KNOWPROXY: ADAPTING LARGE LANGUAGE MODELS BY KNOWLEDGE-GUIDED PROXY

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting large language models (LLMs) using smaller proxy models has been shown to improve training efficiency, where the LLMs remain frozen while the proxies are tuned on top. However, this approach typically requires access to the output probability distributions of LLMs, which are often inaccessible or unstable. To address this limitation, we propose KNOWPROXY, a knowledge-guided proxy framework in which the proxy is trained with textual knowledge rather than probability distributions. Specifically, we first elicit textual knowledge and reasoning from frozen LLMs through prompting, and then the proxy model learns to adapt this reasoning to target task distributions. We evaluate KNOWPROXY on diverse reasoning benchmarks with different fine-tuning scenarios. Comprehensive results show that KNOWPROXY achieves competitive or even better performance without direct access to probability distributions, thereby providing a scalable and versatile alternative to traditional fine-tuning.[1]

1 INTRODUCTION

Large Language Models (LLMs) have achieved remarkable performance across a variety of NLP tasks (OpenAI et al., 2024; Team et al., 2024; Grattafiori et al., 2024). To ensure that such LLM behaviors align with human intentions and the specific requirements of downstream tasks, fine-tuning plays a crucial role. However, their computational demands make direct fine-tuning of these LLMs resource-intensive and often impractical for many cases (Zhao et al., 2024; Miles et al., 2024). Moreover, when dealing with proprietary LLMs, which are frequently closed-source and inaccessible, direct modification becomes impossible.

One promising approach to this limitation is to train smaller proxy language models on top of the outputs from LLMs (Liu et al., 2024; Ormazabal et al., 2023). For example, proxy-tuning (Liu et al., 2024) adjusts LLM's outputs through a lightweight proxy that reweights its probability distributions. Similarly, CombLM (Ormazabal et al., 2023) trains a smaller model separately and combines its predictive distribution with those of the LLM. Despite their effectiveness, these approaches share two critical limitations: (i) they assume access to the full probability distributions and a shared vocabulary between LLMs and proxies, restricting applicability to black-box LLMs that provide only textual outputs. (ii) recent studies show that LLM-generated probability distributions are often unstable and unreliable (Atil et al., 2024; Gu et al., 2024), which can degrade downstream performance.

To overcome these limitations, we propose KNOWPROXY, a knowledge-guided proxy framework for adapting LLMs without relying on their probability distributions. Instead of accessing predictive probabilities, KNOWPROXY elicits textual knowledge and reasoning from frozen LLMs through prompting. A lightweight proxy model is then optimized on the elicited knowledge together with the input query, learning to map LLM-derived reasoning and knowledge into the target task distribution. This design enables adaptation even for proprietary black-box LLMs, while also mitigating instability by avoiding reliance on probability distributions.

A remaining challenge for proxy-based adaptation is the additional inference cost introduced by always involving the proxy model. To address this, KNOWPROXY incorporates a dynamic routing mechanism that adaptively determines when the proxy is required. Specifically, we elicit uncertainty scores for the LLM's generated reasoning and knowledge, and use these scores to decide whether

[1] Our code and data are available at `https://anonymous.4open.science/r/knowproxy-FC79`

to invoke the proxy model. In this way, KNOWPROXY maintains efficiency by selectively engaging the lightweight model only when the LLM's outputs are deemed unreliable.

We evaluate KNOWPROXY on a diverse set of reasoning benchmarks under multiple configurations. Experimental results show that KNOWPROXY consistently outperforms existing proxy-based methods, achieving superior accuracy and robustness while requiring no direct access to probability distributions. Notably, it delivers strong performance even in black-box settings where conventional fine-tuning is infeasible. In summary, the contributions of this work include the followings:

- We introduce KNOWPROXY, a novel proxy-based fine-tuning framework that adapts LLMs through textual knowledge and reasoning rather than probability distributions, enabling applicability to black-box settings.
- We integrate a dynamic routing mechanism into KNOWPROXY, which adaptively activates the proxy model only when necessary, thereby enhancing efficiency while preserving accuracy.
- We demonstrate that KNOWPROXY outperforms proxy-based methods and achieves comparable performance with the direct fine-tuning, highlighting its practical value in fine-tuning scenarios.

2 RELATED WORKS

2.1 FINE-TUNING LARGE LANGUAGE MODELS BY PROXY

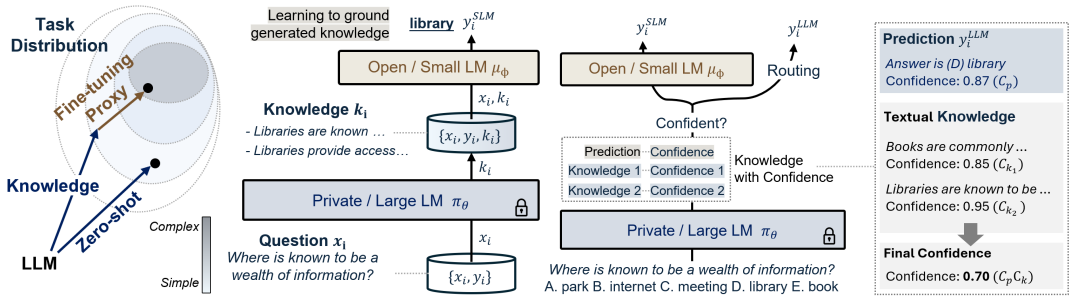
As billion-scale LLMs increasingly dominate applications and research communities, fine-tuning them has become even more challenging—even with parameter-efficient methods such as low-rank adaptation (LoRA) (Hu et al., 2022) and adapters (Houlsby et al., 2019). To address these challenges, an alternative line of work has investigated proxy-based approaches, where smaller language models (i.e., proxy) are trained on the outputs of frozen LLMs to adapt them to the target domain (Liu et al., 2024; Ormazabal et al., 2023). For instance, CombLM (Ormazabal et al., 2023) trains a separate smaller model and combines its predictive distribution with that of the LLM. Similarly, proxy-tuning (Liu et al., 2024) employs a lightweight model to reweight the predictive distributions of LLMs. However, these methods require access to the probability distributions of LLMs and assume a shared vocabulary space between the LLMs and the smaller model.

Compared to these methods, KNOWPROXY has distinct properties. Instead of relying on the probability distributions of LLMs, KNOWPROXY leverages the textual knowledge and reasoning elicited from frozen LLMs. This design makes it directly applicable to black-box settings, where only text outputs are available. Moreover, by shifting from probability distributions to textual representations, KNOWPROXY avoids the instability and unreliability issues often observed in LLM-generated distributions (Atil et al., 2024; Gu et al., 2024). The proxy model is thus trained to internalize and adapt reasoning expressed in text, yielding more stable and transferable performance across tasks. Finally, KNOWPROXY integrates an adaptive routing mechanism, ensuring that the proxy is invoked only when additional reasoning is required, thereby improving both efficiency and robustness.

2.2 ELICITING UNCERTAINTY FROM LANGUAGE MODELS

Assessing the confidence (or uncertainty) elicited from LLMs is crucial for improving the factuality of their responses and enhancing the quality of generated text (Geng et al., 2024). Previous studies have primarily extracted uncertainty from the output probabilities (Mielke et al., 2022; Kuhn et al., 2023; Duan et al., 2024). However, this approach faces applicability constraints, as it cannot be applied to black-box models whose internal probabilities are inaccessible. To address this limitation, several studies have proposed prompting-based methods to compute uncertainty scores directly from LLM textual outputs—applicable to both open-source and API-based models—showing that these scores correlate well with model performance and output quality (Tian et al., 2023; Xiong et al., 2024; Dong et al., 2024).

Building on these studies, we use elicited uncertainty scores primarily for adaptive routing. Unlike prior proxy-based methods that always invoke the proxy model, KNOWPROXY selectively engages it only when the LLM's outputs are judged uncertain or unreliable, thereby reducing inference overhead while maintaining robustness. Moreover, whereas previous work has considered uncertainty only at the prediction level, our design estimates uncertainty for each piece of elicited knowledge, allowing finer-grained routing decisions and more stable adaptation.



(a) Concept (b) Training (c) Inference

Figure 1: Conceptual illustration of KNOWPROXY. (a) A smaller proxy model is trained using textual knowledge generated by LLMs to better capture the target task distribution. (b) During training, knowledge is elicited from LLMs via prompting and incorporated as an auxiliary input to the proxy. (c) At inference, KNOWPROXY utilizes multiple confidence scores from LLMs to perform dynamic routing, thereby balancing efficiency and accuracy.

3 KNOWPROXY: KNOWLEDGE-GUIDED PROXY

We introduce KNOWPROXY, a framework that adapts LLMs using smaller proxy models based on textual knowledge and reasoning elicited from frozen LLMs. We first outline the problem setup of the LLM adaptation (§3.1), then describe how we elicit textual knowledge and reasoning from LLMs and train proxy models to align these representations with target task distributions (§3.2). Finally, we present an adaptive routing mechanism that reduces inference overhead by selectively invoking the proxy model only when needed (§3.3). Figure 1 illustrates the workflow of KNOWPROXY.

3.1 PRELIMINARIES AND PROBLEM FORMULATION

We first revisit the traditional fine-tuning and define the problem setup in our method.

Direct Fine-tuning. A straightforward way to train LLMs is to perform direct fine-tuning using a supervised objective defined over a training dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=0}^{N-1}$, where N denotes the dataset size. The fine-tuning objective is formulated as:

$$\min_{\theta} -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \pi_\theta(y \mid x)\right] \tag{1}$$

Here, π_θ denotes the language modeling function of the LLM parameterized by θ. However, directly fine-tuning θ is often impractical due to two major challenges: (1) the massive scale of θ, which renders optimization computationally prohibitive, and (2) the restricted accessibility of θ in black-box LLMs, where the underlying parameters are not exposed.

Proxy-based Fine-tuning. To overcome these challenges, proxy-based fine-tuning approaches have been proposed, in which a smaller, accessible model μ_ϕ, parameterized by ϕ, serves as a proxy for the fine-tuned LLM (Ormazabal et al., 2023; Liu et al., 2024). The training objective is reformulated as:

$$\min_{\phi} -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \mu_\phi(y \mid x)\pi_\theta(y \mid x)\right] \tag{2}$$

By optimizing the smaller model μ_ϕ on the dataset using the predictive distributions provided by the LLM, this approach effectively performs fine-tuning through the proxy without requiring access to the LLM's parameters.

However, this approach has two key limitations rooted in its reliance on probability distributions: (i) it requires access to the predictive distributions of LLMs, which are often unavailable, and (ii) even when accessible, these distributions are frequently unstable (Atil et al., 2024; Gu et al., 2024). To address this, we reformulate proxy-based fine-tuning to leverage textual knowledge generated by LLMs, making it applicable even in black-box scenarios. Given an input x, the LLM generates

knowledge k according to $\pi_\theta(k \mid x)$, where k denotes textual knowledge or reasoning extracted from the model. This leads to the following knowledge-guided objective:

$$\min_{\phi} -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \mu_\phi(y \mid x, k)\right] \text{ where } k \sim \pi_\theta(k|x)$$

Here, k represents the generated textual knowledge that provides additional context for solving the query x. This reformulation enables effective fine-tuning through proxies without relying on the LLM's predictive distributions. Moreover, because the proxy model directly learns to leverage knowledge generated by the LLM, it can internalize this information in a more stable and controlled manner. In the following sections, we describe how textual knowledge is elicited from LLMs and how the proxy model is trained to align this knowledge and reasoning to target tasks.

3.2 Adaptation with Knowledge-guided Proxy

Knowledge and Reasoning Generation. To optimize the reformulated training objective (Eq. equation 9), KnowProxy begins by eliciting textual knowledge for the training dataset $\mathcal{D}$. In this work, we define knowledge as the relevant cues required to solve a given problem, such as underlying principles (Cai et al., 2024), reasoning steps (Wei et al., 2022), or relevant facts (Park et al., 2024). This knowledge is obtained through knowledge-eliciting prompts to LLMs. Along with the knowledge, we also extract a confidence score for each piece, which serves as an estimate of its reliability and allows weighting according to its expected correctness. This process can be formally represented as follows:

$$k, c = \pi_\theta(P_k, x) \tag{3}$$

Here, P_k denotes the knowledge-eliciting prompt (see Appendix §B.3 for the complete list of prompts), while k and $c \in [0, 1]$ represent the extracted textual knowledge and its associated confidence score for solving the given query x. Importantly, rather than relying on a single output, we generate multiple knowledge–confidence pairs. This design captures the diversity of potential reasoning paths and mitigates the risk of over-reliance on any individual extraction.

However, the knowledge generated by LLMs is not always reliable for solving the given query, as it may contain hallucinations or irrelevant information. To mitigate this issue, we apply a filtering process to the generated knowledge, defined as follows:

$$k = \{k_i \mid (k_i, c_i) \in \mathcal{K}, c_i > \alpha\}, \tag{4}$$

Here, $\mathcal{K}$ denotes the set of knowledge–confidence pairs for the given query, and α is a predefined threshold that specifies the minimum confidence level required to retain knowledge. After applying this filtering process, we construct the knowledge-augmented training dataset $\mathcal{D}_\mathcal{K} = \{(x^i, k^i, y^i)\}_{i=0}^{N-1}$, where each training instance (x^i, k^i, y^i) consists of the original input x^i, the corresponding filtered knowledge set k^i, and the target output y^i.

Proxy Optimization with Generated Knowledge. Based on the knowledge-augmented dataset, we train a smaller language model to align the reasoning and knowledge elicited from LLMs with the target task distributions. Given a query x and its associated knowledge k, we construct an augmented input by concatenating x and k, and train the proxy model μ_ϕ with a standard supervised objective. This training procedure enables the proxy to map LLM-derived reasoning into task-specific outputs, thereby adapting the frozen LLM to downstream requirements. Further details on the knowledge adaptation process are provided in Appendix A.

3.3 Adaptive Reasoning in KnowProxy

Although adapting smaller models to fine-tune LLMs through proxies is effective, it introduces additional inference costs for every query. A more efficient strategy is to invoke the proxy model only when necessary—specifically, when the LLM alone is unlikely to produce a correct output. To achieve this, we incorporate a dynamic routing mechanism into the inference phase of KnowProxy.

Confidence Elicitation To implement the dynamic routing process, we first obtain the LLM's prediction for the given query along with its associated confidence score $C_{\textbf{prediction}}$. We efficiently achieve this by augmenting the knowledge-eliciting prompt P_k (Eq. equation 3) with instructions to

elicit both the prediction and its confidence. For the routing decision, we incorporate not only the prediction confidence but also the confidence scores of the generated knowledge. This is because the reliability of the generated knowledge provides additional insight into whether the LLM's reasoning is sound; low-confidence knowledge may indicate uncertainty or hallucination, signaling the need to invoke the small model for improved accuracy. We thus derive the final confidence scores as follows:

$$C_{\text{final}} = C_{\text{knowledge}} \cdot C_{\text{prediction}}, \text{ where } C_{\text{knowledge}} = \prod_{k=1}^{K} c_k \tag{5}$$

Here, K denotes the predefined number of generated knowledge instances. Since multiple pieces of knowledge are elicited for each query, we aggregate their confidence scores to obtain a single reliability measure[2]. Note that, when aggregating the final confidence score, we include all confidence scores—even those from filtered knowledge—in order to capture the LLM's comprehensive understanding of the given query.

Adaptive Reasoning Paths KNOWPROXY determines whether to rely directly on the LLM's prediction or to engage the proxy model trained to leverage generated knowledge. If the aggregated confidence score (Eq. equation 5) exceeds a predefined threshold, the system outputs the LLM's prediction as the final answer without invoking the proxy. Conversely, if the confidence score falls below the threshold, KNOWPROXY activates the proxy model, which incorporates the generated knowledge to refine the prediction. This decision process can be formalized as follows:

$$y = \begin{cases} \pi_\theta(y|x), & \text{if } C_{\text{final}} \geq \tau \\ \mu_\phi(y \mid x, k \sim \pi_\theta(k|x)), & \text{if } C_{\text{final}} < \tau \end{cases} \tag{6}$$

Here, τ denotes the predefined threshold. Through this adaptive reasoning path, KNOWPROXY dynamically balances computational efficiency and prediction accuracy by invoking the proxy model only when the LLM's prediction is deemed unreliable. This design ensures that inference remains efficient while maintaining accurate predictions across diverse queries.

4 EXPERIMENTS

In this section, we evaluate KNOWPROXY to verify its efficacy on fine-tuning scenarios. Specifically, we aim to answer the following research questions:

- Does KNOWPROXY achieve performance comparable to existing proxy-based methods?
- Is KNOWPROXY effective across diverse fine-tuning scenarios, including black-box LLMs?
- Can KNOWPROXY effectively reduce the additional costs introduced by proxy-based training?

4.1 EXPERIMENTAL SETUPS

Datasets. We evaluate KNOWPROXY on a broad suite of complex reasoning benchmarks, including OpenBookQA (Bhakthavatsalam et al., 2021), ARC-Challenge (Mihaylov et al., 2018), CommonsenseQA (Talmor et al., 2019), QASC (Khot et al., 2020), PhysicalIQA (Bisk et al., 2019), SocialIQA (Sap et al., 2019), Winogrande (Sakaguchi et al., 2019), BoolQ (Clark et al., 2019), and StrategyQA (Geva et al., 2021). We also include distinct benchmarks such as TruthfulQA (Lin et al., 2022), and ScienceQA (Lu et al., 2022). Further details on these datasets are provided in Appendix B.1.

Baselines. We primarily compare KNOWPROXY with recent proxy-based approaches, including CombLM (Ormazabal et al., 2023) and Proxy-tuning (Liu et al., 2024). We also include BBox-Adapter (Sun et al., 2024), which, while not a proxy-based method, trains a smaller evaluation model to select better answers from multiple samples generated by the LLM. This provides a complementary baseline, as it adapts LLM outputs through answer selection rather than distributional alignment. In addition, we evaluate against established reasoning-based approaches, i.e., Self-talk (Shwartz et al., 2020), Zero-shot-CoT (Kojima et al., 2022), and Plan-and-Solve (Wang et al., 2023).

[2]In Section 4, we demonstrate that our confidence aggregation strategy yields more reliable estimates than existing confidence elicitation methods.

Method	OBQA	ARC$_h$	PIQA	CSQA	QASC	SIQA	WNGR	StrategyQA	BoolQ	Avg.
Fine-tuning LLM	82.2	76.2	87.7	79.5	82.9	80.5	87.3	71.5	86.9	81.6
Fine-tuning SLM	73.2	60.9	80.3	72.0	68.0	74.9	75.4	66.5	85.4	73.0
Advanced Zero-shot Reasoning (Frozen LLM)										
Zero-shot	72.2	68.6	75.8	67.7	75.9	65.3	53.6	60.9	78.3	68.7
Self-Talk	74.8	74.2	75.9	72.3	**80.5**	67.0	54.6	57.6	<u>78.5</u>	70.6
Chain-of-Thought	77.6	**80.0**	75.6	73.1	79.0	68.6	57.8	<u>69.0</u>	76.7	73.1
Plan-and-Solve	76.6	<u>75.3</u>	74.3	<u>73.7</u>	<u>79.8</u>	66.4	58.2	66.8	73.9	71.7
Proxy-based Training (Frozen LLM + SLM)										
Proxy-tuning	77.2	69.6	80.1	70.8	69.9	72.6	65.7	64.6	76.2	71.9
CombLM	<u>78.6</u>	72.6	<u>81.1</u>	72.5	76.9	<u>73.7</u>	<u>69.3</u>	67.2	76.8	<u>74.3</u>
BBox-Adapter	76.2	68.6	73.8	73.3	73.8	72.7	53.7	<u>69.0</u>	70.5	70.2
KNOWPROXY (ours)	**80.2**	75.2	**83.4**	**75.0**	78.1	**76.3**	**77.8**	**72.9**	**85.1**	**78.2**

Table 1: Evaluation results for test accuracy (%) on nine reasoning benchmarks. The best and second-best results are highlighted in **boldface** and <u>underlined</u>, respectively. In these experiments, we use Llama-3.2 (3B) as the frozen LLM and Llama-3.2 (1B) as the smaller proxy (SLM).

Backbone. To demonstrate the applicability of KNOWPROXY across diverse LLMs, we evaluate it on two categories: API-based models (ChatGPT (GPT-3.5-turbo)) and open-source models (i.e.,Llama-3.2-3B-Instruct[3], Mistral-7B-Instruct-v0.2[4], and Llama-2-13B-Chat (Touvron et al., 2023). Furthermore, to showcase the adaptability of KNOWPROXY, we conduct experiments using various small models, such as Llama-3.2-1B-Instruct[5], LaMini-GPT-774M (Wu et al., 2024), Qwen2.5-0.5B-Instruct (Qwen et al., 2025), and Pythia family (Biderman et al., 2023), covering a range of model sizes and model families. The experimental details are provided in Appendix B.

4.2 MAIN RESULTS

To validate the effectiveness of our approach, we first compare KNOWPROXY with existing proxy-based methods that adapt LLMs by redistributing their predictive distributions through lightweight models (e.g., Proxy-tuning, CombLM) or by generating adapted responses via multi-step beam search, where candidate responses are ranked by small models (e.g., BBox-Adapter). As shown in Table 1, KNOWPROXY consistently outperforms these methods across all reasoning benchmarks by a substantial margin. Notably, KNOWPROXY even achieves accuracy comparable to direct fine-tuning of LLMs on several tasks (e.g., OpenBookQA, ARC-Challenge, StrategyQA, and BoolQ)—a performance level not previously attained by proxy-based approaches. Additionally, we observe that KNOWPROXY is effective even on domain-specific tasks, including generative tasks. More detailed results can be found in the Appendix C.1.

We further observe that existing proxy-based methods sometimes underperform even the zero-shot reasoning capabilities of LLMs on certain benchmarks (e.g., QASC and BoolQ), likely due to the instability of LLM probability distributions. In contrast, KNOWPROXY delivers substantial improvements on these tasks, indicating that proxies trained on textual representations provide greater robustness and reliability than distribution-based approaches.

4.3 GENERAL APPLICABILITY TO ARCHITECTURES AND SCALES

Applicability across LLM architectures and scales. To demonstrate the broad applicability of KNOWPROXY (i.e., its model-agnostic nature), we evaluate its indirect fine-tuning effectiveness across diverse scenarios, including quantized LLMs (Llama-2 (13B))[6] and API-based black-box LLMs (ChatGPT). As shown in Table 2, KNOWPROXY consistently enhances performance across all benchmarks, regardless of the underlying LLM's capabilities. Notably, KNOWPROXY achieves significant performance gains even with black-box models (e.g., ChatGPT). These results highlight the plug-and-play nature of KNOWPROXY, which can effectively enhance both computationally demanding open-source models and black-box models with inaccessible internal parameters. By

[3]https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

[4]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[5]`https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`

[6]We follow setup from (Dettmers et al., 2023).

LLM	Approach	OBQA	ARC$_h$	PIQA	CSQA	StrategyQA	QASC	SIQA	Avg.
Mistral (7B)	Fine-tuning	87.2	74.5	88.4	82.9	72.9	81.5	79.8	81.0
	Zero-shot	72.6	72.1	75.7	70.4	42.8	65.9	69.6	67.0
	KNOWPROXY	**81.0**	**73.8**	**84.3**	**74.5**	**69.4**	**75.8**	**77.4**	**76.6**
Llama 2 (13B) (4-bit quantized)	Fine-tuning	84.6	73.8	87.8	82.0	70.7	78.6	81.7	79.9
	Zero-shot	61.0	56.3	72.6	53.3	55.0	52.9	48.6	57.1
	KNOWPROXY	**75.0**	**63.9**	**81.3**	**75.4**	**71.2**	**71.3**	**75.6**	**73.4**
ChatGPT (gpt-3.5-turbo)	Zero-shot	78.8	81.2	82.8	76.3	68.1	79.0	72.3	76.9
	BBox-Adapter	79.2	83.3	**88.3**	77.7	73.8	80.0	73.2	79.4
	KNOWPROXY	**85.0**	**83.9**	87.2	**78.1**	**74.7**	**80.2**	**77.0**	**80.9**

Table 2: Evaluation results for test accuracy (%) with diverse backbone LLMs. The best results are highlighted in **boldface**. Here, we use Llama-3.2 (1B) as the proxy.

Model	Approach	OBQA	ARC$_h$	PIQA	CSQA	StrategyQA	QASC	SIQA	Avg.
Llama 3.2 (3B)	Fine-tuning	82.2	76.2	87.7	79.5	71.5	82.9	80.5	80.1
	Zero-shot	72.2	68.6	75.8	67.7	60.9	75.9	65.3	69.5
w/ Llama 3.2 (1B)	KNOWPROXY	**80.2**	75.2	**83.4**	**75.0**	**72.9**	78.1	**76.3**	**77.3**
w/ LaMini-GPT (0.7B)	KNOWPROXY	74.6	75.2	78.7	72.5	67.7	78.0	71.9	74.1
w/ Qwen 2.5 (0.5B)	KNOWPROXY	76.2	**75.3**	79.9	72.8	69.4	**78.4**	73.9	75.1

Table 3: Evaluation results for test accuracy (%) with diverse small language models. The best results are highlighted in **boldface**. Here, we use Llama-3.2 (3B) as the frozen LLM.

seamlessly integrating a smaller language model, KNOWPROXY provides a robust mechanism for indirect fine-tuning across a wide range of LLMs.

Applicability across proxy choices. We further analyze the general applicability of KNOWPROXY using a diverse set of smaller language models, with the results presented in Table 3. The experimental results demonstrate that KNOWPROXY consistently enhances the zero-shot performance of LLMs across all smaller models. Moreover, we observe that the performance gains of LLMs are proportional to the capability of the smaller model (see Appendix C.2 for more details). Notably, through these results, along with additional analysis on the scalability of small models (Appendix C.3), KNOWPROXY demonstrates the practical adaptability of proxies, aligning with the characteristics of scaling laws (Kaplan et al., 2020) under environments restricted to small models.

4.4 COMPONENT ANALYSIS IN KNOWPROXY

Ablation study. We conduct an ablation study to assess the contribution of each component in KNOWPROXY to downstream task performance, with results summarized in Table 4. Our analysis considers the following variants: (i) **w/o routing**: Removes the dynamic routing mechanism, such that the proxy model is invoked for all inputs regardless of the LLM's confidence. (ii) **w/o filtering**: Omits the filtering process during knowledge generation, allowing all elicited knowledge to be used. (iii) **w/o adaptation**: Excludes the training of the proxy model on LLM-generated knowledge; instead, the proxy is trained only on the original inputs, and the generated knowledge is incorporated solely at inference time.

Method	OBQA	PIQA	StrategyQA	SIQA
KNOWPROXY	**85.0**	**87.2**	**74.7**	**77.0**
w/o routing	82.0	87.2	74.7	76.8
w/o filtering	85.0	86.2	72.1	76.0
w/o adaptation	80.6	85.1	59.4	75.3

Table 4: Ablation of KNOWPROXY. We use ChatGPT as the LLM and Llama-3.2 (1B) as the proxy.

The results demonstrate that removing the filtering and adaptation processes from KNOWPROXY substantially degrades performance on downstream tasks, underscoring the importance of both components. In particular, knowledge adaptation is critical, as its removal causes the largest performance drop (81.0 to 75.1 average). By contrast, excluding routing inference results in only a marginal decrease (81.0 to 80.2 average), indicating that our routing design—introduced primarily to enable efficient inference—does not compromise accuracy. This outcome confirms that the mechanism

achieves its intended purpose: routing only the more difficult queries to the proxy model, while easier cases are effectively handled by the LLM alone, thereby preserving accuracy while reducing inference overhead.

Routing reliability. We further analyze the reliability of the routing mechanism under the proposed uncertainty measure. For routing to be reliable in the proxy framework, the samples directed to the LLM (rather than the proxy) should be those that the LLM can solve easily. To examine this property, we evaluate the performance of the routed samples handled by the LLM. Specifically, we compare two uncertainty metrics: the previous approach Tian et al. (2023), which relies on a single confidence score derived from the LLMs, and our proposed approach, which aggregates confidence scores across all generated knowledge. As shown in Figure 2, our method achieves higher performance as the confidence threshold increases, demonstrating its ability to distinguish easier cases that the LLM can solve reliably. In contrast, the baseline measure yields almost flat performance across different confidence thresholds, suggesting that a single prediction-level confidence does not provide meaningful guidance for routing. These results highlight that aggregating uncertainty across generated knowledge enables more effective confidence estimation and, in turn, more reliable routing decisions.

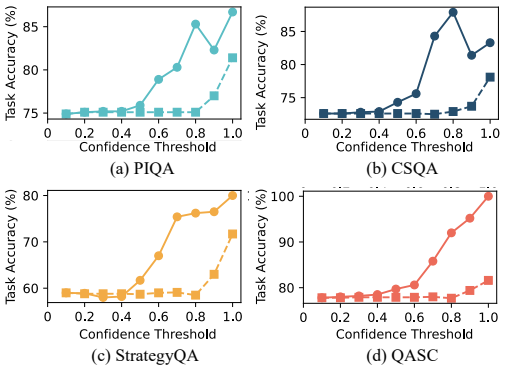


Figure 2: Confidence comparison between our method and baselines.

Number of Knowledge. Since the effectiveness of KNOWPROXY hinges on the amount of knowledge it elicits, we analyze how varying the number of generated knowledge instances influences performance. As shown in Figure 3, leveraging knowledge consistently improves results compared to the zero-shot LLM baseline, confirming its importance in adaptation. However, increasing the number of knowledge instances does not always yield additional gains and may even lead to performance degradation due to noisy or redundant reasoning, suggesting that generating a moderate number of knowledge pieces provides the most reliable improvements.

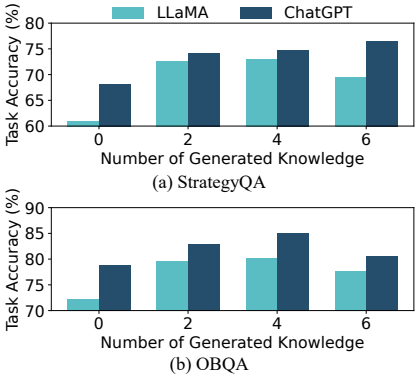


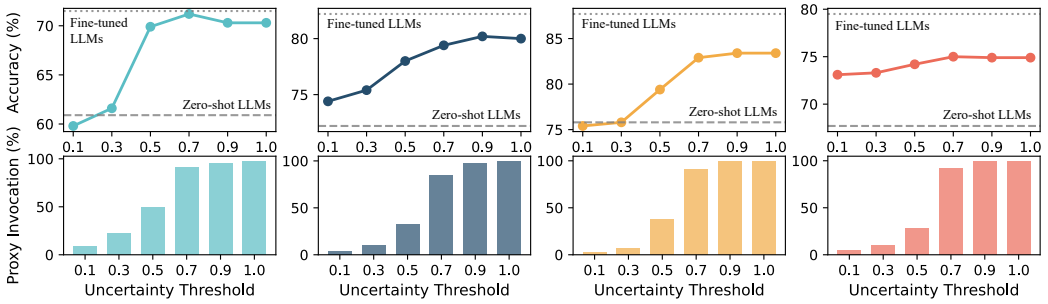
Figure 3: Performance across the different number of generated knowledge.

4.5 TRADE-OFF OF PROXY ROUTING

To further validate the dynamic routing, we explore the balance between performance and the frequency of proxy model invocation in Figure 4. Unlike prior methods that rely on the small model for every input, KNOWPROXY selectively invokes the proxy only when needed. Our results show that even with fewer proxy invocations, the framework maintains or improves performance across tasks, reaching accuracy levels competitive with direct fine-tuning of LLMs. This demonstrates that routing based on uncertainty not only improves efficiency by reducing unnecessary proxy use but also preserves accuracy, underscoring the benefit of training proxies on LLM-generated knowledge rather than redistributing predictive distributions. Additionally, We observe that KNOWPROXY, which leverages textual knowledge, is more memory-efficient than the LoRA approach during training. The detailed analysis can be found in Appendix C.4.

4.6 TRANSFERABILITY ACROSS DIVERSE LLMS

One notable advantage of KNOWPROXY is that it decouples the proxy model from the backbone LLM, allowing them to be swapped independently. This flexibility enables seamless replacement of the backbone LLM with newer models without requiring retraining of the proxy. To empirically validate this property, we conduct experiments by pairing a target LLM (ChatGPT) with proxy models



(a) StrategyQA (b) OBQA (c) PIQA (d) CSQA

Figure 4: Comparison of inference cost and performance according to confidence thresholds.

Method	Trained Knowledge	OBQA	ARC$_h$	PIQA	CSQA	StrategyQA	QASC	Avg.
Zero-shot	-	78.8	81.2	82.8	76.3	68.1	79.0	77.7
KnowProxy	ChatGPT	85.0	83.9	87.2	78.1	74.7	80.2	81.5
	Llama 3.2	81.0	83.5	85.1	76.7	69.9	78.4	79.1
	Mistral-v0.2	82.2	84.3	86.1	76.4	71.6	80.2	80.1

Table 5: Evaluation results for test accuracy (%). In these experiments, we train the small language models on different generated knowledge other than the target LLMs (i.e., ChatGPT).

that were trained on different LLMs. As shown in Table 4.6, these cross-trained proxies maintain strong performance when transferred to ChatGPT, despite being optimized with other LLMs. This demonstrates that the proxy learns to leverage elicited textual knowledge in a model-agnostic way, with language itself serving as the universal interface between LLMs and proxy models.

4.7 Analysis of Reasoning Gains from KnowProxy

KnowProxy adaptively combines the outputs of the LLM with those of the trained proxy model to generate the final reasoning. To examine how the proxy contributes beyond the LLM alone, we construct a confusion matrix based on the reasoning produced by the LLM and by KnowProxy (i.e., the LLM integrated with the fine-tuned proxy model). As shown in Figure 5, KnowProxy successfully corrects 38.7% of the reasoning that is incorrect when generated by the LLM. Notably, it achieves a 5.5% higher rate of correct reasoning (10.9%) compared to incorrect reasoning (5.4%). These results highlight that KnowProxy effectively adapts LLMs by training a smaller model to leverage the elicited knowledge, thereby improving reasoning quality over the base LLM.

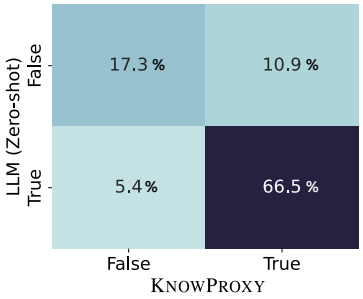


Figure 5: Confusion matrix comparing reasoning generated by the LLM and KnowProxy.

5 Conclusions

We have proposed KnowProxy, a novel framework for adapting LLMs through proxy models guided by elicited textual knowledge. Unlike prior approaches that have relied on predictive distributions, KnowProxy has leveraged knowledge expressed in natural language, making it broadly applicable across a wide range of LLMs, including black-box models that have only provided textual outputs. We have extensively evaluated the method across diverse benchmarks and training setups, including black-box LLMs, quantized LLMs with varying scales. The results have demonstrated that KnowProxy has consistently outperformed existing proxy-based approaches and even achieved performance comparable to direct fine-tuning. Our analysis has confirmed that the adaptive reasoning mechanism in KnowProxy has effectively balanced accuracy with efficiency, highlighting its promise as a promising alternative to direct fine-tuning.

REFERENCES

Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. Llm stability: A detailed analysis with some surprises. *arXiv preprint arXiv:2408.04667*, 2024.

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

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. *CoRR*, abs/1911.11641, 2019. URL http://arxiv.org/abs/1911.11641.

Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li. The role of deductive and inductive reasoning in large language models. *arXiv preprint arXiv:2410.02892*, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Yijiang River Dong, Tiancheng Hu, and Nigel Collier. Can LLM be a personalized judge? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10126–10141, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.592. URL https://aclanthology.org/2024.findings-emnlp.592/.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.276. URL https://aclanthology.org/2024.acl-long.276/.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.444. URL https://aclanthology.org/2024.emnlp-main.444/.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6577–6595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Bowen Gu, Rishi J Desai, Kueiyu Joshua Lin, and Jie Yang. Probabilistic medical predictions of large language models. *npj Digital Medicine*, 7(1):367, 2024.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090, Apr. 2020. doi: 10.1609/aaai.v34i05.6319. URL https://ojs.aaai.org/index.php/AAAI/article/view/6319.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. Tuning language models by proxy. In *First Conference on Language Modeling*, 2024.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=HjwK-Tc_Bc.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022. doi: 10.1162/tacl_a_00494. URL https://aclanthology.org/2022.tacl-1.50/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Roy Miles, Pradyumna Reddy, Ismail Elezi, and Jiankang Deng. Velora: Memory efficient training using rank-1 sub-token projections. *Advances in Neural Information Processing Systems*, 38, 2024.

OpenAI, Josh Achiam, and Steven Adler et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Aitor Ormazabal, Mikel Artetxe, and Eneko Agirre. Comblm: Adapting black-box language models through small fine-tuned models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2961–2974, 2023.

Jun-Hyung Park, Mingyu Lee, Junho Kim, and SangKeun Lee. Coconut: Contextualized commonsense unified transformers for graph-based commonsense augmentation of language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 5815–5830, 2024.

Ben Prystawski, Michael Y. Li, and Noah Goodman. Why think step by step? reasoning emerges from the locality of experience. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=rcXXNFVlEn.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WINOGRANDE: an adversarial winograd schema challenge at scale. *CoRR*, abs/1907.10641, 2019. URL http://arxiv.org/abs/1907.10641.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019. URL http://arxiv.org/abs/1904.09728.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4615–4629, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.373. URL https://aclanthology.org/2020.emnlp-main.373/.

Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. BBox-adapter: Lightweight adapting for black-box large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47280–47304. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sun24p.html.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Gemini Team, Rohan Anil, and Sebastian Borgeaud et al. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

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, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 944–964, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.57/.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. In *Proceedings of the International Conference on Machine Learning*, 2024.

A Proof for Knowledge Adaptation

We define knowledge-augmented training datasets as $\mathcal{D}_{\mathcal{K}} = \{(x^i, k^i, y^i)\}_{i=0}^{N-1}$. Using the constructed dataset $\mathcal{D}_{\mathcal{K}}$, we train a small model $\mu_\phi(y \mid x, k)$ parameterized by ϕ, to leverage the model-generated knowledge for the given target domains. The process by which the small language model predicts answers by leveraging knowledge from the large language model can be formulated as follows:

$$\pi'(y \mid x) \geq \mu_\phi(y \mid x, k)\, \pi_\theta(k \mid x), \tag{7}$$
$$\text{where} \quad \pi'(y \mid x) = \sum_{k_{all}} \mu_\phi(y \mid x, k_a)\, \pi_\theta(k_a \mid x), k_{all} \sim \pi_\theta(k|x).$$

By applying the empirical loss function on the dataset D, we can derive equation 8 as follows:

$$-\mathbb{E}_{\mathcal{D}}[\log \pi'(y \mid x)] \leq -\mathbb{E}_{\mathcal{D}}[\log \mu_\phi(y \mid x, k)] - \mathbb{E}_{\mathcal{D}}[\log \pi_\theta(k \mid x)]. \tag{8}$$

Therefore, our knowledge-guide objective function is as follows:

$$\min -\mathbb{E}_{\mathcal{D}}[\log \pi'(y \mid x)] \leq \min_\phi -\mathbb{E}_{\mathcal{D}}[\log \mu_\phi(y \mid x, k)] + \min_\theta -\mathbb{E}_{\mathcal{D}}[\log \pi_\theta(k \mid x)],$$
$$\text{where} \quad \pi'(y \mid x) = \sum_{k_{all}} \mu_\phi(y \mid x, k_a)\, \pi_\theta(k_a \mid x), k_{all} \sim \pi_\theta(k|x).$$

Here, based on previous studies (Prystawski et al., 2023; Gekhman et al., 2024) and empirical observations (Table 4), we assume as follows.

$$\min_\phi -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \mu_\phi(y \mid x, k')\right] \leq \min_\phi -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \mu_\phi(y \mid x, k)\right],$$
$$\text{where} \quad k' = \{k_i \mid (k_i, c_i) \in \mathcal{K}, c_i > \alpha\}.$$
$$\therefore L_{\text{KnowProxy}} = \min_\phi -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log \mu_\phi(y \mid x, k')\right]. \tag{9}$$

B Experimental Details

B.1 Dataset Details

OpenBookQA (Bhakthavatsalam et al., 2021) is a multiple-choice question-answering dataset on elementary science, designed to assess a model's commonsense knowledge.

ARC-Challenge (Mihaylov et al., 2018) is a multiple-choice question-answering dataset consisting of scientific questions that are difficult to solve using either a retrieval-based algorithm or a word co-occurrence algorithm, designed to evaluate a model's complex reasoning ability.

CommonsenseQA (Talmor et al., 2019) is a multiple-choice question-answering dataset designed to evaluate a model's commonsense knowledge across common world scenarios.

QASC (Khot et al., 2020) is a multiple-choice question-answering dataset in grade school science, designed to evaluate the multi-hop reasoning ability of models.

SocialIQA (Sap et al., 2019) is a multiple-choice question-answering dataset designed to measure a model's social and emotional intelligence.

ScienceQA (Lu et al., 2022) is a multi-modal dataset designed for question-answering in the science domain, which is accompanied by annotated answers, lectures, and explanations. The dataset contains approximately 21,000 multi-modal questions. Following the experimental setting of (Sun et al., 2024), we exclude questions requiring image input and then randomly sample 2,000 questions for training and 500 for testing from the original train and test splits.

PhysicalIQA (Bisk et al., 2019) is a binary question-answering dataset designed to evaluate a model's physical commonsense reasoning ability.

Winogrande (Sakaguchi et al., 2019) is a binary question-answering dataset designed to assess a model's commonsense knowledge by evaluating its ability to solve paired instances of coreference resolution.

BoolQ (Clark et al., 2019) is a binary question-answering dataset designed to assess a model's comprehensive reasoning ability using knowledge-intensive contexts and their associated questions.

StrategyQA (Geva et al., 2021) is a binary question-answering dataset designed to evaluate the ability of models to perform implicit multi-hop reasoning.

TruthfulQA (Lin et al., 2022) is a widely used generative dataset to evaluate a model's response quality in terms of truthfulness, factuality, and accuracy. The original dataset consists only of a test set. For evaluation, following the experimental setting of (Sun et al., 2024), we randomly select 100 samples to construct a test set and utilize the remaining samples as a train set.

The statistics of the datasets are provided in Table 6.

Dataset	Answer type	# of train data	# of test data
OpenBookQA	Multiple-choice	4,957	500
ARC-Challenge	Multiple-choice	1,119	299
CommonsenseQA	Multiple-choice	9,741	1,221
QASC	Multiple-choice	8,134	926
SocialIQA	Multiple-choice	33,410	1,954
ScienceQA	Multiple-choice	2,000	500
PhysicalIQA	Binary-choice	16,113	1,838
Winogrande	Binary-choice	40,398	1,267
BoolQ	Binary (True/False)	9,427	3,270
StrategyQA	Binary (True/False)	2,061	229
TruthfulQA	Open-ended text	717	100

Table 6: Dataset descriptions and statistics.

B.2 Implementation

We implement KnowProxy on Huggingface Transformers and PyTorch, and conduct all our experiments on two NVIDIA RTX A6000 GPUs. We fine-tune all models with LoRA in float32 mixed-precision, except for the Llama-2-13B-Chat model trained with QLoRA, and inference all open-sourced models in bfloat16 mixed-precision for efficiency. The hyperparameter settings for training KnowProxy and baselines are described in Table 7.

Hyperparameter	Value
Epoch	6
Batch size	16
Maximum input length	512
Optimizer	Adam
β_1	0.9
β_2	0.999
Learning rate	2e-4
Learning rate scheduling	Cosine decay
Weight decay	0.0
Warmup steps	0.0
LoRA rank	64
LoRA alpha	8
LoRA dropout	0.1

Table 7: Hyperparameter settings for training.

B.3 Knowledge-Eliciting Prompt Template

We design a knowledge generation prompt template for each target task, taking into account the specific characteristics of each task. The detailed prompt template is provided below:

1) Classification benchmarks (e.g., QASC, ScienceQA, etc.): Figure 7, Figure 8, and Figure 10.

2) TruthfulQA benchamrk: Figure 9.

C Detailed Experiment Results

C.1 Distinct Question-Answering Tasks

Method	TruthfulQA (True + Info (%))	ScienceQA (Acc. (%))
Fine-tuning LLMs	72.0	79.4
Zero-shot LLMs	66.0	77.6
Fine-tuning SLMs	61.0	66.6
KnowProxy	83.0	78.6

Table 8: Evaluation results on distinct tasks spanning truthful and informative (TruthfulQA) and scientific (ScienceQA) domains. We employ Llama 3.2 (3B) as LLM and Llama 3.2 (1B) as SLM.

To assess the effectiveness of KnowProxy across diverse domains, we conduct evaluations on two representative benchmarks: truthfulness and informativeness (TruthfulQA) and scientific reasoning (ScienceQA).

C.2 Detailed Results on Applicability across Proxy Choices

As shown in Table 9, we observe that the performance of KnowProxy is influenced by the capability of the lightweight proxy model.

Model	Approach	OBQA	ARC_h	PIQA	CSQA	StrategyQA	QASC	SIQA	Average
Llama 3.2 (3B)	Fine-tuning	82.2	76.2	87.7	79.5	71.5	82.9	80.5	80.1
	Zero-shot	72.2	68.6	75.8	67.7	60.9	75.9	65.3	69.5
w/ Llama 3.2 (1B)	Fine-tuning	73.2	60.9	80.3	72.0	66.5	68.0	74.9	70.8
	KnowProxy	80.2	75.2	83.4	75.0	72.9	78.1	76.3	77.3
w/ LaMini-GPT (0.7B)	Fine-tuning	56.0	27.3	70.4	49.5	61.9	20.4	68.2	50.5
	KnowProxy	74.6	75.2	78.7	72.5	67.7	78.0	71.9	74.1
w/ Qwen 2.5 (0.5B)	Fine-tuning	68.8	43.9	73.9	65.4	59.7	64.2	69.1	63.6
	KnowProxy	76.2	75.3	79.9	72.8	69.4	78.4	73.9	75.1

Table 9: The detailed results on applicability across proxy models. We employ Llama 3.2 (3B) as the frozen LLM.

C.3 Proxy Model Scalability

To demonstrate the relationship between the scalability of proxy models and KnowProxy, we evaluate KnowProxy on various reasoning benchmarks using the Pythia family, which differs only in model size while maintaining consistent design choices and training processes.

C.4 Resource Analysis

We further conduct a resource analysis to demonstrate the efficiency of KnowProxy in terms of the resources required during training. To further highlight the contribution of KnowProxy, which independently trains smaller proxy models while leveraging textual knowledge, we analyze how textual knowledge influences resource burden for training. As shown in Table 10, we observe that KnowProxy, across all proxy choices, is consistently more memory-efficient than parameter-efficient methods such as LoRA during training. Notably, on PhysicalIQA with long input sequences, KnowProxy exhibits substantial memory efficiency comparable to existing proxy-based approaches that fine-tune only the smaller model in target domains. These results demonstrate that KnowProxy is a resource-efficient approach for effectively adapting LLMs to target domains, ranging from white-box to black-box settings.

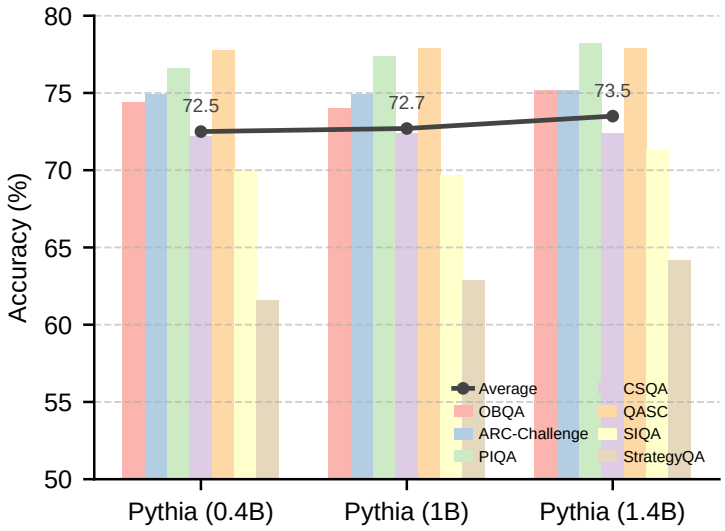


Figure 6: Comparison of effectiveness with respect to the scalability of small models. Here, we utilize llama 3.2 (3B) as LLM.

Approach	PIQA	SIQA	OBQA	QASC
Direct Fine-tuning (LoRA)	74.5	38.0	30.4	36.7
Proxy-tuning	33.9	16.9	15.5	16.1
CombLM	40.4	23.4	22.0	22.6
KNOWPROXY (Llama 3.2 (1B))	33.9	24.6	26.9	27.1
KNOWPROXY (LaMini-GPT (0.7B))	38.1	26.8	29.2	31.2
KNOWPROXY (Qwen 2.5 (0.5B))	24.6	16.8	19.6	19.3

Table 10: Comparison of the maximum GPU memory required for training process. We set the target LLM for training in each task to Llama 3.2 (3B). For each proxy model, KNOWPROXY is evaluated under the hyperparameter settings that yield the best performance.

Llama 3.2 (3B) & Llama 2 (13B)

You are a helpful assistant that extracts the knowledge needed to easily solve the given question and then predicts the correct answer.

Guidelines
- Your task is to extract the knowledge needed to easily solve the given question and then predict the correct answer using your generated knowledge.
- Specifically, the knowledge you extract must be relevant to what you believe is the correct answer.

CAUSION
- You MUST NOT respond any of the given answer choices as the answer.
- Correctly generate the knowledge.
- Strictly follow the answer format mentioned.
- Do not overestimate your confidence.

Question
{Specific Question Here}
Human: Given all of the above, read the question, break down the problem into 4 steps which are knowledge needed to easily solve the given problem, think step by step, give your confidence in each step. Note: The confidence indicates how likely you think your answer is true.

Use the following format to answer:
{Specific Output Format Here}
Assistant: The Response is (

Figure 7: The designed prompt for Llama family aimed at generating knowledge in complex reasoning benchmarks.

ChatGPT

You are a helpful assistant that extracts the knowledge needed to easily solve the given question and then predicts the correct answer.

Guidelines
1. Your task is to extract the knowledge needed to easily solve the given question and then predict the correct answer using your generated knowledge.
2. Specifically, the knowledge you extract must be relevant to what you believe is the correct answer.
3. Given all of the above, read the question, break down the problem into 4 steps which are knowledge needed to easily solve the given problem, think step by step, give your confidence in each step.
4. Note: The confidence indicates how likely you think your answer is true.
5. Use the following format to answer:
{Specific Output Format Here}

CAUSION
- You MUST NOT respond any of the given answer choices as the answer.
- Correctly generate the knowledge.
- Do not overestimate your confidence.

Question
{Specific Question Here}

Figure 8: The designed prompt for ChatGPT aimed at generating knowledge in complex reasoning benchmarks.

Llama 3.2 (3B) (TruthfulQA)

You are a helpful, respectful, and honest assistant that extracts the knowledge needed to solve the question easily and helpfully, and then predicts the correct and safe answer.

Guidelines
- Your task is to extract the knowledge required to solve the given question easily and helpfully, and then predict the correct and safe answer based on your generated knowledge.
- Specifically, the knowledge you extract must be relevant to what you believe is the correct and safe answer.
- Your responses should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content.
- If the given question does not make sense or is not factually coherent, please don't answer false information.

CAUSION
- Correctly generate the knowledge.
- Please ensure that your knowledge and answer are socially unbiased and positive in nature.
- Strictly follow the answer format mentioned.
- Do not overestimate your confidence.

Question
{Specific Question Here}
Human: Given all of the above, read the question, break down the problem into 4 steps which are knowledge needed to solve the given problem easily and helpfully, think step by step, give your confidence in each step. Note: The confidence indicates how likely you think your answer is true.

Use the following format to answer:
{Specific Output Format Here}
Assistant: The Response is (

Figure 9: The designed prompt for Llama 3.2 aimed at generating knowledge associated with TruthfulQA dataset.

Llama 3.2 (3B) (ScienceQA)

You are a helpful assistant that extracts the knowledge needed to easily solve the given question and then predicts the correct answer.

Guidelines
- Your task is to extract the knowledge needed to easily solve the given question and then predict the correct answer using your generated knowledge.
- Specifically, the knowledge you extract must be relevant to what you believe is the correct answer.

CAUSION
- You MUST NOT respond any of the given answer choices as the answer.
- Correctly generate the knowledge.
- Strictly follow the answer format mentioned.
- Do not overestimate your confidence.

Question
{Specific Question Here}
Human: Given all of the above, read the question, break down the problem into 4 steps which are knowledge needed to easily solve the given problem, think step by step, give your confidence in each step. Note: The confidence indicates how likely you think your answer is true.

Use the following format to answer:
{Specific Output Format Here}
Assistant: The Response is (

Figure 10: The designed prompt for Llama 3.2 aimed at generating knowledge associated with the ScienceQA dataset.