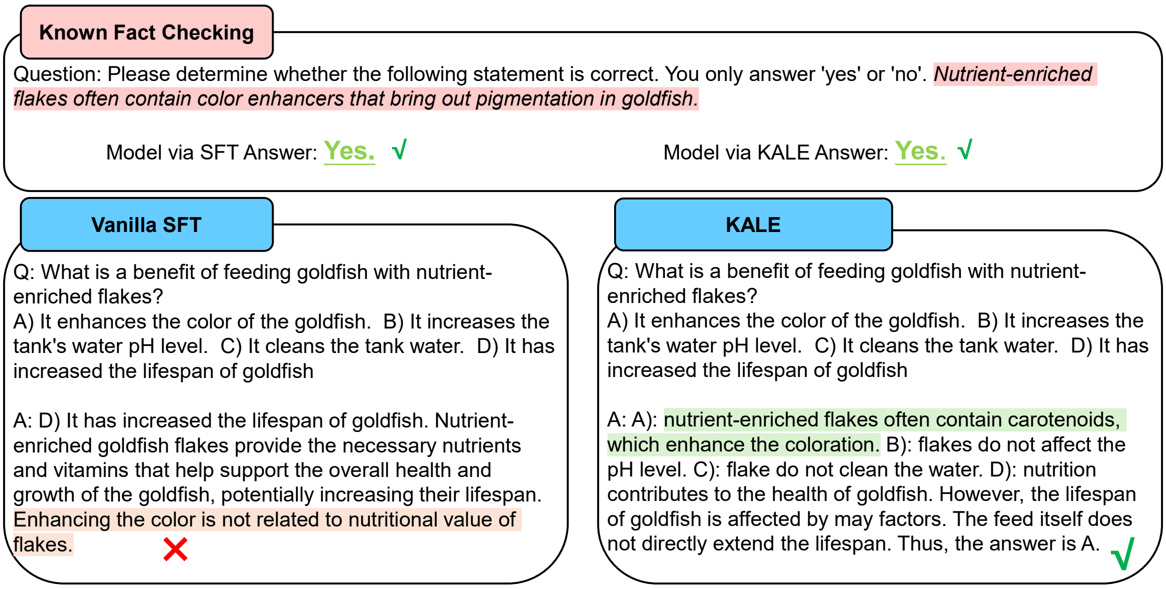

Figure 5: An example of *known&incorrect* phenomenon comparison using LlaMA3 8B as the backbone model.

E Proof of Admissibility of the Proposed Multi-path A* Algorithm

In this section, we show that our proposed heuristic estimated cost in (2) is admissible,i.e., $h(\mathbf{e}) \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$ for any node $\mathbf{e}$, which means that our proposed multi-path A* algorithm can find the best solution. We resort to the triangle inequality property of the distance metric $\mathrm{dist}(x, y)$. For any three nodes A, B, C, the triangle inequality states:

$$\mathrm{dist}(A, C) \leq \mathrm{dist}(A, B) + \mathrm{dist}(B, C) \tag{5}$$

Let us consider an arbitrary landmark α_i from the set $\{\alpha_i\}_{i=1}^{k}$. Applying the triangle inequality with $A = \alpha_i$, $B = \mathbf{e}$, and $C = \mathbf{e}_g$, we have:

$$\mathrm{dist}(\alpha_i, \mathbf{e}_g) \leq \mathrm{dist}(\alpha_i, \mathbf{e}) + \mathrm{dist}(\mathbf{e}, \mathbf{e}_g) \tag{6}$$

Rearranging Equation (6), we obtain:

$$\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e}) \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g) \tag{7}$$

Now, let $X_i = \mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})$. The term in the heuristic function involving α_i is $[X_i]^+ = \max(X_i, 0)$. We consider two cases for the value of X_i:

1. **Case 1:** $X_i < 0$, due to the non-negative property of the distance, we have:
$$[\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})]^+ = 0 \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$$
2. **Case 2:** $X_i \geq 0$, from Eq.(7), we know that $\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e}) \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$. Therefore:
$$[\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})]^+ \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$$

In both cases, for any anchor α_i ($1 \leq i \leq k$), we have shown that:

$$[\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})]^+ \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g) \tag{8}$$

The heuristic function $h(\mathbf{e})$ is defined as the maximum of these terms over all i:

$$h(\mathbf{e}) = \max_{1 \leq i \leq k} \Big[\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})\Big]^+$$

Since each term $[\mathrm{dist}(\alpha_i, \mathbf{e}_g) - \mathrm{dist}(\alpha_i, \mathbf{e})]^+$ is less than or equal to $\mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$, their maximum must also be less than or equal to $\mathrm{dist}(\mathbf{e}, \mathbf{e}_g)$. Thus,

$$h(\mathbf{e}) \leq \mathrm{dist}(\mathbf{e}, \mathbf{e}_g) \tag{9}$$

This inequality holds for any node $\mathbf{e}$. **Therefore, our proposed multi-path A* algorithm is admissible, which means that for any node, our proposed multi-path A* algorithm can find the best solution.**

F Pseudo code of the Proposed Multi-path A*

In Section 4.1, we introduce our multi-path A* algorithm, which efficiently extracts inference paths from question entities to answer entities. Here, we provide the algorithm pseudo code in Algorithm 1.

G More Details of Benchmarks and Experiment Setups

G.1 Implementation Details

In our implementation details, we conduct fine-tuning on all evaluated benchmarks across 3

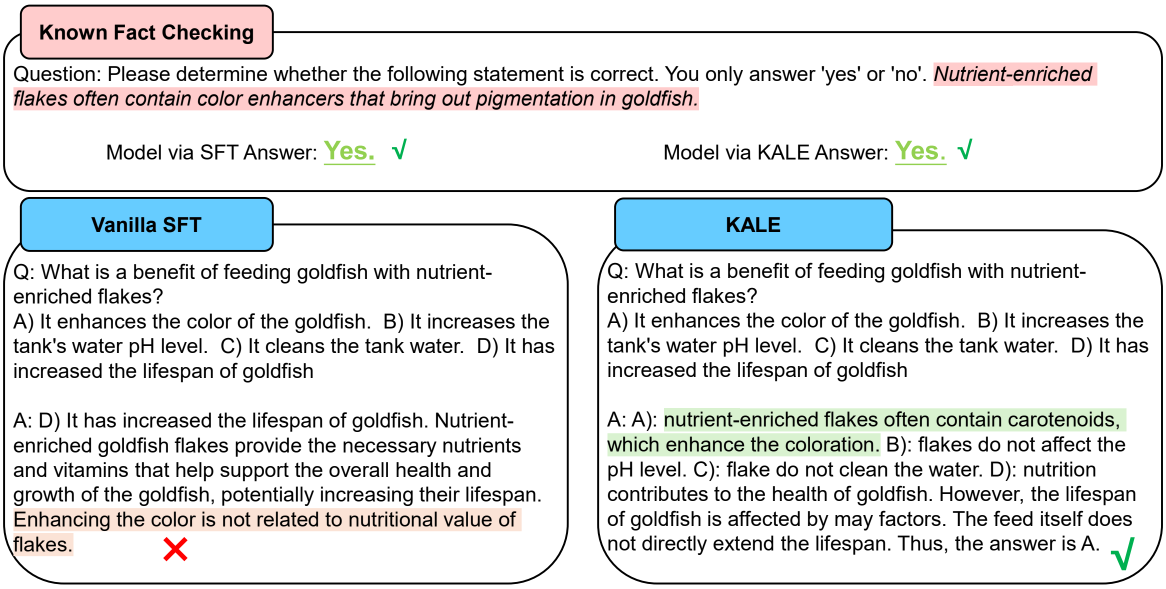

Figure 6: An example of *known&incorrect* phenomenon comparison using Qwen2.5 32B as the backbone model.

epochs with a consistent batch size of 16, utilizing NVIDIA A100 GPUs (80 GB) for computational processing. The computational resources are allocated based on model scale, with 8 GPUs employed for the 7B and 8B parameter models, while the larger 32B parameter models use 16 GPUs to accommodate their increased computational demands during the fine-tuning process. For all answer entities $\mathbf{e}_a$, we choose 10 anchor entities randomly sampled from their 3-hop neighbors. To guarantee stable and reproducible results, we utilize greedy decoding by setting the temperature parameter to 0 in all experiments. The optimization process employs a peak learning rate of 3e-5, implemented in conjunction with a learning rate warmup strategy that gradually increases the learning rate over the initial 1% of training iterations to ensure stable convergence. We set the maximum truncated length as 2048 for all the benchmarks.

In the context of Equation (4) for the KA process, q_θ is fixed and derived from a frozen checkpoint, while p_θ is fine-tuned starting from a separate checkpoint. The KL divergence is computed at the token level using teacher forcing. We apply deepspeed[4] to accelerate the training process. We implement our approach based on PyTorch 2.5.1[5] and Huggingface's Transformers[6]. For the training code of KALE, we modified the training scripts based on LLaMAFactory (Zheng et al., 2024). We are committed to providing the source code of our approach, if accepted. During testing, for all models, we follow MeanLearn (Xiong et al., 2024) to use the same system prompt for a fair comparison: "You are a cautious assistant. You carefully follow instructions. You are helpful and harmless, and you follow ethical guidelines and promote positive behavior. You are given a question together with a few options. You should give an explanation first and then answer the question." More details for the best performance of each task and benchmark can be seen within our code.

[4]https://www.deepspeed.ai/
[5]https://pytorch.org/
[6]https://github.com/huggingface/transformers

G.2 Baseline Methods.

We compare **thirteen** baselines: (i) **Vanilla:** standalone LLMs without modifications. (ii) **CoT** (Wei et al., 2022b): prompting LLMs to generate internal thoughts. (iii) **Think-on-Graph (TOG)** (Sun et al., 2023): applying iterative beam search to enhance LLMs' reasoning ability. (iv) **Struct-GPT** (Jiang et al., 2023b): proposing iterative reading-then-reasoning based on structured data. (v) **GraphRAG** (Edge et al., 2024): integrating KG traversal to retrieve structured relationships. (vi) **SFT** (Wei et al., 2022a): standalone SFT process. (vii) **Self-Distillation Fine-Tuning (SDFT)** (Yang et al., 2024b): guiding fine-tuning with a dataset generated by model itself. (viii) **Dual-stage Mixed Fine-tuning (DMT)** (Dong et al., 2024): achieving a balance between general and specialized ability. (ix) **MeanLearn** (Xiong et al., 2024): teaching LLMs to leverage generic facts. (x) **KG-SFT**

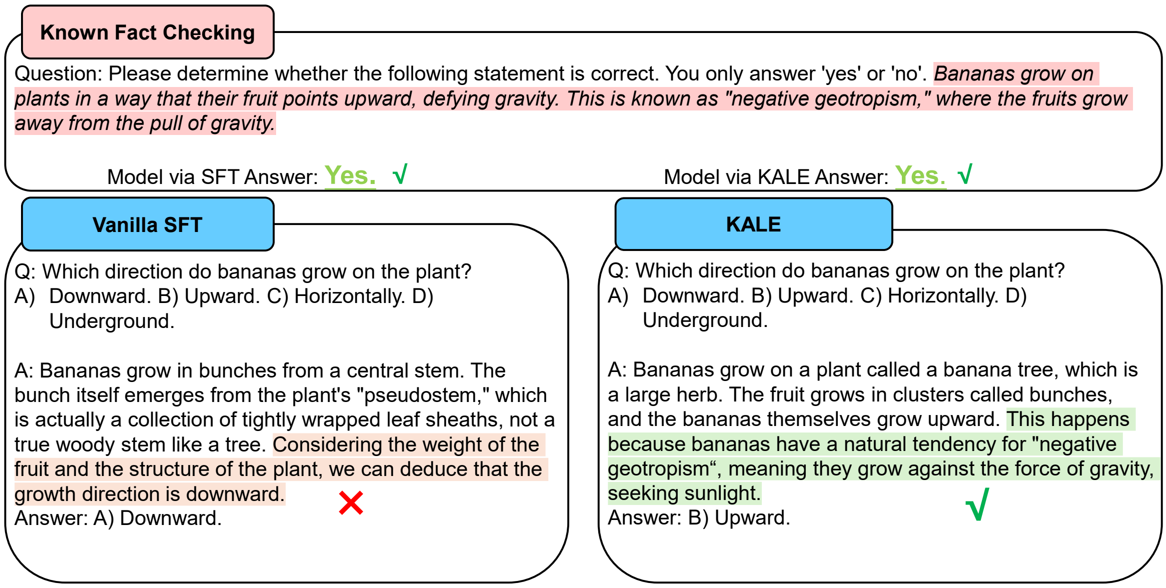

Figure 7: An example of *known&incorrect* phenomenon comparison using Gemma2 9B as the backbone model.

(Chen et al., 2025a): utilizing KGs to filter SFT data to enhance LLMs' ability. (xi) **Self-Taught Reasoner (STaR)** (Zelikman et al., 2024): generating a rationale dataset from a few initials iteratively. (xii) **AugGPT** (Dai et al., 2023): using an LLM to rephrase questions in original data. (xiii) **GPT3Mix** (Yoo et al., 2021): prompting an LLM to generate similar questions in the SFT data.

G.3 Benchmark Details

Table 3: The statistics of AbsR the benchmark.

	Examples	Questions	Generic Facts
Train	18,020	9,010	4,613
Test	844	844	104

For **more details of benchmarks**, we list below all the benchmarks used in **logical reasoning, reading comprehension, and natural language understanding,** respectively, by KALE as follows. **Logical Reasoning Task** we employ AbsR (Xiong et al., 2024), Commonsense (Xiong et al., 2023), and Big Bench Hard (BBH) (Suzgun et al., 2023) as our evaluation benchmarks. Specifically, **the AbsR benchmark** was constructed using GPT-4 as the primary data annotator, following (Chen et al., 2024; Zheng et al., 2023). For each generic fact r_i, GPT-4 was prompted to generate samples $S_i = \{s_1^i, \ldots, s_{m_i}^i \mid 1 \leq m_i \leq 3\}$ in diverse scenarios. Each sample s_j^i consists of a question X_j^i with multiple options, a response Y_j^i containing an answer and an explanation guided by r_i, and forms a triple $s_j^i = \langle X_j^i, r_i, Y_j^i \rangle$. From each sample in the training set s_j^i, two types of examples were derived: (i) K-example, which predicts Y_j^i given $\langle X_j^i, r_i \rangle$, and (ii) R-example, which predicts Y_j^i given only X_j^i. These examples are designed to implicitly enhance abstract reasoning in LLMs through the knowledge and reasoning pathways. In the testing set, only the R-example is provided for each sample. The statistics of the AbsR benchmark are summarized in Table 3.

Table 4: The statistics of the Commonsense.

Dataset	Task Type	Size
α**NLI** (Bhagavatula et al., 2020)	2 Choices	1,507
CSQA (Talmor et al., 2019)	5 Choices	1,221
COPA (Gordon et al., 2012)	2 Choices	500
e-CARE (Du et al., 2022)	2 Choices	2,122
Social IQa (Sap et al., 2019)	3 Choices	1,935
PIQA (Bisk et al., 2020)	2 Choices	1,838
StrategyQA (Geva et al., 2021)	Yes or No	2,290

The Commonsense benchmark (Xiong et al., 2023) is a multiple-choice question-answering benchmark designed to evaluate the ability of LLMs to perform complex reasoning based on commonsense knowledge. Each question in the benchmark is associated with five candidate answers, only one of which is correct. The dataset spans a diverse range of domains, including everyday scenarios, social interactions, and physical phenomena, making it a comprehensive testbed for

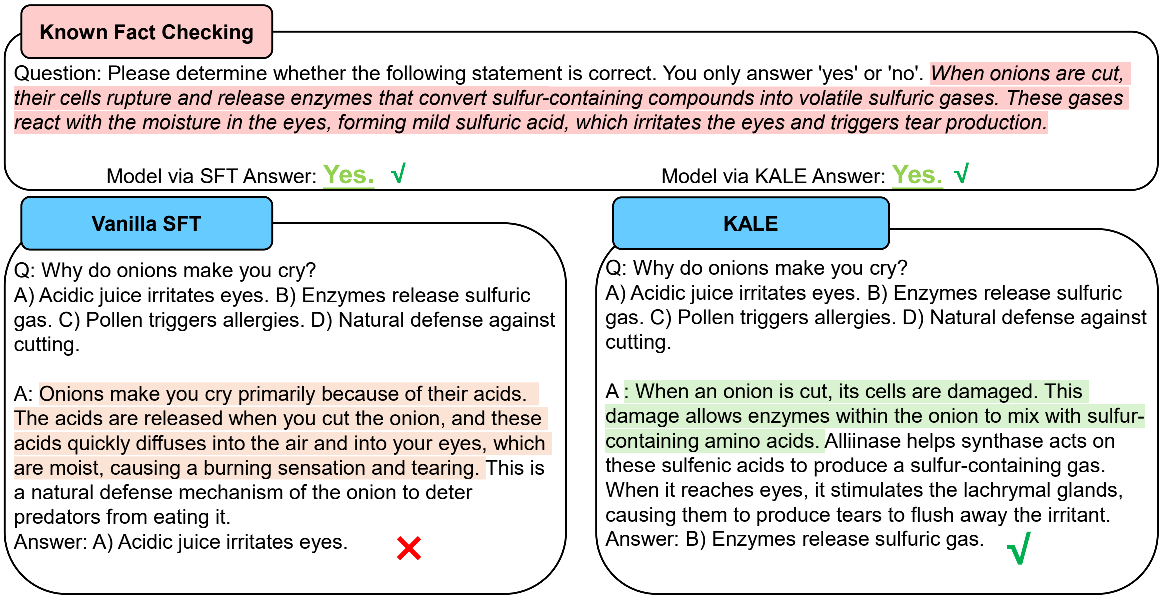

Figure 8: An example of *known&incorrect* phenomenon comparison using OLMOE 7B as the backbone model.

evaluating the commonsense reasoning capabilities of LLMs. We summarize the key statistics and characteristics of Commonsense in Table 4. For the BBH benchmark (Suzgun et al., 2023), it consists of a curated suite of 23 challenging tasks derived from the broader BIG-Bench benchmark (Srivastava et al., 2023). These tasks were specifically selected because prior language model evaluations failed to surpass the average human-rater performance, making them particularly suitable for assessing the limits of current models. The tasks span a wide range of domains, including logical reasoning, mathematical problem-solving, and linguistic understanding, requiring models to demonstrate robust reasoning and contextual comprehension. BBH focuses on the importance of structured reasoning pathways in tackling complex tasks. We summarize the filtering process of BBH in Table 5.

Reading Comprehension Task We employ RACE-M (middle school level reading comprehension task) and RACE-H (high school level reading comprehension task) (Lai et al., 2017) as our benchmarks. RACE is collected from the English exams for middle and high school Chinese students in the age range between 12 to 18. RACE consists of nearly $28,000$ passages and nearly $100,000$ questions generated by human experts (English instructors), and covers a variety of topics that are carefully designed to evaluate the student's ability to understand and reason. The reasoning types of RACE include word matching, paraphrasing, single-sentence reasoning, multi-sentence reasoning, and insufficient/ambiguous. We summarize the details in Table 6.

Natural Language Understanding Task For the natural language understanding task, we employ the **Massive Multitask Language Understanding (MMLU) benchmark** (Hendrycks et al., 2021) and the ARC benchmark for evaluation. **MMLU** is a comprehensive dataset designed to assess the breadth and depth of LLMs' knowledge and problem-solving abilities. MMLU consists of 57 tasks spanning diverse domains, including STEM (Science, Technology, Engineering, and Mathematics), humanities (e.g., law, philosophy, history), social sciences (e.g., economics, sociology, psychology), and other specialized fields (e.g., medicine, finance). The dataset comprises $15,908$ questions, divided into three splits: a dev set with 5 questions per subject for few-shot evaluation, a validation set with $1,540$ questions for hyperparameter tuning, and a test set with $14,079$ questions, ensuring at least 100 test examples per subject.

The questions in MMLU are designed to require extensive world knowledge and expert-level reasoning, making it a rigorous benchmark for evaluating language models' generalization across multiple disciplines. We summarize the key statistics and characteristics of the MMLU dataset in Table 7. The **AI2 Reasoning Challenge (ARC)** benchmark (Bhakthavatsalam et al., 2021) is a comprehensive dataset designed to assess the ability of language

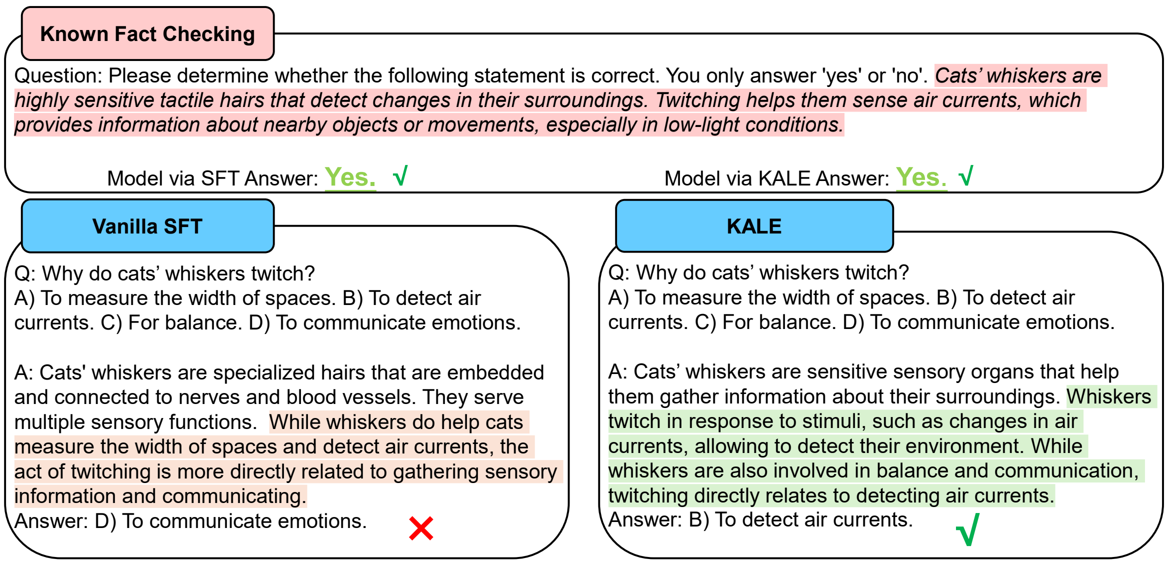

Figure 9: An example of *known&incorrect* phenomenon comparison using Orca2 7B as the backbone model.

Prompt Templates for Known Fact Checking

```
You are a cautious assistant. You carefully follow instructions. You are helpful
and harmless and you follow ethical guidelines and promote positive behavior.
Question: Please determine whether the following statement is correct. You only
answer 'yes' or 'no'. Known Fact.
```

Figure 10: The prompt template used for known fact checking.

models to answer complex, multi-faceted science questions on scientific reasoning and knowledge integration capabilities. The ARC dataset consists of $7,787$ multiple-choice questions derived from grade-school-level science exams, spanning grades 3 through 9. These questions are divided into two subsets: the Easy Set (ARC-E) and the Challenge Set (ARC-C), with the latter containing $2,590$ questions that are particularly difficult and require advanced reasoning skills. The Easy Set (ARC-E) comprises $5,197$ questions that are relatively straightforward and can often be answered using basic retrieval or word co-occurrence methods. In contrast, the Challenge Set (ARC-C) includes questions that were specifically selected because they could not be correctly answered by retrieval-based algorithms (e.g., Information Retrieval Solver) or word co-occurrence methods (e.g., Pointwise Mutual Information Solver). These questions demand deeper comprehension, reasoning, and the integration of distributed knowledge across multiple sentences or concepts. Each question in the ARC dataset is presented with four answer choices, with less than 1% of questions having either three or five options. The dataset is further partitioned into training, validation, and test splits to facilitate model development and evaluation. For instance, the Challenge Set includes $1,119$ training examples, 299 validation examples, and $1,172$ test examples. We summarize the key statistics and characteristics of the ARC dataset in Table 12.

Medical Domain Benchmarks We use multiple-choice medical questions benchmarks in six languages as the representative knowledge-intensive domain, including MedQA (English and Chinese) (Jin et al., 2020), IgakuQA (Japanese) (Kasai et al., 2023), RuMedDaNet (Qiu et al., 2024), FrenchMedMCQA (Labrak et al., 2022), and HeadQA (Vilares and Gómez-Rodríguez, 2019) to provide a comprehensive understanding of our KALE. We provide the statistics of each dataset in Table 8.

H More In-depth Analysis of KALE

H.1 More Results of KALE on Knowledge-intensive Domains

In Tables 1 and 13, we present the performance of KALE across various downstream tasks. To further

Prompt Templates for Rationale Generation

```
You are a cautious assistant. You carefully follow instructions. You are helpful and harmless and you follow ethical guidelines and promote positive behavior. You are given the question: Question. The corresponding answer is: Answer. The reasoning paths are: Reasoning Path. Please provide a detailed explanatory rationale that references these reasoning paths. If you determine that the reasoning path is irrelevant to the current QA pair, you may generate rationales based on your own knowledge.
```

Figure 11: The prompt template used for rationale generation.

Prompt Templates for Main Results

```
You are a cautious assistant. You carefully follow instructions. You are helpful and harmless and you follow ethical guidelines and promote positive behavior. You are given a question together with a few options, you should give an explanation first and then answer the question. Your response should follow the format like Explanation: ___ Answer: ___ Below is the Question and Options: Question Options
```

Figure 12: The prompt template used for main results.

demonstrate the capabilities of KALE, this section provides its evaluation on several knowledge-intensive tasks. Following the same experimental setting of KG-SFT (Chen et al., 2025a), we use MedQA as the benchmark using LlaMA2 7B as the backbone model. As shown in Table 9, we still observe that our proposed KALE significantly outperforms existing state-of-the-art baselines by a large margin, which also demonstrates that our KALE can effectively work under the knowledge-intensive scenarios.

H.2 Inference Time Comparison

As mentioned in Section 2.2, KALE is a post-training method designed to enhance the knowledge manipulation capabilities of LLMs. **Once the model completes training, KALE maintains identical autoregressive inference characteristics to vanilla LLMs during the decoding phase, introducing zero additional temporal overhead and requiring no retrieval operations from external knowledge bases.** We conduct comparative measurements of average inference latency per sample across different methodologies (vanilla LLM, CoT, TOG, StructGPT, GraphRAG, and KALE) using an Nvidia A100 GPU (80GB). The quantitative results in Table 10 reveal that KALE achieves nearly identical inference speed to vanilla LLMs. At the inference stage, both KALE and Vanilla models follow a similar logic: they directly take the instruction and question as input to the LLM. **Therefore, any observed speed differences between them are primarily attributable to slight variations in the length of their generated outputs.** There are instances where the Vanilla model's output length is marginally longer than KALE's, leading to KALE being slightly faster, and vice versa. This minor difference in token generation directly impacts the overall inference time. In contrast, RAG-based approaches requiring knowledge retrieval and CoT methods with extended prompt sequences incur additional computational overhead.

H.3 More results on the Hyperparameter sensitivity evaluation of KALE

Regarding the sensitivity of KALE to the hyperparameters of each component, we conduct experiments to demonstrate its robustness. For all datasets, our default setting involved randomly sampling 10 anchor entities from their 3-hop neighbors. The consistent superior performance of KALE across diverse datasets under these unified parameter settings highlights its general effectiveness.

```
Prompt Templates for reasoning trace quality evaluation

You are a cautious assistant. You carefully follow instructions. You are helpful and harmless and you follow ethical guidelines and promote positive behavior. You are given a rationale for a question. Evaluate the given rationale along five dimensions—Factual Accuracy, Logical Validity, Coherence, Completeness, and Interpretability. For each dimension, output True if the rationale is correct or meets the criterion; otherwise, output False. You should produce a five-element list in the form like [True,True,True,True,True]. Below are the Question Question and the Rationales Rationales.
```

Figure 13: The prompt template used for reasoning trace quality evaluation.

```
Science Domain Example

Question: what is the true color of the Sun?
Answer Choices:
A) Red
B) Yellow
C) White
D) Blue
Extracted Reasoning Paths:
the Sun-emits->full spectrum light-integrates_into->white light
Generated Rationales:
The Sun emits light that contains the entire visible spectrum. When these different colors of light are combined, they create white light.
```

Figure 14: An example of generated rationales on science domain.

Moreover, to further investigate KALE's robustness, we conduct experiments by varying these key hyperparameters. As shown in Table 11, we can observe that KALE exhibits robustness to changes in both the number of anchor entities and hops for neighbors. This further underscores the practical potential and reliability of our KALE framework in real-world applications.

H.4 Average Steps of Extracted Reasoning Paths

By default, we generate three-hop reasoning paths from questions to answers for each question-answer pair. If a 3-hop reasoning path cannot reach the answer entity, we still provide these paths as auxiliary information to facilitate rationale generation by the LLM. We currently provide the proportion of each hop within the generated reasoning paths, where '3hop complete' indicates that the three-hop reasoning path successfully reached the answer, and '3hop partial' indicates that the reasoning path did not reach the answer entity. As shown in Table 12, we find that most of the reasoning paths can directly lead to the final answer entity. Specifically, less than 2% of the reasoning paths cannot reach the answer entity. This suggests that the extracted reasoning paths can effectively elucidate the underlying logic and correlations between the question and the answer.

H.5 More Results of Different Backbone Models

As mentioned in Section 5.2, we select LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as representative models in Table 1. In this section, to further demonstrate the generalization and versatility of KALE, we also conducted experiments on several popular open-source LLMs, including Gemma2 9B, OLMOE-1B-7B, and Orca2 7B. As shown in Table 13, we can still observe that our KALE method significantly outperforms existing baselines on these backbone models as well. This further demon-

Medicine Domain Example

Question: Which of the following is a typical symptom of cancer?
Answer Choices:
A) Weight gain
B) Persistent fever
C) Sore muscles
D) Acne
Extracted Reasoning Paths:
cancer-may cause->decreased resistance-may cause->persistent fever
cancer-may cause->decreased immune function-may cause->persistent fever
Generated Rationales:
The common symptoms that cancer may cause include decreased resistance, which can lead to fever of unknown origin, usually manifested as persistent fever.

Figure 15: An example of generated rationales on medicine domain.

Common Knowledge Domain Example

Question: What do people use to absorb extra ink from a fountain pen?
Answer Choices:
A) shirt pocket
B) calligrapher's hand
C) desk drawer
D) blotter
Extracted Reasoning Paths:
extra ink-absorbed by->absorbent paper-also is->blotter
fountain pen-produces->excess ink- absorbed by->blotter
Generated Rationales:
A blotter is a piece of special absorbent paper. People use it to press against fresh ink from a fountain pen to soak up any excess, which prevents smudging and helps the ink dry faster.

Figure 16: An example of generated rationales on common knowledge domain.

strates the effectiveness of our KALE approach. We also present radar charts for each backbone model to provide a more intuitive performance comparison in Figures 20 and 21. The effectiveness of our KALE across various popular open-source models further demonstrates its strong versatility and generalization capabilities.

H.6 More Results of Applying Different KGs to Extract Rationales

In the main experiments, we used Wikidata as the default KG for extracting reasoning paths. To further evaluate the robustness of KALE under different, smaller-scale KGs, we additionally extracted reasoning paths from alternative KGs and generated corresponding rationales. Specifically, we employed DBpedia (Auer et al., 2007) and ConceptNet (Speer et al., 2017) to extract reasoning paths, based on which we generated rationales for training. We used LLaMA3-8B as the backbone model.

As shown in Table 14, We observe that our KALE model exhibits relatively robust performance across different KGs. This implies a strong potential for KALE to generalize to various KGs in complex real-world datasets, thereby demonstrating its significant applicability in practical scenarios.

Computer Science Domain Example

Question: Which protocol secures data for websites (padlock in the address bar)?
Answer Choices:
A) HTTP
B) FTP
C) HTTPS
D) SMTP
Extracted Reasoning Paths:
websites-secure transport->TLS-implemented as->HTTPS
websites-handle->sensitive data-requires->encryption-provided by->HTTPS
Generated Rationales:
Modern websites handle sensitive user data that requires encryption to prevent interception. HTTPS (Hypertext Transfer Protocol Secure) is the solution; it is essentially the HTTP protocol layered on top of a secure encryption protocol, TLS (Transport Layer Security).

Figure 17: An example of generated rationales on computer science domain.

Table 5: Filtering criteria to create the BIG-Bench Hard (BBH) benchmark.

# Tasks	Criteria
209	All BIG-Bench tasks
187	- After filtering out tasks with more than three subtasks
130	- After filtering out tasks with fewer than 103 examples (3 for few-shot, 100 for evaluation)
85	- After filtering out tasks without human-rater baselines
78	- After filtering out tasks that do not use multiple-choice or exact match as the evaluation metric
78	Clean multiple-choice or exact match tasks
36	- After filtering out tasks in which the best reported model beats average reported human-rater score
23	- After filtering out extremely difficult tasks that are outside the scope of this work
23	**Remaining tasks = BIG-Bench Hard (BBH)**

H.7 More Results of Ablation Study

In Section 5.3, we report the results of the ablation study using LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as the backbone model. In this section, we will further present the results using Gemma2 9B, OLMOE 7B, and Orca2 7B as backbone models to obtain more insights into the individual components constituting KALE across various backbone models. As illustrated in Tables 15, we still observe that the absence of each component within KALE leads to a decline in performance across diverse domains for almost all applied backbone models in all tested benchmarks, which further demonstrates that KALE organically integrates the knowledge-induced data synthesis method and knowledge-aware fine-tuning into a unified framework as well. We still observe that the absence of knowledge-aware fine-tuning (**KALE**$_{\text{w/o KA}}$) leads to a more significant decline in accuracy, which further demonstrates the importance of effectively implicit knowledge learning.

H.8 More Results of KNOWN&INCORRECT Phenomenon on Different Baselines

As mentioned in Section 5.4, models fine-tuned using vanilla SFT still exhibit a serious known-incorrect phenomenon. In this section, we provide more analysis of the known-incorrect phenomenon to include additional baselines involving the training of LLMs. As shown in Table 16, we observe that our KALE consistently achieves the best results in knowledge manipulation analysis. If the model possesses relevant knowledge, KALE exhibits the lowest *known&incorrect* rate. Specifi-

Economics Domain Example

Question: If demand increases while supply remains constant, what happens to the equilibrium price?
Answer Choices:
A) Lower equilibrium price
B) Stays the same
C) Higher equilibrium price
D) Becomes zero
Extracted Reasoning Paths:
demand–shifts right->demand curve–causes->higher equilibrium price
Generated Rationales:
A rightward shift in the demand curve, with supply held constant, leads to a higher equilibrium price. This occurs because at the original price, a shortage is created, causing buyers to compete and bid the price upward to a new equilibrium.

Figure 18: An example of generated rationales on economics domain.

Table 6: Statistics of the reading comprehension benchmarks, RACE-H and RACE-M. The values below the Training/Valid/Testing Set are the number of passages and questions in each dataset, respectively. Passage/Question/Option Len denotes the average length of the passages, questions, and options, respectively. Vocab size denotes the number of words in the vocabulary.

Dataset	Training Set	Valid Set	Testing Set	Passage Len	Question Len	Option Len	Vocab Size
RACE-M	6,409/25,421	368/1,436	362/1,436	231.1	9.0	3.9	32,811
RACE-H	18,728/62,445	1,021/3,451	1,045/3,498	353.1	10.4	5.8	125,120

Table 7: Statistics for MMLU, ARC-C, and ARC-e datasets.

Statistics	**Train**	**Dev**	**Test**
MMLU	99,842	1,540	14,079
ARC-C	1,119	299	1,172
ARC-e	2,251	570	2,376

cally, on Qwen-2.5 32B, KALE demonstrates only a 1.07% *known&incorrect* rate. This further indicates that KALE effectively enhances LLMs' knowledge manipulation ability in downstream tasks.

H.9 More Results of SFT and KALE with Varying Ratios of Training Data

In Section 5.4, we utilized LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as backbone models to investigate the performance of models trained with the SFT and KALE methods on downstream tasks under varying ratios of training rationales.

In this section, we provide additional results using other LLMs as backbone models, including Gemma2 9B, OLMOE 7B, and Orca2 7B. As shown in Figure 22, we observed that KALE demonstrated superior performance on downstream tasks even when only a small proportion of rationales was used for training. Specifically, the improvement of the OLMOE model can reach over 20% on low-data scenarios. These findings highlight the effectiveness of KALE in low-resource scenarios, which also impies a great potentials of our KALE for scenarios with limited high-quality data.

H.10 More Results of Rationales Generated by Different LLMs

In our main experiments, we utilize GPT-4o to generate rationales for each sample. We choose GPT-4o due to its exceptional performance in generating high-quality rationales, as it has demonstrated impressive results on numerous understanding and reasoning tasks. To demonstrate the generalizability of our KALE, we also incorporate two popular open-source LLMs—i.e., DeepSeek V3

Table 8: Statistical results for medical multiple-choice questions benchmarks in six languages.

Dataset	**Language**	**Source**	**Train**	**Test**
MedQA	English	United States Medical Licensing Examination	10178	1273
MedQA	Chinese	United States Medical Licensing Examination	27400	3426
IgakuQA	Japanese	Japan's medical licensure exams (2018-2022)	1590	199
RuMedDaNet	Russian	Russian medical judgment question dataset	1052	256
FrenchMedMCQA	French	Professional exams for the French Pharmacy degree	2171	622
Head-QA	Spanish	Exams for positions in the Spanish healthcare	2657	2742

Table 9: Experiment results for existing methods on knowledge-intensive domains. The results of the mentioned methods are taken from KG-SFT (Chen et al., 2025a). We **bold** the best results for each dataset.

Method	MedQA (English)	MedQA (Chinese)	IgakuQA (Russian)	RuMedDaNet (Spanish)	MedMCQA (French)	HeadQA (Japanese)	Average
Vanilla	28.20	28.37	51.17	32.97	12.76	11.10	27.43
COT	37.65	39.01	65.23	40.33	25.08	23.63	38.48
TOG	34.27	28.13	48.42	35.59	12.47	19.61	29.75
KGR	33.15	26.88	47.52	34.74	13.39	17.29	28.83
KAPING	36.39	27.24	54.66	34.98	11.54	15.91	30.45
SFT	33.62	29.33	66.40	35.19	12.67	21.11	32.30
AugGPT	40.29	36.54	62.14	40.70	22.99	27.13	38.30
GPT3Mix	39.35	37.97	66.01	41.50	25.08	26.13	39.34
KG-SFT	41.71	39.31	68.75	44.40	28.45	28.14	41.79
KALE (ours)	**45.89**	**42.77**	**69.81**	**45.58**	**30.39**	**28.79**	**43.53**

Table 10: Average testing time for each sample on the AbsR dataset for each method (Unit: second)

Backbone Models	Vanilla	CoT	TOG	StructGPT	GraphRAG	KALE (ours)
LlaMa3 8B	7.44	7.91	8.21	7.88	9.08	7.50
Mistral 7B	2.19	3.11	4.97	5.45	10.10	2.11
Qwen2.5 32B	11.20	11.90	11.8	12.8	12.30	11.09
Gemma2 9B	3.73	4.19	4.82	3.98	8.40	3.93
OLMOE 7B	8.33	8.75	10.70	14.60	11.04	8.55
Orca2 7B	3.97	4.33	4.95	7.09	8.20	3.67

Table 11: Hyperparameter sensitivity evaluation on the number of anchor entities and the hop of neighbors.

Anchors	Hops	Absr	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
5	2	82.94	80.03	84.18	66.34	61.79	**57.98**	68.27	73.33
5	4	84.50	78.50	84.13	63.64	61.33	57.82	66.60	71.73
15	2	82.94	75.09	85.00	64.95	60.03	56.13	65.75	69.64
15	4	**85.31**	76.19	**87.40**	65.44	60.68	54.45	65.41	71.03
10 (ori)	3(ori)	83.62	**81.23**	86.45	**65.69**	**63.27**	57.33	**68.61**	**74.12**

Art Domain Example

Question: The technique of dramatic light-dark contrast in painting is called:
Answer Choices:
A) Impasto
B) Fresco
C) Chiaroscuro
D) Sfumato
Extracted Reasoning Paths:
painting-contrast of->light and dark-technique named->chiaroscuro
painting-modeling of form->using dramatic light-a key feature of->chiaroscuro
Generated Rationales:
Chiaroscuro is the technique in painting that uses strong, dramatic contrasts between light and dark. Artists employ this method not only to create a sense of volume for modeling three-dimensional subjects, but also to produce a powerful, theatrical mood.

Figure 19: An example of generated rationales on art domain.

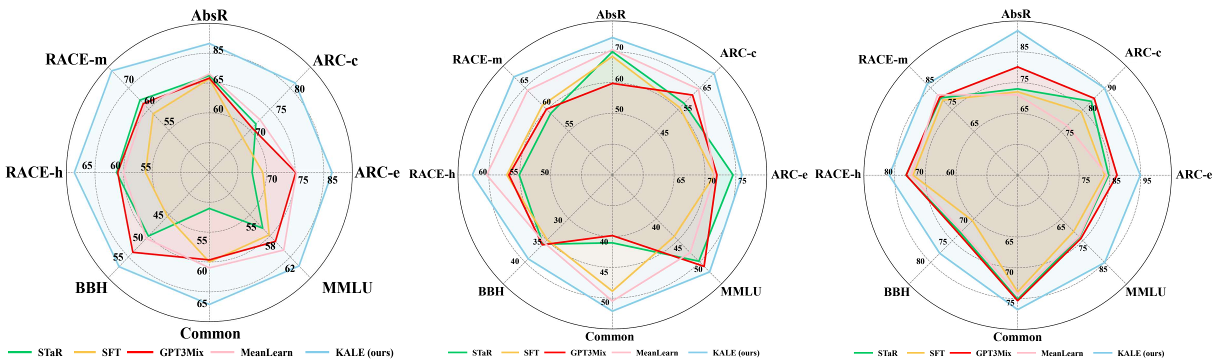

Figure 20: KALE achieves state-of-the-art performance on a broad range of scientific optimization tasks compared with existing methods, using LlaMA3 8B, Mistral 7B, and Qwen2.5 32B as backbone models, respectively.

Table 12: Statistics of average step in reasoning path on the AbsR dataset.

1-hop	2-hop	3-hop complete	3-hop partial
15.76	54.03	28.27	1.94

and LLaMA3.1-70B-Instruct—for rationale generation and apply LLaMA3 8B as the backbone model. The results in Table 18 indicate that training on rationales generated by LLaMA3 70B and DeepSeekV3 still yields performance that significantly surpasses vanilla methods and achieves results comparable to those derived from GPT-4.0-generated rationales. **This demonstrates that KALE is relatively robust to rationales generated by different LLMs, highlighting its effectiveness for practical applications.**

H.11 More Results of Different Types of Generated Rationales

To further investigate whether the model genuinely benefits from meaningful knowledge or merely from the presence of any rationale, we generate two sets of modified rationales based on the original reasoning paths:

- KALE$_{irrelated}$: We instruct GPT-4o to generate factually irrelevant rationales to the reasoning paths.
- KALE$_{contrast}$: We instruct GPT-4o to generate rationales that are factually contrasting to the reasoning paths.

We denote our original method as KALE$_{ori}$ and present the comparative results in Table 19. The performance obtained using irrelevant or contrasting rationales is significantly lower than that of

Table 13: More results of our KALE using Gemma2 9B, OLMOE 7B, and Orca2 7B as backbone models. We **bold** the best results and underline the suboptimal results for each backbone model.

Backbone	Category	Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Gemma2 9B	Prompt-based	Vanilla	52.49	79.95	88.89	57.66	53.56	48.93	73.13	78.62
		CoT	67.54	81.06	86.91	61.43	57.35	53.37	71.07	79.32
	Retrieval-based	TOG	72.04	79.27	81.65	63.06	59.31	51.53	75.99	79.53
		StructGPT	59.24	83.28	86.87	59.30	61.40	57.98	79.87	81.27
		GraphRAG	64.57	85.07	84.13	65.00	62.53	60.89	76.80	81.69
	SFT-based	SFT	61.37	81.06	89.06	58.97	55.26	51.38	74.93	80.78
		SDFT	75.83	82.42	90.91	60.85	57.67	55.37	74.19	81.20
		DMT	77.13	81.83	91.12	62.00	56.69	53.22	76.99	83.01
		MeanLearn	72.04	80.20	89.90	63.06	58.98	57.36	75.53	81.20
		KG-SFT	74.76	80.20	88.26	64.54	59.37	55.06	77.16	82.73
	Augmented-based	STaR	76.66	77.22	84.43	60.20	54.54	56.60	75.53	81.55
		AugGPT	59.60	82.34	81.86	53.73	55.26	58.89	78.88	83.33
		GPT3Mix	59.72	75.43	88.22	64.29	61.01	55.83	79.07	83.98
		KALE (ours)	**81.52**	**88.57**	**94.70**	**68.63**	**65.32**	**65.49**	**83.30**	**87.74**
OLMOE 7B	Prompt-based	Vanilla	49.88	62.03	65.99	44.06	38.73	35.73	57.18	65.74
		CoT	51.06	63.13	67.34	45.62	39.91	36.66	59.46	64.83
	Retrieval-based	TOG	54.50	64.42	69.82	47.26	40.89	38.34	60.35	67.75
		StructGPT	56.87	65.70	71.12	51.60	41.61	40.64	60.03	69.63
		GraphRAG	57.82	60.75	71.25	50.61	43.50	41.56	60.66	68.04
	SFT-based	SFT	53.31	63.91	68.52	49.14	40.43	37.58	59.18	69.63
		SDFT	59.95	65.52	70.16	50.61	42.78	38.65	59.06	67.84
		DMT	60.43	66.04	70.83	51.26	42.36	39.57	61.18	68.45
		MeanLearn	71.09	66.30	67.80	54.55	44.21	43.10	60.03	72.42
		KG-SFT	61.26	66.41	70.58	52.66	43.17	38.04	61.09	65.25
	Augmented-based	STaR	59.24	66.12	71.04	50.36	43.76	41.41	62.84	66.04
		AugGPT	61.73	66.55	71.54	52.00	43.76	43.40	60.98	70.19
		GPT3Mix	62.20	67.06	72.60	53.23	43.50	42.02	60.26	75.48
		KALE (ours)	**81.99**	**72.78**	**74.60**	**58.25**	**46.96**	**45.88**	**64.35**	**75.84**
Orca2 7B	Prompt-based	Vanilla	61.37	68.34	70.75	47.67	44.09	37.27	72.36	75.49
		CoT	67.77	70.90	77.40	50.86	43.77	39.20	72.58	75.84
	Retrieval-based	TOG	59.60	73.89	75.72	62.24	51.14	42.94	73.41	74.09
		StructGPT	65.17	67.66	77.95	53.40	45.40	46.01	76.01	78.41
		GraphRAG	67.06	69.97	78.87	54.71	50.75	47.70	76.02	75.77
	SFT-based	SFT	63.98	71.33	76.56	48.24	52.90	47.70	73.33	76.88
		SDFT	76.66	72.53	75.72	52.33	52.25	46.63	73.99	75.14
		DMT	75.24	73.55	77.15	51.27	52.63	48.31	73.41	77.30
		MeanLearn	77.01	77.22	86.57	66.50	53.04	35.58	73.36	78.76
		KG-SFT	78.91	72.44	78.87	52.42	54.00	48.93	74.01	76.90
	Augmented-based	STaR	71.68	75.00	81.57	64.53	45.85	44.33	75.33	77.30
		AugGPT	61.73	73.89	80.05	53.89	47.81	44.32	75.24	78.55
		GPT3Mix	69.79	74.58	79.67	54.46	50.75	45.25	75.53	77.51
		KALE (ours)	**83.41**	**78.16**	**88.51**	**69.62**	**61.20**	**50.77**	**78.62**	**80.02**

KALE_{ori}. This demonstrates the effectiveness of our knowledge-induced data synthesis, confirming that the model truly benefits from high-quality, factually accurate, and logically coherent rationales.

H.12 More Results on Reasoning Trace Quality Evaluation of Generated Rationales

We incorporate a reasoning trace quality metric to evaluate the quality of the generated rationales to provide more insight into our KALE. We assess rationale quality across five critical dimensions: **Factual Accuracy, Logical Validity, Coherence, Completeness, and Interpretability** (Lee and Hockenmaier, 2025). Each dimension is evaluated as a binary classification task. Following the "LLM-as-a-judge" paradigm (Zheng et al., 2023), we utilize GPT-5 for this assessment. As shown in Table 20, we use the AbsR dataset as an example and find that rationales generated via our KALE exhibit strong performance across all five dimensions, which further validates the effectiveness of KALE.

Table 14: Results comparison of KALE using different KGs to extract reasoning path using Llama3 8B as the backbone model.

	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
KALE$_{DBpedia}$	80.81	77.89	83.77	60.07	61.28	**58.00**	65.58	68.73
KALE$_{ConceptNet}$	79.93	**81.54**	84.19	62.03	61.06	55.94	66.93	71.17
KALE$_{Wikidata}$	**83.62**	81.23	**86.45**	**65.69**	**63.27**	57.33	**68.61**	**74.21**

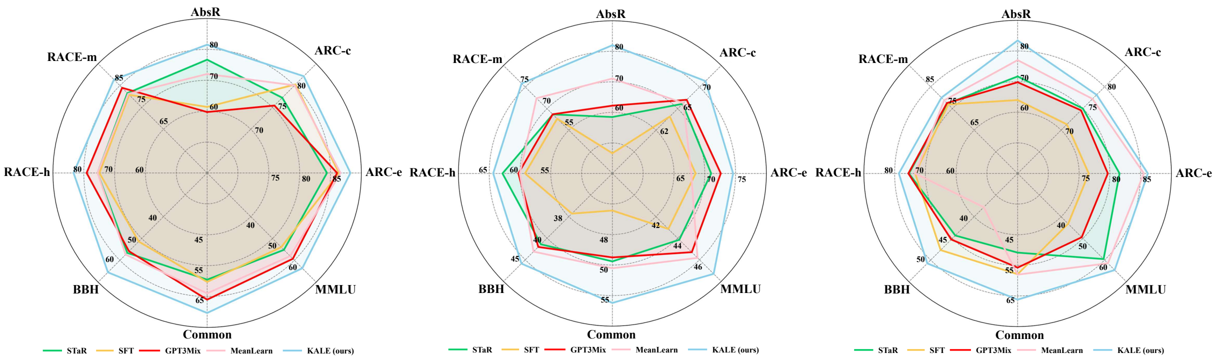

Figure 21: KALE achieves state-of-the-art performance on a broad range of scientific optimization tasks compared with existing methods, using Gemma2 9B, OLMOE 7B, and Orca2 7B as backbone models, respectively.

Table 15: More results of the ablation study of our KALE, using Gemma2 9B, OLMOE 7B, and Orca2 7B as the backbone models.

Backbone	Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Gemma2 9B	**KALE**$_{\text{w/o KI}}$	76.54$_{\downarrow 4.98}$	84.47$_{\downarrow 4.10}$	92.17$_{\downarrow 2.53}$	65.52$_{\downarrow 3.11}$	61.14$_{\downarrow 4.18}$	61.35$_{\downarrow 4.14}$	80.02$_{\downarrow 3.28}$	84.26$_{\downarrow 3.48}$
	KALE$_{\text{w/o KA}}$	73.22$_{\downarrow 8.30}$	78.41$_{\downarrow 10.16}$	90.32$_{\downarrow 4.38}$	66.99$_{\downarrow 1.64}$	63.42$_{\downarrow 1.90}$	60.12$_{\downarrow 5.37}$	78.70$_{\downarrow 4.60}$	82.66$_{\downarrow 5.08}$
	KALE	**81.52**	**88.57**	**94.70**	**68.63**	**65.32**	**65.49**	**83.30**	**87.74**
OLMOE 7B	**KALE**$_{\text{w/o KI}}$	78.91$_{\downarrow 3.08}$	69.80$_{\downarrow 2.98}$	73.23$_{\downarrow 1.37}$	56.51$_{\downarrow 1.74}$	40.89$_{\downarrow 6.07}$	43.25$_{\downarrow 2.63}$	62.92$_{\downarrow 1.43}$	70.26$_{\downarrow 5.58}$
	KALE$_{\text{w/o KA}}$	74.17$_{\downarrow 7.82}$	68.26$_{\downarrow 4.52}$	70.92$_{\downarrow 3.68}$	55.28$_{\downarrow 2.97}$	44.35$_{\downarrow 2.61}$	42.48$_{\downarrow 3.40}$	60.26$_{\downarrow 4.09}$	69.22$_{\downarrow 6.62}$
	KALE	**81.99**	**72.78**	**74.60**	**58.25**	**46.96**	**45.88**	**64.35**	**75.84**
Orca2 7B	**KALE**$_{\text{w/o KI}}$	79.68$_{\downarrow 3.73}$	76.37$_{\downarrow 1.79}$	84.18$_{\downarrow 4.33}$	67.81$_{\downarrow 1.81}$	58.59$_{\downarrow 2.61}$	48.31$_{\downarrow 2.46}$	74.96$_{\downarrow 3.66}$	77.99$_{\downarrow 2.03}$
	KALE$_{\text{w/o KA}}$	77.61$_{\downarrow 5.80}$	75.43$_{\downarrow 2.73}$	82.49$_{\downarrow 6.02}$	65.52$_{\downarrow 4.10}$	54.41$_{\downarrow 6.79}$	45.86$_{\downarrow 4.91}$	73.16$_{\downarrow 5.46}$	75.91$_{\downarrow 4.11}$
	KALE	**83.41**	**78.16**	**88.51**	**69.62**	**61.20**	**50.77**	**78.62**	**80.02**

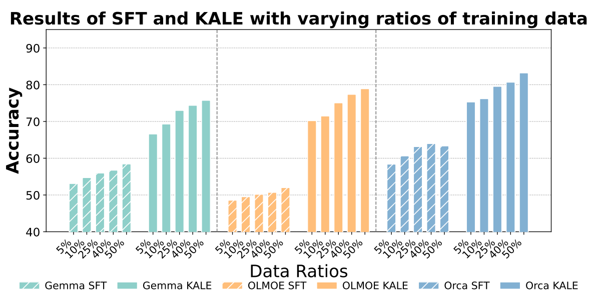

Figure 22: Results of different ratios of augmented rationales on SFT and KALE on Gemma2 9B, OLMOE 7B, and Orca2 7B, respectively.

H.13 More Results of Combining SFT with KALE

We also conduct an additional experiment using Llama3 8B as the backbone model. We compare two approaches: our original KALE method (denoted as KALE$_{ori}$) and a sequential approach where the model is first fine-tuned with SFT and then further trained with KALE (denoted as KALE$_{joint}$)

As shown in Table 21, we find that while combining SFT first with KALE (KALE$_{joint}$) yields improvements only on some datasets. This presents a promising avenue for future work to thoroughly explore the optimal integration of KALE with existing post-training methods to achieve more consistent and significant performance enhancements for specific downstream domains.

Table 16: Experiment results on the AbsR benchmark in six LLM backbones range for the knowledge manipulation analysis. We **bold** the best results for each method.

Category	Method	LlaMA3 8B	OLMOE 7B	Qwen2.5 32B	Gemma2 9B	Mistral 7B	Orca2 7B
Known&Correct	SFT	34.95	39.33	48.34	41.11	50.12	47.88
	SDFT	56.28	47.87	53.00	58.89	54.03	55.57
	DMT	56.64	44.91	65.17	61.85	51.78	57.35
	Meanlearn	48.43	59.12	60.19	48.93	56.52	59.12
	KG-SFT	59.83	50.36	67.06	55.33	59.36	62.90
	STaR	48.93	45.97	58.41	51.09	56.75	59.60
	AugGPT	47.27	45.97	65.76	43.48	43.01	48.93
	GPT3Mix	54.15	48.34	62.90	42.30	43.84	56.52
	KALE	**82.94**	**79.86**	**87.56**	**77.01**	**71.09**	**75.00**
Known&Incorrect	SFT	28.43	44.08	27.49	35.55	29.50	35.90
	SDFT	19.87	12.08	20.73	16.79	19.79	21.09
	DMT	17.93	15.52	10.07	15.28	21.44	17.89
	Meanlearn	22.75	11.97	10.90	23.11	14.45	17.89
	KG-SFT	18.37	10.90	11.85	19.43	13.03	16.00
	STaR	21.02	13.27	14.58	24.76	13.27	12.08
	AugGPT	17.18	15.76	13.15	16.12	22.27	12.80
	GPT3Mix	14.12	13.86	17.20	17.42	15.88	13.27
	KALE	**4.15**	**2.37**	**1.07**	**2.13**	**7.7**	**8.06**

Table 17: Experiment results on the AbsR benchmark in six LLM backbones range from different data ratios. We **bold** the better results for each backbone model.

	LlaMA3 8B		Mistral 7B		Qwen2.5 32B		Gemma2 9B		OLMOE-1B-7B		Orca2 7B	
% Data	SFT	KALE	SFT	KALE	SFT	KALE	SFT	KALE	SFT	KALE	SFT	KALE
5%	63.98	**74.88**	63.03	**66.35**	67.06	**82.78**	53.12	**66.60**	48.58	**70.22**	58.39	**75.31**
10%	65.17	**75.32**	63.86	**67.28**	67.65	**85.53**	54.73	**69.31**	49.53	**71.49**	60.63	**76.20**
25%	65.76	**78.31**	65.40	**67.79**	67.89	**86.01**	55.98	**73.00**	50.19	**75.07**	63.17	**79.55**
40%	66.35	**81.89**	66.23	**70.38**	68.48	**89.36**	56.75	**74.39**	50.71	**77.38**	63.99	**80.69**
50%	66.94	**82.93**	66.94	**74.11**	69.31	**90.22**	58.44	**75.75**	52.01	**78.91**	63.35	**83.21**

H.14 More Results of KALE on Open-ended Generations

Our primary objective in KALE is to improve LLM performance on knowledge manipulation tasks, which are essential for enabling models to reason and respond accurately based on existing factual and procedural knowledge, such as in mathematics and multi-hop QA (Allen-Zhu and Li, 2025). Consequently, our core evaluations primarily focus on QA-style benchmarks that reflect these capabilities.

In this section, **we investigate KALE's general abilities, particularly in open-ended generation, which requires fluency, coherence, and creativity.** To this end, we conduct an additional evaluation on open-ended generation using MT-Bench (Zheng et al., 2023), a benchmark suite that covers a broad range of tasks, including creative and free-form responses. We compare KALE, SFT, and vanilla models instantiated on the Llama 3 8B backbone, and we employ a GPT-5 LLM-Judge for automatic evaluation.

When evaluating with MT-Bench, we perform pairwise comparisons between the KALE model and each of the vanilla and SFT models separately. To mitigate position bias (Zheng et al., 2023), the evaluation protocol for each *(question, Model A, Model B)* instance is defined as follows:

1. We first evaluate using the response order (A, B).

2. We then swap the responses and evaluate again using the order (B, A).

We only count a "win" for a model if both judgments agree on the same winner. If the two judgments yield conflicting outcomes, we record the result as a "tie." We report the corresponding proportions of win rate, loss rate, and tie rate.

The results in Table 22 show that the **KALE-trained model consistently outperforms both the**

Table 18: Results of KALE for rationales generated by different LLMs.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Vanilla	62.68	66.79	69.90	58.72	55.88	46.54	53.35	57.02
KALE_DeepSeek V3	82.70	**81.48**	**86.70**	64.70	62.25	**58.13**	64.69	71.03
KALE_Llama3 70B	78.91	77.56	84.05	63.72	60.03	54.45	65.52	69.78
KALE_GPT-4o (Original)	**83.62**	81.23	86.45	**65.69**	**63.27**	57.33	**68.61**	**74.12**

Table 19: Comparion of KALE on different types of generated rationales.

	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
KALE$_{irrelated}$	65.05	64.76	66.20	56.18	54.21	47.55	50.03	54.11
KALE$_{contrast}$	59.60	59.64	63.56	51.84	52.25	42.48	51.11	48.96
KALE$_{ori}$	**83.62**	**81.23**	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**

vanilla model and the SFT baseline in terms of open-ended generation quality, achieving a higher win rate under the LLM-based evaluation. This indicates that KALE not only preserves but can also enhance the model's general generation abilities. Overall, these findings suggest that KALE does not harm, and may in fact improve, general fluency and creativity.

H.15 More Results of KALE on Thinking Style Models

We also conduct additional experiments evaluating our KALE pipeline on Qwen3-32B (thinking) (Yang et al., 2025): a state-of-the-art model with enhanced multi-step reasoning capabilities. As shown in Table 23, the KALE pipeline continues to provide significant improvements over vanilla and SFT methods when applied to these newer LLMs. This supports our claims regarding the robustness and generalizability of our KALE across different model backbones and reasoning capabilities.

H.16 More Results of KALE on Self-taught Settings

To validate the quality of our generated rationales in a self-taught setting, we conduct an additional experiment using LLaMA3-8B as both the teacher and student, as shown in Table 24. Even under this constrained setting, KALE consistently improves performance, highlighting that the reasoning paths synthesized from KI are of sufficient quality to facilitate learning even without access to a powerful teacher.

H.17 More Results of KALE on Comparison of GRPO

Recently, methods such as GRPO (Shao et al., 2024) represent a compelling alternative by relying solely on questions and ground truth answers, thereby bypassing the need for high-quality intermediate textual reasoning annotations—the central challenge KALE is designed to address.

To investigate this, we have conducted additional experiments using GRPO with LLaMA3-8B as the backbone, implemented via LLaMAFactory (Zheng et al., 2024). The results in Table 25 demonstrate that GRPO, when directly trained on QA samples, still underperforms compared to our proposed KALE pipeline, especially on the ARC-e dataset, which leads to 12.92%. This empirical evidence further supports the effectiveness of KALE in enhancing the reasoning capabilities of LLMs.

Moreover, we note that outcome-based RL approaches often suffer from lower training efficiency. Specifically, RL methods require extensive sample rollouts and typically operate under sparse reward settings. This results in a high computational burden during training. In contrast, the KI component of KALE utilizes an efficient multi-path A* algorithm, while the KA component optimizes a sample-level distribution loss that provides dense supervision signals, leading to significantly more efficient training.

We also note that KALE and RL-based methods are not mutually exclusive. KALE is designed to mitigate the "known and incorrect" phenomenon commonly observed in SFT models, which tend to overfit to explicit input-output mappings and fail to dynamically retrieve and manipulate task-relevant

Table 20: Reasoning trace quality evaluation for rationales on the AbsR dataset via GPT-5.

Factual Accuracy	Logical Validity	Coherence	Completeness	Interpretability
98.82	97.63	99.53	100.00	99.64

Table 21: Comparison results of combining SFT with KALE pipeline using Llama3 8B as the backbone model.

	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
KALE$_{joint}$	80.21	**82.25**	84.18	61.34	60.42	53.22	67.98	72.91
KALE$_{ori}$	**83.62**	81.23	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**

knowledge (Allen-Zhu and Li, 2025). KALE instead enhances the model's knowledge manipulation capabilities—specifically, its ability to recall, reason, and transfer relevant knowledge effectively.

On the other hand, RL methods often aim to elicit emergent reasoning capabilities by leveraging trial-and-error dynamics, sometimes leading to "aha moments" in reasoning. From an engineering standpoint, one could further fine-tune a KALE-pretrained checkpoint using GRPO (or similar RL techniques) to improve performance on specific downstream tasks such as long-chain-of-thought reasoning or self-correction.

We believe that integrating KALE into a broader LLM post-training framework that includes RL techniques is a promising direction for future research. Such a unified approach may yield deeper insights into the post-training landscape and further unlock the full potential of knowledge-aware language modeling.

H.18 More Results of KALE on Comparison of Prompt Distillation

Our proposed KA module utilizes rationales only during training by distilling the behavior induced by rationale-augmented inputs into models that do not require rationales at inference time. **This process is akin to knowledge distillation, where the model learns to emulate the internal reasoning patterns induced by rationales, without depending on them at test time.** Specifically, we use the rationale-enhanced model distribution as a soft target and train the model with an additional KL divergence loss to align the rationale-free predictions during test time. This approach allows the model to internalize the reasoning path during training and retrieve task-relevant knowledge during inference, even in the absence of explicit rationales. A variety of knowledge distillation approaches have investigated this paradigm (Lyu et al., 2024; Kang et al., 2023; Chan et al., 2023; Wang et al., 2023a). Therefore, we select and compare some recent popular knowledge distillation method, including Distilling Step-by-Step (abbreviated as Distill-SBS) (Hsieh et al., 2023), PD (Kujanpää et al., 2025), and CoT distillation (abbreviated as COT-Distill) (Chen et al., 2025b) to provide a more comprehensive understanding of KALE. To ensure a fair comparison, we adopt the standard settings of the PD method. For both CoT-Distill and Distill-SBS, we consistently use GPT-4o as the teacher model. Regarding CoT-Distill, we report the results obtained at the optimal granularity level. As shown in Table 26, our KALE consistently outperforms this prior method (referred to as prompt distillation (PD)), which also demonstrates the effectiveness of our KALE. We also emphasize that our primary objective lies not in distilling a compact model from a superior teacher, but rather in enhancing the model's knowledge manipulation capabilities via our KALE framework. From this perspective, our approach is orthogonal to distillation-based methods, and investigating their synergistic integration represents a promising avenue for future research.

We also want to clarify that KALE is not in conflict with any distillation methods. KALE is a flexible framework that provides a high-quality rationale generation approach and an effective training method to enhance the model's knowledge manipulation capabilities. **Any more specific methods for improving the model's performance on downstream tasks, including RL, distillation, contrastive learning, and others, can be seamlessly integrated with KALE to further enhance the model's performance in specific domains.**

Table 22: Comparison of vanilla, SFT, and KALE models on open-ended generation settings.

	Vanilla			SFT		
Model	Win Rate	Loss Rate	Tie Rate	Win Rate	Loss Rate	Tie Rate
KALE	82.5	6.25	11.25	77.5	8.75	13.75

Table 23: Results of KALE on Qwen3-32B-thinking model.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Vanilla	82.33	81.06	84.60	71.99	83.61	87.38	77.47	81.48
SFT	85.82	84.22	87.54	74.53	85.56	88.73	80.05	84.57
KALE	**93.60**	**91.09**	**93.43**	**79.08**	**89.44**	**92.02**	**84.61**	**88.54**

H.19 More Results of KALE on Joint Training Settings

In the main text, we adopt the setting of fine-tuning on each benchmark's training set separately, primarily to ensure consistency with prior work, such as MeanLearn (Xiong et al., 2024), which follows the same protocol. This alignment allows for a more direct and fair comparison with existing methods.

To validate the effectiveness of KALE on cross-task generalization settings, we have conducted an additional joint training experiment. Specifically, we fine-tune our model using the merged training sets of the eight benchmarks listed in Table 27 in the original manuscript, using Llama3 8B as the backbone model.

Notably, we observe that joint training does not significantly degrade performance across the benchmarks. On the contrary, the model even achieves improved performance for the logical reasoning task on datasets including AbsR, Commonsense, and BBH.

These findings highlight that KALE is not limited to a task-specific fine-tuning paradigm. Instead, it is compatible with multi-task or unified training setups, further demonstrating its flexibility and potential for real-world deployment scenarios where a single model is required to handle diverse tasks.

I More Discussions On KALE

What named entity recognition method is employed in KALE, and does it have any tailored designs? Given the relative maturity of named entity recognition (NER), we do not elaborate on it in the main text. Considering the need for rapid deployment and ease of implementation, we utilized the SpaCy open-source library for the NER component. Moreover, we employ noun phrase extraction from the NLTK library to retain some non-named yet significant nouns in given Q&A pairs. We reference the entity list from Wikidata for entity recognition.

The entity linking is a very important component in the proposed approach. Which entity linker is used for KALE, and does it have any tailored designs? In our entity linking process, we employ a simple keyword-matching algorithm as our entity linker. Given that a KG may contain multiple distinct entities with identical names, our entity linking might link incorrect entities, i.e., entity linking errors. In the entity linking process of KALE, potential errors may not be consistently measured and filtered out. Instead, for each set of candidate entities, we link their neighbor entities to enrich candidate entities. Then, we utilize the context to retain the relevant candidates. Generally, incorrect candidate entities are unrelated to the context, which are less likely to be retained. For instance, neighbor entities related to "Apple" (fruit) include "Cider Making", "Wild Apples", and "Apples from Maine" whereas those related to "Apple" (company) include "FileMaker", "Apple Germany", and various products like "iPod", "iMac", and "iPhone". If the correct entity in the reference context is "Apple" (fruit), it is less likely that neighbors related to Apple (company) appear in the reference context and vice versa. Moreover, in certain cases, we might still retain some incorrect entities (unrelated to the query). However, in most situations where the context and query are related, these incorrect entities are less likely to be key lexical units of the context. We acknowledge that integrating statistical or deep learning-based

Table 24: Results of self-taught KALE using Llama3 8B as the backbone model.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Vanilla	62.68	66.79	69.90	58.72	55.88	46.54	53.35	57.02
KALE$_{self\ taught}$	**75.31**	**73.19**	**80.08**	**60.84**	**58.59**	**52.11**	**61.19**	**67.57**

Table 25: Comparison of GRPO and KALE using Llama3 8B as the backbone model.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
GRPO	75.83	75.77	73.53	63.64	59.37	51.07	63.32	68.87
KALE	**83.62**	**81.23**	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**

entity linkers—such as TAGME (Ferragina and Scaiella, 2010), DBpedia Spotlight (Mendes et al., 2011), BLINK (Wu et al., 2020), and GENRE (Cao et al., 2021)—represents a promising direction for further optimizing KALE. Crucially, it is worth noting that the modules within KALE are designed to be highly decoupled. This architecture ensures flexibility in practical applications, allowing users to independently select and substitute linkers for downstream tasks according to specific resource constraints or precision requirements.

Why is the A* algorithm employed for knowledge-induced data synthesis instead of the naïve BFS algorithm? In knowledge-induced data synthesis, we select the A* algorithm over the naïve BFS based on algorithmic efficiency. The A* algorithm guides the search direction by incorporating a heuristic function $h(\mathbf{n})$, which significantly reduces the exploration scope. Particularly in large-scale KGs such as Wikidata, employing BFS to identify reasoning paths often requires days of computation. As mentioned in Section 4.1, the extraction of reasoning paths from the AbsR's training set **requires over one week**. Therefore, we propose an efficient multi-path A* algorithm to extract reasoning paths. It requires **less than 4 hours** to extract all reasoning paths on the same set. Consequently, we adopt the A* algorithm as a scalable and efficient solution for reasoning path search.

Is it possible for some reasoning paths to not reach the answer entity? During the process of extracting reasoning paths, instances may arise where the hop between the question entity and the answer entity exceeds the predefined threshold m. Nevertheless, the statistical data on the ABS dataset, as in Table 12, indicates that less than 2% of the 3-hop inference paths are unable to reach the answer entity. This suggests that employing 3-hop inference paths is a highly effective approach for extracting relevant information from the question to the answer. In such cases, we utilize the partial reasoning path that can be extracted-the path from the question entity to its neighboring entities within three hops—as enriched information for input. The ablation study results in Tables 2 and 15 further demonstrate the simplicity and effectiveness these types of reasoning paths.

If the KG contains errors that lead to incorrect reasoning paths, would GPT-4o generate wrong rationales? (i) Owing to Wikidata's factually accurate, high-quality, community-driven, and dynamically growing character, extracted reasoning paths contain **negligible** factual or logical errors. This motivates us to generate high-quality rationales via large-scale Wikidata. (ii) To address potential errors in the rationale generation, we leverage GPT's in-context learning (ICL) by incorporating specific instructions in the prompt. This allows for a filtering and correction mechanism to be implicitly applied during reasoning. As shown in Appendix C, we instruct LLM to generate rationales by referring to the provided reasoning path: *however, if the given reasoning path is irrelevant to the QA, generate a rationale based on your own knowledge.* This instruction helps that incorrect information is reduced. We empirically observe that utilizing the in-context learning ability is simple yet effective to reduce the error propagation with great domain robustness. Moreover, the design logic behind KALE is simplicity and practicality. We acknowledge that using a dedicated filtering mechanism could be an option.

What is the reason for choosing the KL divergence as the loss function? The selection of KL

Table 26: Comparison of PD and KALE using Llama3 8B as the backbone model.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
Distill-SBS	77.61	74.32	75.25	60.85	58.72	56.75	60.29	63.16
PD	75.31	72.17	76.98	57.73	60.17	55.41	58.89	68.91
COT-Distill	80.64	77.75	82.92	61.64	60.94	54.14	65.50	69.57
KALE	**83.62**	**81.23**	**86.45**	**65.69**	**63.27**	**57.33**	**68.61**	**74.12**

Table 27: Results of KALE on joint training experiments using Llama3 8B as the backbone model.

Method	AbsR	ARC-c	ARC-e	Common	MMLU	BBH	RACE-h	RACE-m
KALE_ori	83.62	**81.23**	**86.45**	65.69	**63.27**	57.33	**68.61**	**74.12**
KALE_joint	**84.48**	78.16	84.13	**67.81**	60.68	**58.13**	64.41	71.57

divergence is due to its ability to quantify the difference between two probability distributions. It encourages the model to compress the information contained in the rationale into its parameters θ. By forcing the two distributions to align, the model must "internalize" the information from the rationale $\mathbf{x}^{\mathbf{rats}}$ into its parameters θ, such that it can perform well even when $\mathbf{x}^{\mathbf{rats}}$ is unavailable (e.g., at test time). This minimization process implicitly guides the model to capture the underlying structure of the data, thereby facilitating the learning of meaningful representations without explicit supervision. Furthermore, KL divergence is essentially composed of entropy and cross-entropy. The knowledge-aware learning module in KALE can be viewed as a distillation process, designed to enhance the knowledge manipulation capabilities of LLMs during the testing phase, where rationale input is unavailable. **The addition of KL divergence is intended to enable the model to dynamically retrieve the task-relevant knowledge it has already mastered, which improves its knowledge manipulation capability.** We also believe that a theoretical analysis of KALE, especially the KL divergence part, could lead to a deeper understanding of our KALE. We agree that this is a promising direction for future work.

J LLM Usage

We used a large language model (LLM)–based writing assistant for grammar and wording improvements on draft text. The LLM did not generate research ideas, claims, proofs, algorithms, code, figures, or analyses, and it did not have access to any non-public data. During rationale generation, we use LLMs to transfer reasoning path into rationales. All edits suggested by the LLM were manually reviewed and either accepted or rewritten by the authors, who take full responsibility for the final content. The LLM is not an author of this paper.